MODEL CHECKING CTL IS ALMOST ALWAYS INHERENTLY SEQUENTIAL [*]

OLAF BEYERSDORFF [a], ARNE MEIER [b], MARTIN MUNDHENK [c], THOMAS SCHNEIDER [d], MICHAEL THOMAS [e], AND HERIBERT VOLLMER [f]

[a,b,e,f] Theoretical Computer Science, Leibniz University of Hannover, Germany
e-mail address: {beyersdorff, meier, thomas, vollmer}@thi.uni-hannover.de

[c] Computer Science, University of Jena, Germany
e-mail address: martin.mundhenk@uni-jena.de

[d] Computer Science, Saarland University, Germany
e-mail address: schneider@ps.uni-saarland.de

ABSTRACT. The model checking problem for CTL is known to be P-complete (Clarke, Emerson, and Sistla (1986), see Schnoebelen (2002)). We consider fragments of CTL obtained by restricting the use of temporal modalities or the use of negations—restrictions already studied for LTL by Sistla and Clarke (1985) and Markey (2004). For all these fragments, except for the trivial case without any temporal operator, we systematically prove model checking to be either inherently sequential (P-complete) or very efficiently parallelizable (LOGCFL-complete). For most fragments, however, model checking for CTL is already P-complete. Hence our results indicate that, in cases where the combined complexity is of relevance, approaching CTL model checking by parallelism cannot be expected to result in any significant speedup.

We also completely determine the complexity of the model checking problem for all fragments of the extensions ECTL, CTL^+, and $ECTL^+$.

1. INTRODUCTION

Temporal logic was introduced by Pnueli [Pnu77] as a formalism to specify and verify properties of concurrent programs. Computation Tree Logic (CTL), the logic of branching time, goes back to Emerson and Clarke [EC82] and contains temporal operators for expressing that an event occurs at some time in the future (F), always in the future (G), in the next point of time (X), always in the future until another event holds (U), or as long as it is not released by the occurrence of another event (R), as well as path quantifiers (E, A) for speaking about computation paths. The full language obtained by these operators and quantifiers

is called CTL* [EH86]. In CTL, the interaction between the temporal operators and path quantifiers is restricted. The temporal operators in CTL are obtained by path quantifiers followed directly by any temporal operator, *e.g.*, AF and AU are CTL-operators. Because they start with the universal path quantifier, they are called *universal* CTL-*operators*. Accordingly, EX and EG are examples for *existential* CTL-*operators*.

Since properties are largely verified automatically, the computational complexity of reasoning tasks is of great interest. Model checking (MC)—the problem of verifying whether a given formula holds in a state of a given model—is one of the most important reasoning tasks [Sch03]. It is intractable for CTL* (PSPACE-complete [EL87, Sch03]), but tractable for CTL (complete for polynomial time [CES86, Sch03]).

Although model checking for CTL is tractable, its P-hardness means that it is presumably not efficiently parallelizable. We therefore search for fragments of CTL with a model checking problem of lower complexity. We will consider all subsets of CTL-operators, and examine the complexity of the model checking problems for all resulting fragments of CTL. Further, we consider three additional restrictions affecting the use of negation and study the extensions ECTL, CTL$^+$, and their combination ECTL$^+$.

The complexity of model checking for fragments of temporal logics has been examined in the literature: Markey [Mar04] considered satisfiability and model checking for fragments of Linear Temporal Logic (LTL). Under systematic restrictions to the temporal operators, the use of negation, and the interaction of future and past operators, Markey classified the two decision problems into NP-complete, coNP-complete, and PSPACE-complete. Further, [BMS$^+$09] examined model checking for all fragments of LTL obtained by restricting the set of temporal operators and propositional connectives. The resulting classification separated cases where model checking is tractable from those where it is intractable. For model checking paths in LTL an $AC^1(LOGDCFL)$ algorithm is presented in [KF09].

Concerning CTL and its extension ECTL, our results in this paper show that most restricted versions of the model checking problem exhibit the same hardness as the general problem. More precisely, we show that apart from the trivial case where CTL-operators are completely absent, the complexity of CTL model checking is a dichotomy: it is either P-complete or LOGCFL-complete. Unfortunately, the latter case only occurs for a few rather weak fragments and hence there is not much hope that in practice, model checking can be sped up by using parallelism—it is inherently sequential.

Put as a simple rule, model checking for CTL is P-complete for every fragment that allows to express a universal *and* an existential CTL-operator. Only for fragments involving the operators EX and EF (or alternatively AX and AG) model checking is LOGCFL-complete. This is visualized in Figure 4 in Section 5. Recall that LOGCFL is defined as the class of problems logspace-reducible to context-free languages, and $NL \subseteq LOGCFL \subseteq NC^2 \subseteq P$. Hence, in contrast to inherently sequential P-hard tasks, problems in LOGCFL have very efficient parallel algorithms.

For the extensions CTL$^+$ and ECTL$^+$, the situation is more complex. In general, model checking CTL$^+$ and ECTL$^+$ is Δ_2^p-complete [LMS01]. We show that for $T \subseteq \{A, E, X\}$, both model checking problems restricted to operators from T remain tractable, while for $T \nsubseteq \{A, E, X\}$, they become Δ_2^p-complete. Yet, for negation restricted fragments with only existential or only universal path quantifiers, we observe a complexity decrease to NP- resp. coNP-completeness.

This paper is organized as follows: Section 2 introduces CTL, its model checking problems, and the non-basics of complexity theory we use. Section 3 contains our main results,

separated into upper and lower bounds. We also provide a refined analysis of the reductions between different model checking problems with restricted use of negation. The results are then generalized to extensions of CTL in Section 4. Finally, Section 5 concludes with a graphical overview of the results.

2. Preliminaries

2.1. **Temporal Logic.** We inductively define CTL*-formulae as follows. Let Φ be a finite set of atomic propositions. The symbols used are the atomic propositions in Φ, the constant symbols $\top$, $\bot$, the Boolean connectives $\neg$, $\wedge$, and $\vee$, and the temporal operator symbols A, E, X, F, G, U, and R.

A and E are called a *path quantifiers*, temporal operators aside from A and E are *pure temporal operators*. The atomic propositions and the constants $\top$ and $\bot$ are *atomic formulae*. There are two kinds of formulae, *state formulae* and *path formulae*. Each atomic formula is a state formula, and each state formula is a path formula. If φ, ψ are state formulae and χ, π are path formulae, then $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $\mathsf{A}\chi$, $\mathsf{E}\chi$ are state formulae, and $\neg\chi$, $(\chi \wedge \pi)$, $(\chi \vee \pi)$, $\mathsf{X}\chi$, $\mathsf{F}\chi$, $\mathsf{G}\chi$, $[\chi\mathsf{U}\pi]$, and $[\chi\mathsf{R}\pi]$ are path formulae. The set of CTL*-formulae (or *formulae*) consists of all state formulae.
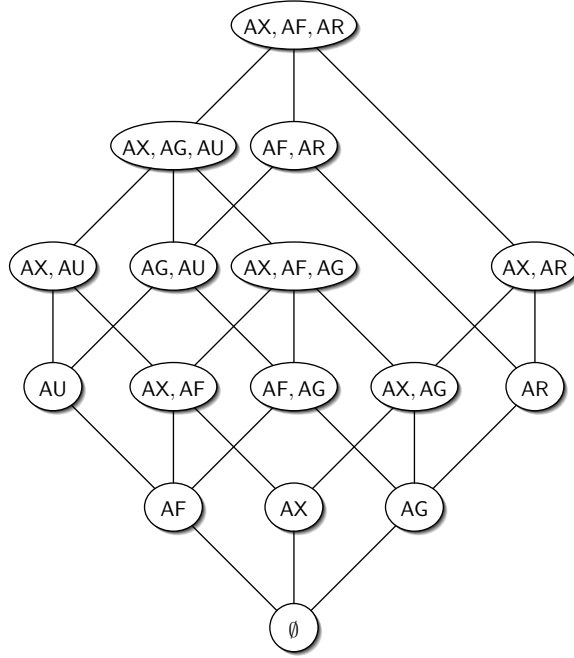
A *Kripke structure* is a triple $K = (W, R, \eta)$, where W is a finite set of states, $R \subseteq W \times W$ a total relation (*i.e.*, for each $w \in W$, there exists a w' such that $(w, w') \in R$), and $\eta \colon W \to \mathfrak{P}(\Phi)$ is a labelling function. A *path* x is an infinite sequence $x = (x_1, x_2, \ldots) \in W^\omega$ such that $(x_i, x_{i+1}) \in R$, for all $i \geq 1$. For a path $x = (x_1, x_2, \ldots)$ we denote by x^i the path $(x_i, x_{i+1}, \ldots)$.

Let $K = (W, R, \eta)$ be a Kripke structure, $w \in W$ be a state, and $x = (x_1, x_2, \ldots) \in W^\omega$ be a path. Further, let φ, ψ be state formulae and χ, π be path formulae. The truth of a CTL*-formula w.r.t. K is inductively defined as follows:

$$\begin{aligned}
&K, w \models \top && \text{always,} \\
&K, w \models \bot && \text{never,} \\
&K, w \models p && \text{iff } p \in \Phi \text{ and } p \in \eta(w), \\
&K, w \models \neg\varphi && \text{iff } K, w \not\models \varphi, \\
&K, w \models (\varphi \wedge \psi) && \text{iff } K, w \models \varphi \text{ and } K, w \models \psi, \\
&K, w \models (\varphi \vee \psi) && \text{iff } K, w \models \varphi \text{ or } K, w \models \psi, \\
&K, w \models \mathsf{A}\chi && \text{iff } K, x \models \chi \text{ for all paths } x = (x_1, x_2, \ldots) \text{ with } x_1 = w, \\
&K, x \models \varphi && \text{iff } K, x_1 \models \varphi, \\
&K, x \models \neg\chi && \text{iff } K, x \not\models \chi, \\
&K, x \models (\chi \wedge \pi) && \text{iff } K, x \models \chi \text{ and } K, x \models \pi, \\
&K, x \models (\chi \vee \pi) && \text{iff } K, x \models \chi \text{ or } K, x \models \pi, \\
&K, x \models \mathsf{X}\chi && \text{iff } K, x_2 \models \chi \\
&K, x \models [\chi\mathsf{U}\pi] && \text{iff there is a } k \in \mathbb{N} \text{ such that } K, x^k \models \pi \text{ and } K, x^i \models \chi \text{ for } 1 \leq i < k.
\end{aligned}$$

The semantics of the remaining temporal operators is defined via the equivalences: $\mathsf{E}\chi \equiv \neg\mathsf{A}\neg\chi$, $\mathsf{F}\chi \equiv [\top\mathsf{U}\chi]$, $\mathsf{G}\chi \equiv \neg\mathsf{F}\neg\chi$, and $[\chi\mathsf{R}\pi] \equiv \neg[\neg\chi\mathsf{U}\neg\pi]$. A state formula φ is *satisfied by a Kripke structure K* if there exists $w \in W$ such that $K, w \models \varphi$. We will also denoted this by $K \models \varphi$.

We use CTL$^*(T)$ to denote the set of CTL*-formulae using the Boolean connectives $\{\wedge, \vee, \neg\}$, and the temporal operators in T only. If T does not contain any quantifiers, then including any pure temporal operators in T is meaningless.

Figure 1: The expressive power of CTL(T).

A CTL-*formula* is a CTL*-formula in which each path quantifier is followed by exactly one pure temporal operator and each pure temporal operator is preceded by exactly one path quantifier. The set of CTL-formulae forms a strict subset of the set of all CTL*-formulae. For example, $\mathsf{AGEF}p$ is a CTL-formula, but $\mathsf{A}(\mathsf{GF}p \wedge \mathsf{F}q)$ is not. CTL is less expressive than CTL* [EH85, EH86].

Pairs of path quantifiers and pure temporal operators are called CTL-*operators*. The operators AX, AF, AG, AU, and AR are *universal* CTL-operators, and EX, EF, EG, EU, and ER are *existential* CTL-operators. Let ALL denote the set of all universal and existential CTL-operators. Note that $\mathsf{A}[\psi\mathsf{U}\chi] \equiv \mathsf{AF}\chi \wedge \neg\mathsf{E}[\neg\chi\mathsf{U}(\neg\psi \wedge \neg\chi)]$, and thus $\mathsf{E}[\psi\mathsf{R}\chi] \equiv \mathsf{EG}\chi \vee \mathsf{E}[\chi\mathsf{U}(\psi \wedge \chi)]$. Hence $\{\mathsf{AX}, \mathsf{AF}, \mathsf{AR}\}$ is a minimal set of operators for CTL (in presence of all Boolean connectives), whereas $\{\mathsf{AX}, \mathsf{AG}, \mathsf{AU}\}$ is not [Lar95].

By CTL(T) we denote the set of CTL-formulae using the connectives $\{\wedge, \vee, \neg\}$ and the CTL-operators in T only. Figure 1 shows the structure of sets of CTL-operators with respect to their expressive power.

Moreover, we define the following fragments of CTL(T).

– CTL$_{\mathrm{pos}}(T)$ (positive)
   CTL-operators may not occur in the scope of a negation.

– CTL$_{\mathrm{a.n.}}(T)$ (atomic negation)
   Negation signs appear only directly in front of atomic propositions.

– CTL$_{\mathrm{mon}}(T)$ (monotone)
   No negation signs allowed.

This restricted use of negation was introduced and studied in the context of linear temporal logic, LTL, by Sistla and Clarke [SC85] and Markey [Mar04]. Their original notation was $\widetilde{\mathrm{L}}(T)$ for $\mathrm{CTL}_{\mathrm{a.n.}}(T)$ and $\mathrm{L}^{+}(T)$ for $\mathrm{CTL}_{\mathrm{pos}}(T)$.

2.2. **Model Checking.** Now we define the *model checking problems* for the above mentioned fragments of CTL. Let $\mathcal{L}$ be CTL, $\mathrm{CTL}_{\mathrm{mon}}$, $\mathrm{CTL}_{\mathrm{a.n.}}$, or $\mathrm{CTL}_{\mathrm{pos}}$.

Problem: $\mathcal{L}$-MC(T)
Input: A Kripke structure $K = (W, R, \eta)$, a state $w \in W$, and an $\mathcal{L}(T)$-formula φ.
Question: Does $K, w \models \varphi$ hold?

2.3. **Complexity Theory.** We assume familiarity with standard notions of complexity theory as introduced in, *e.g.*, [Pap94]. Next we will introduce the notions from circuit complexity that we use for our results. All reductions in this paper are $\leq_{\mathrm{cd}}$-reductions defined as follows: A language A is *constant-depth reducible* to B, $A \leq_{\mathrm{cd}} B$, if there is a logtime-uniform AC^0-circuit family with oracle gates for B that decides membership in A. That is, there is a circuit family $\mathcal{C} = (C_1, C_2, C_3, \dots)$ such that
- for every n, C_n computes the characteristic function of A for inputs of length n,
- there is a polynomial p and a constant d such that for all input lengths n, the size of C_n is bounded by $p(n)$ and the depth of C_n is bounded by d,
- each circuit C_n consists of unbounded fan-in AND and OR gates, negation gates, and gates that compute the characteristic function of B (the *oracle gates*),
- there is a linear-time Turing machine M that can check the structure of the circuit family, that is, given a tuple $\langle n, g, t, h \rangle$ where n, g, h are binary numbers and $t \in \{\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}, \mathrm{ORACLE}\}$, M accepts if C_n contains a gate g of type t with predecessor h.

Circuit families $\mathcal{C}$ with this last property are called *logtime-uniform* (the name stems from the fact that the time needed by M is linear in the length of its input tuple, hence logarithmic in n). For background information we refer to [RV97, Vol99].

We easily obtain the following relations between model checking for fragments of CTL with restricted negation:

Lemma 2.1. *For every set T of* CTL*-operators, we have*

$$\mathrm{CTL}_{\mathrm{mon}}\text{-}\mathrm{MC}(T) \leq_{\mathrm{cd}} \mathrm{CTL}_{\mathrm{a.n.}}\text{-}\mathrm{MC}(T) \leq_{\mathrm{cd}} \mathrm{CTL}_{\mathrm{pos}}\text{-}\mathrm{MC}(T).$$

Further, for model checking, atomic negation can be eluded, that is, $\mathrm{CTL}_{\mathrm{a.n.}}\text{-}\mathrm{MC}(T) \leq_{\mathrm{cd}}$ $\mathrm{CTL}_{\mathrm{mon}}\text{-}\mathrm{MC}(T)$.

Proof. The first part is straightforward, using the identity function as reduction function. For the second part, let $K = (W, R, \eta)$ be a Kripke structure and let φ be a $\mathrm{CTL}_{\mathrm{a.n.}}(T)$-formula over the propositions $\Phi = \{p_1, \dots, p_n\}$. Every negation in φ appears inside a negative literal. We obtain φ' by replacing every negative literal $\neg p_i$ with a fresh atomic proposition q_i. Further define $K' = (W, R, \eta')$, where $\eta'(w) = \eta(w) \cup \{q_i \mid p_i \notin \eta(w)\}$. Obviously, $K, w \models \varphi$ iff $K', w \models \varphi'$ for all $w \in W$. The mapping $(K, w, \varphi) \mapsto (K', w, \varphi')$ can be performed by an AC^0-circuit. $\square$

In Section 3.3, we complete the picture by proving $\mathrm{CTL_{pos}\text{-}MC}(T) \leq_{\mathrm{cd}} \mathrm{CTL_{mon}\text{-}MC}(T)$.

The class P consists of all languages that have a polynomial-time decision algorithm. A problem is P-complete if it is in P and every other problem in P reduces to it. P-complete problems are sometimes referred to as *inherently sequential*, because P-complete problems most likely (formally: if $P \neq NC$) do not possess NC-algorithms, that is, algorithms running in polylogarithmic time on a parallel computer with a polynomial number of processors. Formally, NC contains all problems solvable by polynomial-size polylogarithmic-depth logtime-uniform families of circuits with bounded fan-in AND, OR, NOT gates.

There is an NC-algorithm for parsing context-free languages, that is, $\mathrm{CFL} \subseteq \mathrm{NC}$. Therefore, complexity theorists have studied the class LOGCFL of all problems reducible to context-free languages (the name "LOGCFL" refers to the original definition of the class in terms of logspace-reductions, however it is known that the class does not change if instead, as everywhere else in this paper, $\leq_{\mathrm{cd}}$-reductions are used). Hence, $\mathrm{LOGCFL} \subseteq \mathrm{NC}$ (even $\mathrm{LOGCFL} \subseteq \mathrm{NC}^2$, the second level of the NC-hierarchy, where the depth of the occurring circuits is restricted to $O(\log^2 n)$). The class LOGCFL has a number of different maybe even somewhat surprising characterizations, *e.g.*, languages in LOGCFL are those that can be decided by nondeterministic Turing machines operating in polynomial time that have a worktape of logarithmic size and additionally a stack whose size is not bounded.

More important for this paper is the characterization of LOGCFL as those problems computable by SAC^1 circuit families, that is, families of circuits that

- have polynomial size and logarithmic depth,
- consist of unbounded fan-in OR gates and bounded fan-in AND gates and negation gates, but the latter are only allowed at the input-level,
- are logtime-uniform (as defined above).

Since the class LOGCFL is known to be closed under complementation, the second condition can equivalently be replaced to allow unbounded fan-in AND gates and restrict the fan-in of OR gates to be bounded.

To summarize:

$$\mathrm{NC}^1 \subseteq L \subseteq \mathrm{NL} \subseteq \mathrm{LOGCFL} = \mathrm{SAC}^1 \subseteq \mathrm{NC}^2;$$

and problems in these classes possess very efficient parallel algorithms: they can be solved in time $O(\log^2 n)$ on a parallel machine with a tractable number of processors. For more background on these and related complexity classes, we refer the reader to [Vol99].

3. Model Checking CTL and CTL$_{\mathrm{POS}}$

This section contains our main results on the complexity of model checking for CTL and $\mathrm{CTL_{pos}}$. We defer the analysis of the fragments $\mathrm{CTL_{a.n.}}$ and $\mathrm{CTL_{mon}}$ to Section 3.3, where we will see that their model-checking problems are computationally equivalent to model checking for $\mathrm{CTL_{pos}}$.

While model checking for CTL in general is known to be polynomial time solvable and in fact P-complete [CES86, Sch03], we improve the lower bound by showing that only one temporal operator is sufficient to obtain hardness for P.

Theorem 3.1. *For each nonempty set T of* CTL*-operators,* CTL*-MC*(T) *is* P*-complete. If $T = \emptyset$, then* CTL*-MC*(T) *is* NC^1*-complete.*

If we consider only formulae from $\mathrm{CTL}_{\mathrm{pos}}$, where no CTL-operators are allowed inside the scope of a negation, the situation changes and the complexity of model checking exhibits a dichotomous behavior. As long as EG or AF are expressible the model checking problem remains P-complete. Otherwise, its complexity drops to LOGCFL.

Theorem 3.2. *Let T be any set of* CTL-*operators. Then* $\mathrm{CTL}_{\mathrm{pos}}$-$\mathrm{MC}(T)$ *is*

– NC^1-*complete if $T = \emptyset$,*
– LOGCFL-*complete if $\emptyset \subsetneq T \subseteq \{\mathsf{EX}, \mathsf{EF}\}$ or $\emptyset \subsetneq T \subseteq \{\mathsf{AX}, \mathsf{AG}\}$, and*
– P-*complete otherwise.*

We split the proofs of Theorems 3.1 and 3.2 into the upper and lower bounds in the following two subsections.

3.1. **Upper Bounds.** In general, model checking for CTL is known to be solvable in P [CES86]. While this upper bound also applies to $\mathrm{CTL}_{\mathrm{pos}}$-$\mathrm{MC}(T)$ (for every T), we improve it for positive CTL-formulae using only EX and EF, or only AX and AG.

Proposition 3.3. *Let T be a set of* CTL-*operators such that $T \subseteq \{\mathsf{EX}, \mathsf{EF}\}$ or $T \subseteq \{\mathsf{AX}, \mathsf{AG}\}$. Then* $\mathrm{CTL}_{\mathrm{pos}}$-$\mathrm{MC}(T)$ *is in* LOGCFL.

Proof. First consider the case $T \subseteq \{\mathsf{EX}, \mathsf{EF}\}$. We claim that Algorithm 1 recursively decides whether the Kripke structure $K = (W, R, \eta)$ satisfies the $\mathrm{CTL}_{\mathrm{pos}}(T)$-formula φ in state $w_0 \in W$. There, S is a stack that stores pairs $(\varphi, w) \in \mathrm{CTL}_{\mathrm{pos}}(T) \times W$ and $R^\star$ denotes the transitive closure of R.

Algorithm 1 Determine whether $K, w_0 \models \varphi$.

Require: a Kripke structure $K = (W, R, \eta)$, $w_0 \in W$, $\varphi \in \mathrm{CTL}_{\mathrm{pos}}(T)$
 1: push$(S, (\varphi, w_0))$
 2: **while** S is not empty **do**
 3:   $(\varphi, w) \leftarrow$ pop(S)
 4:   **if** φ is a propositional formula **then**
 5:     **if** φ evaluates to false in w under η **then**
 6:       **return false**
 7:     **end if**
 8:   **else if** $\varphi = \alpha \wedge \beta$ **then**
 9:     push$(S, (\beta, w))$
10:     push$(S, (\alpha, w))$
11:   **else if** $\varphi = \alpha \vee \beta$ **then**
12:     nondet. push$(S, (\alpha, w))$ or push$(S, (\beta, w))$
13:   **else if** $\varphi = \mathsf{EX}\alpha$ **then**
14:     nondet. choose $w' \in \{w' \mid (w, w') \in R\}$
15:     push$(S, (\alpha, w'))$
16:   **else if** $\varphi = \mathsf{EF}\alpha$ **then**
17:     nondet. choose $w' \in \{w' \mid (w, w') \in R^\star\}$
18:     push$(S, (\alpha, w'))$
19:   **end if**
20: **end while**
21: **return true**

Algorithm 1 always terminates because each subformula of φ is pushed to the stack S at most once. For correctness, an induction on the structure of formulae shows that Algorithm 1 returns **false** if and only if for the most recently popped pair (ψ, w) from S, we have $K, w \not\models \psi$. Thence, in particular, Algorithm 1 returns **true** iff $K, w \models \varphi$.

Algorithm 1 can be implemented on a nondeterministic polynomial-time Turing machine that besides its (unbounded) stack uses only logarithmic memory for the local variables. Thus $\mathrm{CTL_{pos}}$-MC(T) is in LOGCFL.

The case $T \subseteq \{\mathsf{AX}, \mathsf{AG}\}$ is analogous and follows from closure of LOGCFL under complementation. □

Finally, for the trivial case where no CTL-operators are present, model checking $\mathrm{CTL}(\emptyset)$-formulae is equivalent to the problem of evaluating a propositional formula. This problem is known to be solvable in NC^1 [Bus87].

3.2. **Lower Bounds.** The P-hardness of model checking for CTL was first stated in [Sch03]. We improve this lower bound and concentrate on the smallest fragments of monotone CTL—w.r.t. CTL-operators—with P-hard model checking.

Proposition 3.4. *Let T denote a set of* CTL-*operators. Then* $\mathrm{CTL_{mon}}$-MC(T) *is* P-*hard if T contains an existential and a universal* CTL-*operator.*

Proof. First, assume that $T = \{\mathsf{AX}, \mathsf{EX}\}$. We give a generic reduction from the word problem for alternating Turing machines working in logarithmic space, which follows the same line as the classical proof idea (see [Sch03, Theorem 3.8]), and which we will modify in order to be useful for other combinations of CTL-operators. Let M be an alternating logspace Turing machine, and let x be an input to M. We may assume w.l.o.g. that each transition of M leads from an existential to a universal configuration and vice versa. Further we may assume that each computation of M ends after the same number $p(n)$ of steps, where p is a polynomial and n is the length of M's input. Furthermore we may assume that there exists a polynomial q such that $q(n)$ is the number of configurations of M on any input of length n.

Let $c_1, \ldots, c_{q(n)}$ be an enumeration of all possible configurations of M on input x, starting with the initial configuration c_1. We construct a Kripke structure $K := (W, R, \eta)$ by defining the set $W := \{c_i^j \mid 1 \le i \le q(n), 0 \le j \le p(n)\}$ and the relation $R \subseteq W \times W$ as

$$R := \left\{ (c_i^j, c_k^{j+1}) \mid M \text{ reaches configuration } c_k \text{ from } c_i \text{ in one step}, 0 \le j < p(n) \right\}$$

$$\cup \left\{ (c_i^j, c_i^j) \mid c_i^j \text{ has no successor}, 1 \le i \le q(n), 0 \le j < p(n) \right\}$$

$$\cup \left\{ (c_i^{p(n)}, c_i^{p(n)}) \mid 1 \le i \le q(n) \right\}.$$

The labelling function η is defined for all $c_i^j \in W$ as

$$\eta(c_i^j) := \begin{cases} \{t\}, & \text{if } c_i \text{ is an accepting configuration and } j = p(n) \\ \emptyset, & \text{otherwise} \end{cases}$$

where t is the only atom used by this labelling. It then holds that

$$M \text{ accepts } x \iff K, c_1^0 \models \psi_1\Big(\psi_2\big(\cdots \psi_{p(n)}(t)\big)\cdots\Big),$$

where $\psi_i(x) := \mathsf{AX}(x)$ if M's configurations *before* the ith step are universal, and $\psi_i(x) := \mathsf{EX}(x)$ otherwise. Notice that the constructed CTL-formula does not contain any Boolean

connective. Since $p(n)$ and $q(n)$ are polynomials, the size of K and φ is polynomial in the size of (M, x). Moreover, K and φ can be constructed from M and x using AC^0-circuits. Thus, $A \leq_{\mathrm{cd}} \mathrm{CTL}_{\mathrm{mon}}\text{-}\mathrm{MC}(\{\mathsf{AX}, \mathsf{EX}\})$ for all $A \in \mathrm{ALOGSPACE} = \mathrm{P}$.

For $T = \{\mathsf{AF}, \mathsf{EG}\}$ we modify the above reduction by defining the labelling function η and the formula ψ_i as follows:

$$\eta(c_i^j) := \begin{cases} \{d_j, t\}, \text{ if } c_i \text{ is an accepting configuration and } j = p(n) \\ \{d_j\}, \quad \text{otherwise} \end{cases}$$

$$\psi_i(x) := \begin{cases} \mathsf{AF}(d_i \wedge x), \text{ if } M\text{'s configurations } \textit{before} \text{ step } i \text{ are universal,} \\ \mathsf{EG}(D_i \vee x), \text{ otherwise,} \end{cases} \tag{3.1}$$

where d_j are atomic propositions encoding the 'time stamps' of the respective configurations and $D_i = \bigvee_{i \neq j \in \{0,\dots,p(n)\}} d_j$.

For the combinations of T being one of $\{\mathsf{AF}, \mathsf{EF}\}$, $\{\mathsf{AF}, \mathsf{EX}\}$, $\{\mathsf{AG}, \mathsf{EG}\}$, $\{\mathsf{AG}, \mathsf{EX}\}$, $\{\mathsf{AX}, \mathsf{EF}\}$, and $\{\mathsf{AX}, \mathsf{EG}\}$, the P-hardness of $\mathrm{CTL}_{\mathrm{mon}}\text{-}\mathrm{MC}(T)$ is obtained using analogous modifications to η and the ψ_i's.

For the remaining combinations involving the until or the release operator, observe that w.r.t. the Kripke structure K as defined in (3.1), $\mathsf{AF}(d_i \wedge x)$ and $\mathsf{EG}(D_i \vee x)$ are equivalent to $\mathsf{A}[d_{i-1}\mathsf{U}x]$ and $\mathsf{E}[d_{i-1}\mathsf{U}x]$, and that R and U are duals. $\qquad \square$

In the presence of arbitrary negation, universal operators are definable by existential operators and vice versa. Hence, from Proposition 3.4 we obtain the following corollary.

Corollary 3.5. *The model checking problem* $\mathrm{CTL}\text{-}\mathrm{MC}(T)$ *is P-hard for each nonempty set T of* CTL-*operators.*

Returning to monotone CTL, in most cases even one operator suffices to make model checking P-hard:

Proposition 3.6. *Let T denote a set of* CTL-*operators. Then* $\mathrm{CTL}_{\mathrm{mon}}\text{-}\mathrm{MC}(T)$ *is P-hard if T contains at least one of the operators* EG, EU, ER, AF, AU, *or* AR.

Proof. We modify the proof of Proposition 3.4 to work with EG only. The remaining fragments follow from the closure of P under complementation and $\mathsf{F}\chi \equiv \neg\mathsf{G}\neg\chi \equiv [\top\mathsf{U}\chi]$, $[\chi\mathsf{U}\pi] \equiv \neg[\neg\chi\mathsf{R}\neg\pi]$.

Let the machine M, the word x, the polynomials p, q, and K be as above. Further assume w.l.o.g. that M branches only binary in each step. Denote by $W_\exists$ (resp. $W_\forall$) the set of states corresponding to existential (resp. universal) configurations. We construct a Kripke structure $K' := (W', R, \eta)$ consisting of $q(n) + 1$ layers and a 'trap' as follows: let $W' := W \times \{1, \dots, q(n) + 1\} \cup \{z\}$. The transition relation $R \subseteq W' \times W'$ is defined as

$$R := \left\{ ((c_i^j, \ell), (c_k^{j+1}, \ell)) \;\middle|\; \begin{array}{l} c_i^j \in W_\exists, M \text{ reaches } c_k \text{ from } c_i \text{ in one step,} \\ 1 \leq \ell \leq q(n) + 1, 0 \leq j < p(n) \end{array} \right\}$$

$$\cup \left\{ \begin{array}{l} ((c_i^j, \ell), (c_k^{j+1}, i)), \\ ((c_k^{j+1}, i), (c_{k'}^{j+1}, q(n) + 1)), \\ ((c_{k'}^{j+1}, q(n) + 1), z) \end{array} \;\middle|\; \begin{array}{l} c_i^j \in W_\forall, M \text{ reaches } c_k \text{ and } c_{k'} \text{ from } c_i \text{ in} \\ \text{one step}, c_k \leq c_{k'}, 0 \leq j < p(n) \end{array} \right\}$$

$$\cup \left\{ ((c_i^{p(n)}, \ell), (c_i^{p(n)}, \ell)) \mid 1 \leq i \leq q(n), 1 \leq \ell \leq q(n) + 1 \right\}$$

$$\cup \left\{ (z, z) \right\}$$

That is, the arcs leaving an existential configurations c_i lead to the successor configurations of c_i inside each layer; while any universal configuration c_i has exactly one outgoing arc
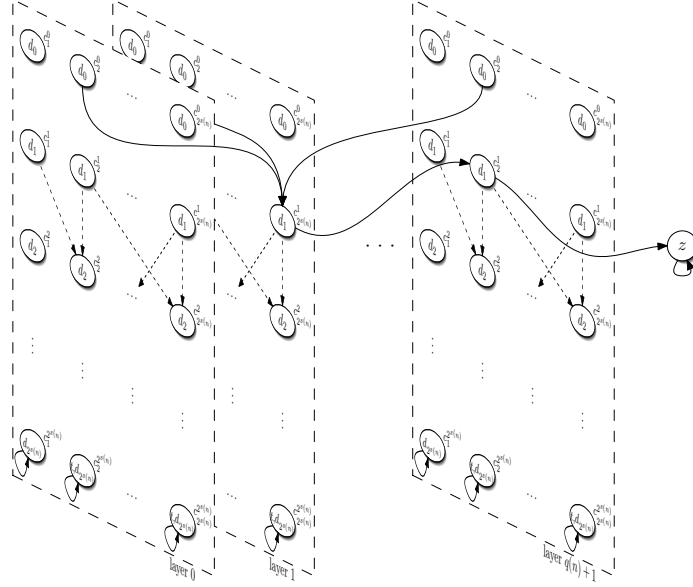
Figure 2: The Kripke structure K'; dashed (resp. solid) arrows correspond to transitions leaving existential (resp. universal) configurations.

pointing to its (lexicographically) first successor configuration in the layer i, from where another arc leads to the second successor of c_i in layer $q(n) + 1$, which in turn has an outgoing arc to the state z (see Figure 2). The labelling function η is defined as $\eta(z) := \{z\}$, $\eta((c_i^j, \ell)) := \{\ell, d_j, t\}$ if c_i is an accepting configuration, and otherwise $\eta((c_i^j, \ell)) := \{\ell, d_j\}$ for $(1 \leq \ell \leq q(n) + 1)$. Define

$$\psi_i(x) := \begin{cases} \mathsf{EG}(d_{i-1} \vee x \vee z), \text{if } M\text{'s configurations in step } i \text{ are universal,} \\ \mathsf{EG}(D_i \vee x), \text{if } M\text{'s configurations in step } i \text{ are existential,} \end{cases}$$

and $D_i = \bigvee_{i \neq j \in \{0,\dots,p(n)\}} d_j$. It is easily verified that $K, w \models \mathsf{AX}(d_i \wedge x)$ iff $K', (w, \ell) \models \mathsf{EG}\big(d_{i-1} \vee x \vee z\big)$, for all $w \in W_\forall$, $1 \leq \ell \leq q(n) + 1$ and $1 \leq i \leq p(n)$. From this, it easily follows that for

$$M \text{ accepts } x \iff K', (c_1^0, 1) \models \psi_1\Big(\psi_2\big(\cdots \psi_{p(n)}(t)\big)\cdots\Big).$$

As we essentially only duplicated the set of states in K and R can be constructed from all triples of states in W', K' remains AC^0 constructible. Concluding $A \leq_{\mathrm{cd}} \mathrm{CTL_{mon}\text{-}MC}(\{\mathsf{EG}\})$ for all $A \in \mathrm{P}$. $\qquad\square$

By Lemma 2.1, $\mathrm{CTL_{mon}\text{-}MC}(T) \leq_{\mathrm{cd}} \mathrm{CTL_{pos}\text{-}MC}(T)$ and hence the above results directly translate to model checking for $\mathrm{CTL_{pos}}$: for any set T of temporal operators, $\mathrm{CTL_{pos}\text{-}MC}(T)$ is P-hard if $T \not\subseteq \{\mathsf{EX}, \mathsf{EF}\}$ or if $T \not\subseteq \{\mathsf{AX}, \mathsf{AG}\}$. These results cannot be improved w.r.t. T, as for $T \subseteq \{\mathsf{EX}, \mathsf{EF}\}$ and $T \subseteq \{\mathsf{AX}, \mathsf{AG}\}$ we obtain a LOGCFL upper bound for model checking from Proposition 3.3. In the following proposition we prove the matching LOGCFL lower bound.

Proposition 3.7. *For every nonempty set T of CTL-operators, the model checking problem* $\mathrm{CTL_{mon}\text{-}MC}(T)$ *is LOGCFL-hard.*

Proof. As explained in Section 2.3, LOGCFL can be characterized as the set of languages recognizable by logtime-uniform SAC^1 circuits, *i.e.*, circuits of logarithmic depth and polynomial size consisting of $\vee$-gates with unbounded fan-in and $\wedge$-gates with fan-in 2. For every single CTL-operator O, we will show that $\mathrm{CTL}_{\mathrm{mon}}\text{-MC}(T)$ is LOGCFL-hard for all $T \supseteq \{O\}$ by giving a generic $\leq_{\mathrm{cd}}$-reduction f from the word problem for SAC^1 circuits to $\mathrm{CTL}_{\mathrm{mon}}\text{-MC}(T)$.

First, consider $\mathsf{EX} \in T$. Let C be a logtime-uniform SAC^1 circuit of depth ℓ with n inputs and let $x = x_1 \ldots x_n \in \{0,1\}^n$. Assume w.l.o.g. that C is connected, layered into alternating layers of $\wedge$-gates and $\vee$-gates, and that the output gate of C is an $\vee$-gate. We number the layers bottom-up, that is, the layer containing (only) the output gate has level 0, whereas the input-gates and negations of the input-gates are situated in layer ℓ. Denote the graph of C by $G = (V, E)$, where $V := V_{\mathrm{in}} \uplus V_\wedge \uplus V_\vee$ is partitioned into the sets corresponding to the (possibly negated) input-gates, the $\wedge$-gates, and the $\vee$-gates, respectively. G is acyclic and directed with paths leading from the input to the output gates. From (V, E) we construct a Kripke structure that allows to distinguish the two predecessors of an $\wedge$-gate from each other. This will be required to model proof trees using $\mathrm{CTL}_{\mathrm{mon}}(\{\mathsf{EX}\})$-formulae.

For $i \in \{1,2\}$, let $V_{\mathrm{in}}^i := \{v^i \mid v \in V_{\mathrm{in}}\}$, $V_\vee^i := \{v^i \mid v \in V_\vee\}$ and define $V_{\mathrm{in},\vee}^i := V_{\mathrm{in}}^i \cup V_\vee^i$. Further define

$$E' := \big\{(v, u^i) \in V_\wedge \times V_{\mathrm{in},\vee}^i \mid (u,v) \in E \text{ and } u \text{ is the } i\text{th predecessor of } v\big\}$$

$$\cup \big\{(v,v) \mid v \in V_{\mathrm{in}}^1 \cup V_{\mathrm{in}}^2\big\} \cup \bigcup_{i \in \{1,2\}} \big\{(v^i, u) \in V_{\mathrm{in},\vee}^i \times V_\wedge \mid (u,v) \in E\big\},$$

where the ordering of the predecessors is implicitly given in the encoding of C. We now define a Kripke structure $K := (V', E', \eta)$ with states $V' := V_{\mathrm{in},\vee}^1 \cup V_{\mathrm{in},\vee}^2 \cup V_\wedge$, transition relation E', and labelling function $\eta : V' \to \mathfrak{P}(\{1,2,t\})$,

$$\eta(v) := \begin{cases} \{i, t\}, & \text{if } (v = v_{\mathrm{in}_j} \in V_{\mathrm{in}}^i \text{ and } x_j = 1) \text{ or } (v = \overline{v}_{\mathrm{in}_j} \in V_{\mathrm{in}}^i \text{ and } x_j = 0), \\ \{i\}, & \text{if } (v = v_{\mathrm{in}_j} \in V_{\mathrm{in}}^i \text{ and } x_j = 0) \text{ or } (v = \overline{v}_{\mathrm{in}_j} \in V_{\mathrm{in}}^i \text{ and } x_j = 1) \text{ or } v \in V_\vee^i, \\ \emptyset, & \text{otherwise}, \end{cases}$$

where $i = 1, 2$, $j = 1, \ldots, n$ and $v_{\mathrm{in}_1}, \ldots, v_{\mathrm{in}_n}, \overline{v}_{\mathrm{in}_1}, \ldots, \overline{v}_{\mathrm{in}_n}$ enumerate the input gates and their negations. The formula φ that is to be evaluated on K will consist of atomic propositions 1, 2 and t, Boolean connectives $\wedge$ and $\vee$, and the CTL-operator EX. To construct φ we recursively define formulae $(\varphi_i)_{0 \le i \le \ell}$ by

$$\varphi_i := \begin{cases} t, & \text{if } i = \ell, \\ \mathsf{EX}\varphi_{i+1}, & \text{if } i \text{ is even } (\vee\text{-layers}), \\ \bigwedge_{i=1,2} \mathsf{EX}(i \wedge \varphi_{i+1}), & \text{if } i \text{ is odd } (\wedge\text{-layers}). \end{cases}$$

We define the reduction function f as the mapping $(C, x) \mapsto (K, v_0, \varphi)$, where v_0 is the node corresponding to the output gate of C and $\varphi := \varphi_0$. We stress that the size of φ is polynomial, for the depth of C is logarithmic only. Clearly, each minimal accepting subtree (cf. [Ruz80] or [Vol99, Definition 4.15]) of C on input x translates into a sub-structure K' of K such that $K', v_0 \models \varphi$, where

(1) K' includes v_0,
(2) K' includes one successor for every node corresponding to an $\vee$-gate, and
(3) K' includes the two successors of every node corresponding to an $\wedge$-gate.

As $C(x) = 1$ iff there exists a minimal accepting subtree of C on x, the LOGCFL-hardness of $\mathrm{CTL}_{\mathrm{mon}}$-$\mathrm{MC}(T)$ for $\mathsf{EX} \in T$ follows.

Second, consider $\mathsf{EF} \in T$. We have to extend our Kripke structure to contain information about the depth of the corresponding gate. We may assume w.l.o.g. that C is encoded such that each gate contains an additional counter holding the distance to the output gate (which is equal to the number of the layer it is contained in, cf. [Vol99]). We extend η to encode this distance i, $1 \leq i \leq \ell$, into the "depth-propositions" d_i as in the proof of Proposition 3.4. Denote this modified Kripke structure by K'. Further, we define $(\varphi'_i)_{0 \leq i \leq \ell}$ as

$$\varphi'_i := \begin{cases} t, & \text{if } i = \ell, \\ \mathsf{EF}(d_{i+1} \wedge \varphi'_{i+1}), & \text{if } i \text{ is even}, \\ \bigwedge_{i=1,2} \mathsf{EF}(d_{i+1} \wedge i \wedge \varphi'_{i+1}), & \text{if } i \text{ is odd}. \end{cases}$$

Redefining the reduction f as $(C, x) \mapsto (K', v_0, \varphi'_0)$ hence yields the LOGCFL-hardness of $\mathrm{CTL}_{\mathrm{mon}}$-$\mathrm{MC}(T)$ for $\mathsf{EF} \in T$.

Third, consider $\mathsf{AX} \in T$. Consider the reduction in case 1 for $\mathrm{CTL}_{\mathrm{mon}}(\{\mathsf{EX}\})$-formulae, and let $f(C, x) = (K, v_0, \varphi)$ be the value computed by the reduction function. It holds that $C(x) = 1$ iff $K, v_0 \models \varphi$, and equivalently $C(x) = 0$ iff $K, v_0 \models \neg\varphi$. Let φ' be the formula obtained from $\neg\varphi$ by multiplying the negation into the formula. Then φ' is a $\mathrm{CTL}_{\mathrm{a.n.}}(\{\mathsf{AX}\})$-formula. Since LOGCFL is closed under complement, it follows that $\mathrm{CTL}_{\mathrm{a.n.}}$-$\mathrm{MC}(\{\mathsf{AX}\})$ is LOGCFL-hard. Using Lemma 2.1, we obtain that $\mathrm{CTL}_{\mathrm{mon}}$-$\mathrm{MC}(\{\mathsf{AX}\})$ is LOGCFL-hard, too. An analogous argument works for the case $\mathsf{AG} \in T$. The remaining fragments are even P-complete by Proposition 3.6. □

Using Lemma 2.1 we obtain LOGCFL-hardness of $\mathrm{CTL}_{\mathrm{pos}}$-$\mathrm{MC}(T)$ for all nonempty sets T of CTL-operators.

In the absence of CTL-operators, the lower bound for the model checking problem again follows from the lower bound for evaluating monotone propositional formulae. This problem is known to be hard for NC^1 [Bus87, Sch10].

3.3. **The Power of Negation.** We will now show that model checking for the fragments $\mathrm{CTL}_{\mathrm{a.n.}}$ and $\mathrm{CTL}_{\mathrm{pos}}$ is computationally equivalent to model checking for $\mathrm{CTL}_{\mathrm{mon}}$, for any set T of CTL-operators. Since we consider $\leq_{\mathrm{cd}}$-reductions, this is not immediate.

From Lemma 2.1 it follows that the hardness results for $\mathrm{CTL}_{\mathrm{mon}}$-$\mathrm{MC}(T)$ also hold for $\mathrm{CTL}_{\mathrm{a.n.}}$-$\mathrm{MC}(T)$ and $\mathrm{CTL}_{\mathrm{pos}}$-$\mathrm{MC}(T)$. Moreover, the algorithms for $\mathrm{CTL}_{\mathrm{pos}}$-$\mathrm{MC}(T)$ also work for $\mathrm{CTL}_{\mathrm{mon}}$-$\mathrm{MC}(T)$ and $\mathrm{CTL}_{\mathrm{a.n.}}$-$\mathrm{MC}(T)$ without using more computation resources. Both observations together yield the same completeness results for all CTL-fragments with restricted negations.

Theorem 3.8. *Let T be any set of CTL-operators. Then $\mathrm{CTL}_{\mathrm{mon}}$-$\mathrm{MC}(T)$, $\mathrm{CTL}_{\mathrm{a.n.}}$-$\mathrm{MC}(T)$, and $\mathrm{CTL}_{\mathrm{pos}}$-$\mathrm{MC}(T)$ are*

− *NC^1-complete if T is empty,*
− *LOGCFL-complete if $\emptyset \subsetneq T \subseteq \{\mathsf{EX}, \mathsf{EF}\}$ or $\emptyset \subsetneq T \subseteq \{\mathsf{AX}, \mathsf{AG}\}$,*
− *P-complete otherwise.*

Moreover, the problems $\mathrm{CTL}_{\mathrm{mon}}$-$\mathrm{MC}(T)$, $\mathrm{CTL}_{\mathrm{a.n.}}$-$\mathrm{MC}(T)$, and $\mathrm{CTL}_{\mathrm{pos}}$-$\mathrm{MC}(T)$ are equivalent w.r.t. $\leq_{\mathrm{cd}}$-reductions.

Algorithm 2 Case distinction for $\mathsf{E}\overset{\infty}{\mathsf{F}}$

1: **else if** $\varphi = \mathsf{E}\overset{\infty}{\mathsf{F}}\alpha$ **then**
2:     nondet. choose $k \leq |W|$ and a path $(w_i)_{1 \leq i \leq k}$ such that $(w, w_1) \in R^{\star}$, $(w_k, w_1) \in R$
3:     nondet. choose some $1 \leq i \leq k$ and push$(S, (\alpha, w_i))$
4: **end if**

This equivalence extends Lemma 2.1. We remark that this equivalence is not straight-forward. Simply applying de Morgan's laws to transform one problem into another requires counting the number of negations on top of $\wedge$- and $\vee$-connectives. This counting cannot be achieved by an AC^0-circuit and does not lead to the aspired reduction. Here we obtain equivalence of the problems as a consequence of our generic hardness proofs in Section 3.2.

4. Model Checking Extensions of CTL

It has been argued that CTL lacks the ability to express fairness properties. To address this shortcoming, Emerson and Halpern introduced ECTL in [EH86]. ECTL extends CTL with the $\overset{\infty}{\mathsf{F}}$-operator, which states that for every moment in the future, the enclosed formula will eventually be satisfied again: for a Kripke structure K, a path $x = (x_1, x_2, \ldots)$, and a path formula χ

$$K, x \models \overset{\infty}{\mathsf{F}}\chi \quad \text{iff} \quad K, x^i \models \mathsf{F}\chi \ \text{ for all } i \in \mathbb{N}.$$

The dual operator $\overset{\infty}{\mathsf{G}}$ is defined analogously. As for CTL, model checking for ECTL is known to be tractable. Moreover, our next result shows that even for all fragments, model checking for ECTL is not harder than for CTL.

Theorem 4.1. *Let T be a set of temporal operators. Then* $\mathrm{ECTL\text{-}MC}(T) \equiv_{\mathrm{cd}} \mathrm{CTL\text{-}MC}(T')$ *and* $\mathrm{ECTL}_{\mathrm{pos}}\text{-}\mathrm{MC}(T) \equiv_{\mathrm{cd}} \mathrm{CTL}_{\mathrm{pos}}\text{-}\mathrm{MC}(T')$, *where T' is obtained from T by substituting* $\overset{\infty}{\mathsf{F}}$ *with* F *and* $\overset{\infty}{\mathsf{G}}$ *with* G.

Proof. For the upper bounds, notice that $\mathrm{ECTL\text{-}MC}(\mathsf{ALL} \cup \{\mathsf{E}\overset{\infty}{\mathsf{F}}, \mathsf{A}\overset{\infty}{\mathsf{F}}\}) \in \mathrm{P}$. It thus remains to show that $\mathrm{ECTL}_{\mathrm{pos}}\text{-}\mathrm{MC}(T) \in \mathrm{LOGCFL}$ for $T \subseteq \{\mathsf{EX}, \mathsf{EF}, \mathsf{E}\overset{\infty}{\mathsf{F}}\}$ and $T \subseteq \{\mathsf{AX}, \mathsf{AG}, \mathsf{A}\overset{\infty}{\mathsf{G}}\}$ First, consider the case that $T \subseteq \{\mathsf{EX}, \mathsf{EF}, \mathsf{E}\overset{\infty}{\mathsf{F}}\}$. We modify Algorithm 1 to handle $\mathsf{E}\overset{\infty}{\mathsf{F}}$ by extending the case distinction in lines 4–19 with the code fragment given in Algorithm 1. The algorithm for $T \subseteq \{\mathsf{AX}, \mathsf{AG}, \mathsf{A}\overset{\infty}{\mathsf{G}}\}$ is analogous and membership in LOGCFL follows from its closure under complementation.

For the lower bounds, we extend the proofs of Propositions 3.4, 3.6 and 3.7 to handle sets T involving also the operators $\mathsf{A}\overset{\infty}{\mathsf{F}}$, $\mathsf{A}\overset{\infty}{\mathsf{G}}$, $\mathsf{E}\overset{\infty}{\mathsf{F}}$, and $\mathsf{E}\overset{\infty}{\mathsf{G}}$. Therefore, we only need modify the accessibility relation R of respective Kripke structure K to be reflexive. The hardness results follow by replacing F with $\overset{\infty}{\mathsf{F}}$ and G with $\overset{\infty}{\mathsf{G}}$ in the respective reductions.

First consider the case that T contains an existential and a universal operator, say $T = \{\mathsf{A}\overset{\infty}{\mathsf{F}}, \mathsf{E}\overset{\infty}{\mathsf{G}}\}$. Let M, x, and p be defined as in the proof of Proposition 3.4. We map (M, x) to $(\tilde{K}, c_1^0, \psi_1)$, where $\tilde{K} = (W, R, \eta)$ is the reflexive closure of the Kripke structure K defined for the P-hardness of $\mathrm{CTL\text{-}MC}(\{\mathsf{AF}, \mathsf{EG}\})$, $c_1^0 \in W$, and $\psi := \psi_1\Big(\psi_2\big(\cdots \psi_{p(n)}(t)\big)\cdots\Big)$, where

$$\psi_i(x) := \begin{cases} \mathsf{A}\overset{\infty}{\mathsf{F}}(d_i \wedge x), & \text{if } M\text{'s configurations in step } i \text{ are universal,} \\ \mathsf{E}\overset{\infty}{\mathsf{G}}(D_i \vee x), & \text{otherwise,} \end{cases}$$

In $\tilde{K}$ it now holds that $d_i \in \eta(w)$ and $(w, w') \in R$ together imply that either $w = w'$ or $d_i \notin \eta(w')$. Hence, for all $w \in W$ and $1 \le i \le p(|x|)$, $\tilde{K}, w \models \mathsf{A}\overset{\infty}{\mathsf{F}}(d_i \wedge x)$ iff $K, w \models \mathsf{AF}(d_i \wedge x)$, and $\tilde{K}, w \models \mathsf{E}\overset{\infty}{\mathsf{G}}(\bigvee_{i \ne j \in \{0,\dots,p(n)\}} d_j \vee x)$ iff $K, w \models \mathsf{EG}(\bigvee_{i \ne j \in \{0,\dots,p(n)\}} d_j \vee x)$. From this, correctness of the reduction follows. The P-hardness of CTL-MC(T) for the remaining fragments follows analogously.

As for $T \subseteq \{\mathsf{EX}, \mathsf{EF}, \mathsf{E}\overset{\infty}{\mathsf{F}}\}$, we will show that ECTL$_{\mathrm{mon}}$-MC$(T)$ is LOGCFL-hard under $\le_{\mathrm{cd}}$-reductions for $T = \{\mathsf{E}\overset{\infty}{\mathsf{F}}\}$. Let C, x, and ℓ be as in the proof of Proposition 3.7. We map the pair (C, x) to the triple $(\tilde{K}', v_0, \varphi_0)$, where $\tilde{K}' = (V', E', \eta)$ is the reflexive closure of the Kripke structure K' defined for the LOGCFL-hardness of CTL-MC$(\{\mathsf{EF}\})$, $v_0 \in V'$, and φ_0 is recursively defined via $(\varphi'_i)_{0 \le i \le \ell}$ as

$$\varphi_i := \begin{cases} t, & \text{if } i = \ell, \\ \mathsf{E}\overset{\infty}{\mathsf{F}}(d_{i+1} \wedge \varphi_{i+1}), & \text{if } i \text{ is even}, \\ \bigwedge_{i=1,2} \mathsf{E}\overset{\infty}{\mathsf{F}}(d_{i+1} \wedge i \wedge \varphi_{i+1}), & \text{if } i \text{ is odd}. \end{cases}$$

Again, we have that in $\tilde{K}'$, $d_i \in \eta(v)$ and $(v, v') \in E'$ together imply that either $v = v'$ or $d_i \notin \eta(v')$. It hence follows $\tilde{K}', v \models \mathsf{E}\overset{\infty}{\mathsf{F}}(d_i \wedge \varphi_i)$ iff $K', v \models \mathsf{EF}(d_i \wedge \varphi_i)$, for all $v \in V'$ and $1 \le i \le \ell$. We conclude that ECTL$_{\mathrm{mon}}$-MC$(\{\mathsf{E}\overset{\infty}{\mathsf{F}}\})$ is LOGCFL-hard. The case $T = \{\mathsf{A}\overset{\infty}{\mathsf{G}}\}$ follows analogously. $\square$

We will now consider CTL$^+$, the extension of CTL by Boolean combinations of path formulae which is defined as follows. A CTL$^+$-*formula* is a CTL*-formula where each pure temporal operator in a state formula occurs in the scope of a path quantifier. The set of all CTL-formulae is a strict subset of the set of all CTL$^+$-formulae, which again forms a strict subset of the set of all CTL*-formulae. For example, $\mathsf{AGEF}p$ and $\mathsf{A}(\mathsf{G}p \wedge \mathsf{F}q)$ are CTL$^+$-formulae, but $\mathsf{AGF}p$ is not. However, CTL is as expressive as CTL$^+$ [EH85].

By CTL$^+(T)$ we denote the set of CTL$^+$-formulae using the connectives $\{\wedge, \vee, \neg\}$ and temporal operators in T only. Analogous to the fragments CTL$_{\mathrm{pos}}(T)$, CTL$_{\mathrm{a.n.}}(T)$, and CTL$_{\mathrm{mon}}(T)$, we define CTL$^+_{\mathrm{pos}}(T)$, CTL$^+_{\mathrm{a.n.}}(T)$, and CTL$^+_{\mathrm{mon}}(T)$ as those fragments of CTL$^+(T)$ that disallow temporal operators in the scope of negations, contain negation signs only directly in front of atomic propositions, and do not contain negation signs at all, respectively.

In contrast to CTL, model checking for CTL$^+$ is not tractable, but Δ_2^{p}-complete [LMS01]. Below we classify the complexity of model checking for both the full and the positive fragments of CTL$^+$.

Theorem 4.2. *Let T be a set of temporal operators containing at least one path quantifier. Then CTL$^+$-MC(T) is*

– NC^1-*complete if $T \subseteq \{\mathsf{A}, \mathsf{E}\}$,*
– P-*complete if $\{\mathsf{X}\} \subsetneq T \subseteq \{\mathsf{A}, \mathsf{E}, \mathsf{X}\}$, and*
– Δ_2^{p}-*complete otherwise.*

Proof. If $T \subseteq \{\mathsf{A}, \mathsf{E}\}$ then deciding CTL$^+$-MC(T) is equivalent to the problem of evaluating a propositional formula, which is known to be NC^1-complete [Bus87, Sch10].

If $\{\mathsf{X}\} \subsetneq T \subseteq \{\mathsf{A}, \mathsf{E}, \mathsf{X}\}$, then CTL$^+$-MC$(T)$ can be solved using a labelling algorithm: Let $K = (W, R, \eta)$ be a Kripke structure, and φ be a CTL$^+(\{\mathsf{A}, \mathsf{E}, \mathsf{X}\})$-formula. Assume w.l.o.g. that φ starts with an E and that it does not contain any A's. Compute $K, w \models \psi$ for all $w \in W$ and all subformulae $\mathsf{E}\psi$ of φ such that ψ is free of path quantifiers, and replace

$\mathsf{E}\psi$ in φ with a new proposition p_ψ while extending the labelling function η such that $p_\psi \in \eta(w) \iff K, w \models \psi$. Repeat this step until φ is free of path quantifiers and denote the resulting (propositional) formula by φ'. To decide whether $K, w \models \varphi$ for some $w \in W$, it now suffices to check whether φ' is satisfied by the assignment implied by $\eta(w)$. As for all of the above subformulae $\mathsf{E}\psi$ of φ, $\psi \in \mathrm{CTL}^+(\{\mathsf{X}\})$, it follows that $K, w \models \psi$ can be determined in polynomial time in the size of K and ψ. Considering that the number of labelling steps is at most $O(|\varphi| \cdot |W|)$ it follows that $\mathrm{CTL}^+\text{-MC}(T)$ is in P. The P-hardness follows from $\mathrm{CTL}\text{-MC}(\{\mathsf{EX}\}) \leq_{\mathrm{cd}} \mathrm{CTL}^+\text{-MC}(\{\mathsf{E},\mathsf{X}\})$ resp. $\mathrm{CTL}\text{-MC}(\{\mathsf{AX}\}) \leq_{\mathrm{cd}} \mathrm{CTL}^+\text{-MC}(\{\mathsf{A},\mathsf{X}\})$.

For all other possible sets T, we have $T \cap \{\mathsf{E},\mathsf{A}\} \neq \emptyset$ and $T \cap \{\mathsf{F},\mathsf{G},\mathsf{U}\} \neq \emptyset$. Consequently, each of the temporal operators A, E, F, and G can be expressed in $\mathrm{CTL}^+(T)$. The claim now follows from [LMS01]. $\qquad\square$

For the positive fragments of CTL^+ we obtain a more complex picture:

Theorem 4.3. *Let T be a set of temporal operators containing at least one path quantifier. Then $\mathrm{CTL}^+_{\mathrm{pos}}\text{-MC}(T)$ is*

- NC^1-*complete if $T \subseteq \{\mathsf{A},\mathsf{E}\}$,*
- LOGCFL-*complete if $T = \{\mathsf{A},\mathsf{X}\}$ or $T = \{\mathsf{E},\mathsf{X}\}$,*
- P-*complete if $T = \{\mathsf{A},\mathsf{E},\mathsf{X}\}$,*
- NP-*complete if $\mathsf{E} \in T$, $\mathsf{A} \notin T$ and T contains a pure temporal operator aside from X,*
- coNP-*complete if $\mathsf{A} \in T$, $\mathsf{E} \notin T$ and T contains a pure temporal operator aside from X, and*
- Δ^{p}_2-*complete otherwise.*

Proof. The first and third claim follow from Theorem 4.2 and the monotone formula value problem being NC^1-complete [Sch10].

For the second claim, consider the case $T = \{\mathsf{E},\mathsf{X}\}$. It is straightforward to adopt Algorithm 1 to guess a successor w' of the current state once for every path quantifier E that has been read and decompose the formula w.r.t. w'. For $T = \{\mathsf{A},\mathsf{X}\}$ analogous arguments hold.

The fourth claim can be solved with a labelling algorithm analogously to the algorithm for $\mathrm{CTL}^+\text{-MC}(\{\mathsf{A},\mathsf{E},\mathsf{X}\})$. In this case, however, whole paths need to be guessed in the Kripke structures. Hence, we obtain a polynomial time algorithm deciding $\mathrm{CTL}^+_{\mathrm{pos}}\text{-MC}(T)$ using an oracle $B \in \mathrm{NP}$ (resp. $B \in \mathrm{coNP}$) . This algorithm is furthermore a monotone $\leq^{\mathrm{P}}_{\mathrm{T}}$-reduction from $\mathrm{CTL}^+_{\mathrm{pos}}\text{-MC}(T)$ to B, in the sense that for any deterministic oracle Turing machine M that executes the algorithm,

$$A \subseteq B \implies L(M, A) \subseteq L(M, B),$$

where $L(M, X)$ is the language recognized by M with oracle X. Both NP and coNP are closed under monotone $\leq^{\mathrm{P}}_{\mathrm{T}}$-reductions [Sel82]. We thus conclude that $\mathrm{CTL}^+_{\mathrm{pos}}\text{-MC}(T) \in \mathrm{NP}$ (resp. $\mathrm{CTL}^+_{\mathrm{pos}}\text{-MC}(T) \in \mathrm{coNP}$).

As for the NP-hardness of $\mathrm{CTL}^+_{\mathrm{pos}}\text{-MC}(T)$, note that the reduction from 3SAT to $\mathrm{LTL}\text{-MC}(\{\mathsf{F}\})$, the model checking problem for linear temporal logic using the F-operator only, given by Sistla and Clarke in [SC85] is a reduction to $\mathrm{CTL}^+_{\mathrm{pos}}\text{-MC}(\{\mathsf{E},\mathsf{F}\})$ indeed. The NP-hardness of $\mathrm{CTL}^+_{\mathrm{pos}}\text{-MC}(\{\mathsf{E},\mathsf{G}\})$ is obtained by a similar reduction: let φ be a propositional formula in 3CNF, *i.e.*, $\varphi = \bigwedge_{i=1}^{n} C_i$ with $C_i = \ell_{i1} \vee \ell_{i2} \vee \ell_{i3}$ and $\ell_{ij} = x_k$ or $\ell_{ij} = \neg x_k$ for all $1 \leq i \leq n$, all $1 \leq j \leq 3$, and some $1 \leq k \leq m$. Recall that for a set A, $\bigvee A$ denotes the disjunction $\bigvee_{a \in A} a$. We map φ to the triple (K, y_0, ψ), where
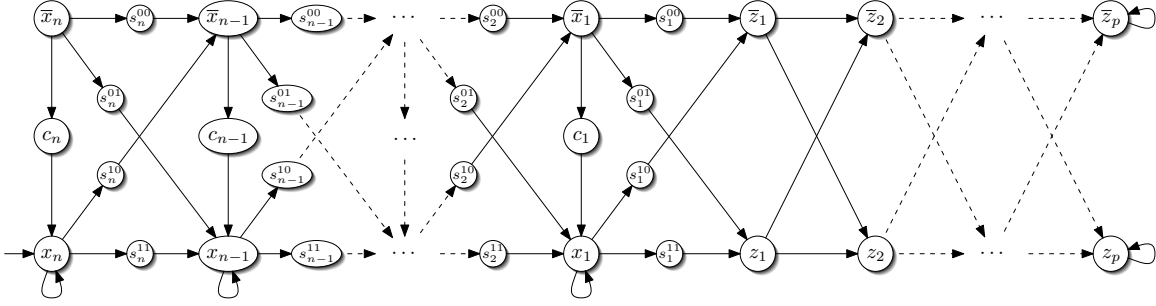
Figure 3: Extended version of the Kripke structure constructed in [LMS01, Figure 3].

$K = (W, R, \eta)$ is the Kripke structure given in (4.1) and $\psi := \mathsf{E} \bigwedge_{i=1}^{n} \bigvee_{j=1}^{3} \mathsf{G} \bigvee (\Phi \setminus \{\sim\ell_{ij}\})$ with $\Phi := \{y_0, y_i, x_i, \overline{x}_i \mid 1 \leq i \leq m\}$ and $\sim\ell_{ij}$ denoting the complementary literal of ℓ_{ij}.

$$W := \{y_0\} \cup \{x_i, \overline{x}_i, y_i \mid 1 \leq i \leq m\},$$
$$R := \{(y_{i-1}, x_i), (x_i, y_i), (y_{i-1}, \overline{x}_i), (\overline{x}_i, y_i) \mid 1 \leq i \leq m\} \cup \{(y_m, y_m)\}, \qquad (4.1)$$
$$\eta(w) := \{w\} \text{ for all } w \in W.$$

Note that the above reductions prove hardness for $\mathrm{CTL}^+_{\mathrm{mon}}\text{-}\mathrm{MC}(T)$ already. The coNP-hardness of $\mathrm{CTL}^+_{\mathrm{pos}}\text{-}\mathrm{MC}(\{\mathsf{A}, \mathsf{G}\})$ and $\mathrm{CTL}^+_{\mathrm{pos}}\text{-}\mathrm{MC}(\{\mathsf{A}, \mathsf{F}\})$ follows from the same reductions.

As for the the last claim, note that the Δ^p_2-hardness of $\mathrm{CTL}^+\text{-}\mathrm{MC}(\{\mathsf{A}, \mathsf{E}, \mathsf{F}, \mathsf{G}\})$ carries over to $\mathrm{CTL}^+_{\mathrm{mon}}\text{-}\mathrm{MC}(\{\mathsf{A}, \mathsf{E}, \mathsf{F}, \mathsf{G}\})$, because any $\mathrm{CTL}^+(\{\mathsf{A}, \mathsf{E}, \mathsf{F}, \mathsf{G}\})$-formula can be transformed into a $\mathrm{CTL}^+_{\mathrm{a.n.}}(\{\mathsf{A}, \mathsf{E}, \mathsf{F}, \mathsf{G}\})$-formula, in which all negated atoms $\neg p$ may be replaced by fresh propositions $\overline{p}$ that are mapped into all states of the Kripke structure whose label does not contain p. It thus remains to prove the Δ^p_2-hardness of $\mathrm{CTL}^+_{\mathrm{pos}}\text{-}\mathrm{MC}(\{\mathsf{A}, \mathsf{E}, \mathsf{F}\})$ and $\mathrm{CTL}^+_{\mathrm{pos}}\text{-}\mathrm{MC}(\{\mathsf{A}, \mathsf{E}, \mathsf{G}\})$. Consider $\mathrm{CTL}^+_{\mathrm{pos}}\text{-}\mathrm{MC}(\{\mathsf{A}, \mathsf{E}, \mathsf{G}\})$. Laroussinie *et al.* reduce from SNSAT, that is the problem to decide, given disjoint sets $Z_1, \ldots, Z_n$ of propositional variables from $\{z_1, \ldots, z_p\}$ and a list $\varphi_1(Z_1), \varphi_2(x_1, Z_2), \ldots, \varphi_n(x_1, \ldots, x_n, Z_n)$ of formulae in conjunctive normal form, whether x_n holds in the unique valuation σ defined by

$$\sigma(x_i) = \top \iff \varphi_i(x_1, \ldots, x_{i-1}, Z_i) \text{ is satisfiable.} \qquad (4.2)$$

An instance I of SNSAT is transformed to the Kripke structure K depicted in Figure 3 and the formula ψ_{2n-1} that is recursively defined as

$$\psi_k := \mathsf{E}\Bigg[\underbrace{\mathsf{G}\Big(\bigvee_{i=1}^{n} \overline{x}_i \to \mathsf{E}(\neg\mathsf{F} \bigvee_{i=1}^{n} (s_i^{00} \vee s_i^{01} \vee s_i^{10} \vee s_i^{11}) \wedge \mathsf{F}(\bigvee_{i=1}^{n} x_i \not\to \psi_{k-1})) \Big)}_{(A)}$$
$$\wedge \underbrace{\mathsf{G}\Big(\bigwedge_{i=1}^{n} \neg c_i \Big)}_{(B)} \wedge \underbrace{\bigwedge_{i=1}^{n} \Big(\mathsf{F}x_i \to \bigwedge_{j} \bigvee_{m} \mathsf{F}\ell_{i,j,m} \Big)}_{(C)} \Bigg],$$

for $1 \leq k \leq n$, $\psi_0 := \top$, and $\varphi_i = \bigwedge_j \bigvee_m \ell_{i,j,m}$, where the $\ell_{i,j,m}$'s are literals over $\{x_1, \ldots, x_n\} \cup Z_i$. Note that the structure K from Figure 3 differs from the Kripke structure constructed in [LMS01] in that we introduce different labels c_i and s_i^j for $1 \leq i \leq n$ and $j \in \{00, 01, 10, 11\}$, as we need to distinguish between the states later on. The intuitive

interpretation of (B) is that the existentially quantified path does actually encode an assignment of $\{x_1, \ldots, x_n\}$ to $\{\bot, \top\}$, while (C) states that this assignment coincides with σ on all propositions that are set to $\top$. Lastly (A) expresses the recursion inherent in the definition of SNSAT. It holds that $I \in \text{SNSAT} \iff K, x_n \models \psi_{2n-1}$ (see [LMS01] for the correctness of this argument).

We modify the given reduction to not use F. First note that ψ_{k-1} occurs negatively in ψ_k. We will therefore consider the formulae $\psi_{2n-1}, \psi_{2n-3}, \ldots, \psi_1$ and $\neg\psi_{2n-2}, \neg\psi_{2n-4}, \ldots, \neg\psi_2$ separately. In $\psi_{2n-1}, \psi_{2n-3}, \ldots, \psi_1$ replace

- (A) with $\mathsf{G}\Big(\bigvee_{i=1}^{n} \overline{x}_i \to \mathsf{E}\big(\mathsf{G}\bigwedge_{i=1}^{n}(\neg s_i^{00} \wedge \neg s_i^{01} \wedge \neg s_i^{10} \wedge \neg s_i^{11}) \wedge \mathsf{G}(\bigvee_{i=1}^{n} \overline{x}_i \vee \bigvee_{i=1}^{n} c_i \vee \neg\psi_{k-1}))\Big)$,

- (C) with $\bigwedge_{i=1}^{n} \Big(\mathsf{G}\neg x_i \vee \bigwedge_{j}\bigvee_{m} \mathsf{G}\bigvee(\Phi \setminus \{\sim\ell_{i,j,m}\})\Big)$;

and in $\neg\psi_{2n-2}, \neg\psi_{2n-4}, \ldots, \neg\psi_2$ replace

- (A) with $\bigvee_{1 \le i \le n} \mathsf{G}\Big(\bigvee(\Phi \setminus \{\overline{x}_i\}) \vee \mathsf{A}\big(\mathsf{G}(\bigvee \Phi \setminus \{c_i\}) \vee \mathsf{G}(c_i \vee \psi_{k-1})\big)\Big)$,

- (B) with $\bigvee_{i=2}^{n} \mathsf{G}\bigvee(\Phi \setminus \{s_i^{00}, s_i^{01}, s_{i-1}^{01}, s_{i-1}^{11}\})$, and

- (C) with $\bigvee_{i=1}^{n} \Big(\mathsf{G}\bigvee(\Phi \setminus \{\overline{x}_i\}) \wedge \bigvee_{j}\bigwedge_{m}\mathsf{G}\neg\ell_{i,j,m}\Big)$,

where $\Phi := \{x_i, \overline{x}_i, c_i, s_i^{00}, s_i^{01}, s_i^{10}, s_i^{11} \mid 1 \le i \le n\} \cup \{z_i, \overline{z}_i \mid 1 \le i \le p\}$ is the set of all propositions used in K. Denote the resulting formulae by ψ'_k, $k \ge 0$. In ψ'_k, all negations are atomic and only the temporal operators E, A and G are used.

To verify that $K, x_k \models \psi_k \iff K, x_k \models \psi'_k$ for all $0 \le k < 2n$, consider ψ_k with k odd first. Suppose $K, x_k \models \psi_k$. Then, by (A), there exists a path π in K such that whenever some $\overline{x}_i$ is labelled in the current state π_p, then there exists a path π' starting in π_p that never visits any state labelled with s_i^j, $1 \le i \le n$, $j \in \{00, 01, 10, 11\}$, and eventually falsifies ψ_{k-1} because it reaches a state where neither $\overline{x}_i$ nor c_i holds for all $1 \le i \le n$. Hence, by construction of K, π' has to visit the states labelled with c_i and x_i for i such that $\overline{x}_i \in \eta(\pi_p)$. This is equivalent to the existence of a path π' starting in π_p which never visits any state labelled with s_i^j, $1 \le i \le n$, $j \in \{00, 01, 10, 11\}$, and that falsifies ψ_{k-1} if the current state is not labelled with c_i or $\overline{x}_i$ for all $1 \le i \le n$. Hence the substitution performed on (A) does not alter the set of states in K on which the formula is satisfied.

The formula (C), on the other hand, states that whenever the path π quantified by the outmost E in ψ_k visits the state labelled x_i, then for every clause j in the ith formula φ_i of given SNSAT instance at least one literal $\ell_{i,j,m}$ occurs in the labels on π (*i.e.*, φ_i is satisfied by the assignment induced by π). The path π is guaranteed to visit either a state labelled x_i or a state labelled $\overline{x}_i$ but not both, by virtue of the subformula (B). Therefore, the eventual satisfaction of x_i is equivalent to globally satisfying $\neg x_i$, whereas the satisfaction of φ_i can be asserted by requiring that for any clause some literal is globally absent from the labels on π. Thus the substitution performed on (C) does not alter the set of states on which the formula is satisfied either. Concluding, $K, x_k \models \psi_k \iff K, x_k \models \psi'_k$ for all odd $0 \le k < 2n$.

Now, if k is even, then

$$\neg\psi_k \equiv \mathsf{A}\Big[\underbrace{\mathsf{F}\Big(\bigvee_{i=1}^n \overline{x}_i \wedge \mathsf{A}\big(\mathsf{F}\bigvee_{i=1}^n (s_i^{00} \vee s_i^{01} \vee s_i^{10} \vee s_i^{11}) \vee \mathsf{G}(\bigvee_{i=1}^n x_i \to \psi_{k-1}))\Big)}_{(A)}$$

$$\vee \underbrace{\mathsf{F}\Big(\bigvee_{i=1}^n c_i\Big)}_{(B)} \vee \underbrace{\bigvee_{i=1}^n \Big(\mathsf{F}x_i \wedge \bigvee_j \bigwedge_m \mathsf{G}\neg\ell_{i,j,m}\Big)}_{(C)}\Big].$$

Here, (A) asserts that on all paths π there is a state π_p such that $\overline{x}_i \in \eta(\pi_p)$ for some $1 \leq i \leq n$ and all paths π' starting in π_p eventually visit a state labelled with s_i^j, $1 \leq i \leq n$, $j \in \{00, 01, 10, 11\}$, or satisfy ψ_{k-1} whenever $x_i \in \eta(\pi_p)$ for some $1 \leq i \leq n$. By construction of K, this is equivalent to stating the all paths π' either pass the state labelled c_i and globally satisfy $c_i \vee \psi_{k-1}$ or do not pass the state labelled c_i. As for the states in K the formula $\mathsf{F}\big(\bigvee_{i=1}^n \overline{x}_i \wedge \chi\big) \equiv \bigvee_{i=1}^n \mathsf{F}\big(\overline{x}_i \wedge \chi\big)$ is satisfied iff $\bigvee_{i=1}^n \mathsf{G}\big(\bigvee(\Phi \setminus \{\overline{x}_i\}) \vee \chi\big)$ is satisfied, the set of states in K on which the ψ_k is satisfied remains unaltered when substituting (A) with $\bigvee_{1 \leq i < n} \mathsf{G}\big(\bigvee(\Phi \setminus \{\overline{x}_i\}) \vee \mathsf{A}\big(\mathsf{G}(\bigvee \Phi \setminus \{c_i\}) \vee \mathsf{G}(c_i \vee \psi_{k-1})\big)\big)$.

Similarly, the set of states in K on which the ψ_k is satisfied remains unaltered when substituting (B) with $\bigvee_{i=2}^n \mathsf{G}\bigvee(\Phi \setminus \{s_i^{00}, s_i^{01}, s_{i-1}^{01}, s_{i-1}^{11}\})$, as any path in K that visits a state labelled with some c_i cannot pass via states labelled with s_i^{00}, s_i^{01}, s_{i-1}^{01}, or s_{i-1}^{11}.

Finally, the equivalence of ψ_k with $\bigvee_{i=1}^n \big(\mathsf{G}\bigvee(\Phi \setminus \{\overline{x}_i\}) \wedge \bigvee_j \bigwedge_m \mathsf{G}\neg\ell_{i,j,m}\big)$ follows from arguments similar to those for the (C) part in the case that k is odd. We conclude that $K, x_k \models \psi_k \iff K, x_k \models \psi'_k$ for all $0 \leq k < 2n$. Hence, $\mathrm{CTL}_{\mathrm{pos}}^+$-$\mathrm{MC}(\{\mathsf{A}, \mathsf{E}, \mathsf{G}\})$ is Δ_2^{p}-hard.

For $T = \{\mathsf{A}, \mathsf{E}, \mathsf{F}\}$ similar modifications show that $\mathrm{CTL}_{\mathrm{pos}}^+$-$\mathrm{MC}(T)$ is Δ_2^{p}-hard, too. This concludes to proof of Theorem 4.3. □

Finally consider ECTL^+, the combination of ECTL and CTL^+. One can adapt the above hardness and membership proofs to hold for $\overset{\infty}{\mathsf{F}}$ and $\overset{\infty}{\mathsf{G}}$ instead of F and G: For example, to establish the Δ_2^{p}-hardness of $\mathrm{ECTL}_{\mathrm{pos}}^+$-$\mathrm{MC}(T)$ in case $T = \{\mathsf{A}, \mathsf{E}, \overset{\infty}{\mathsf{G}}\}$ we modify K such that the states labelled x_n and $\overline{x}_n$ are reachable from z_p and $\overline{z}_p$ and assert that (a) the path quantified by the outmost path quantifier in ψ_k, $1 \leq i < 2n$, additionally satisfies $\bigwedge_{i=1}^n (\overset{\infty}{\mathsf{G}}\neg x_i \vee \overset{\infty}{\mathsf{G}}\neg\overline{x}_i)$ and (b) whenever $\overline{x}_i$ is labelled, then there exists a path that all but a finite number of times satisfies x_i. The changes if $\overset{\infty}{\mathsf{F}}$ is available instead of $\overset{\infty}{\mathsf{G}}$ follow by the duality principle of these operators. For its model checking problem we hence obtain:

Corollary 4.4. *Let T be a set of temporal operators containing at least one path quantifier and let T' by obtained from T by substituting $\overset{\infty}{\mathsf{F}}$ with F and $\overset{\infty}{\mathsf{G}}$ with G. Then ECTL^+-$\mathrm{MC}(T) \equiv_{\mathrm{cd}} \mathrm{CTL}^+$-$\mathrm{MC}(T')$ and $\mathrm{ECTL}_{\mathrm{pos}}^+$-$\mathrm{MC}(T) \equiv_{\mathrm{cd}} \mathrm{CTL}_{\mathrm{pos}}^+$-$\mathrm{MC}(T)$.*

5. Conclusion

We have shown (Theorem 3.2) that model checking for $\mathrm{CTL}_{\mathrm{pos}}(T)$ is already P-complete for most fragments of CTL. Only for some weak fragments, model checking becomes easier: if $T \subseteq \{\mathsf{EX}, \mathsf{EF}\}$ or $T \subseteq \{\mathsf{AX}, \mathsf{AG}\}$, then $\mathrm{CTL}_{\mathrm{pos}}$-$\mathrm{MC}(T)$ is LOGCFL-complete. In the case that no CTL-operators are used, NC^1-completeness of evaluating propositional formulae
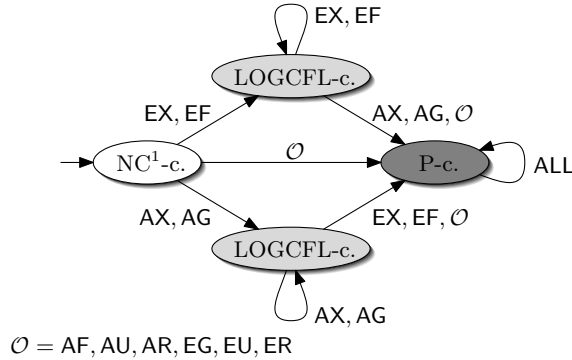
Figure 4: Complexity of $\mathrm{CTL_{pos}\text{-}MC}(T)$ for all sets T of CTL-operators (depicted as a "finite automaton" where states indicate completeness results and arrows indicate an increase of the set of CTL-operators).

applies. As a direct consequence (Theorem 3.1), model checking for $\mathrm{CTL}(T)$ is P-complete for every nonempty T. This shows that for the majority of interesting fragments, model checking $\mathrm{CTL}(T)$ is inherently sequential and cannot be sped up using parallelism.

While all the results above can be transferred to ECTL (Theorem 4.1), CTL^+ and ECTL^+ exhibit different properties. For both logics, the general model checking problem was shown to be complete for Δ_2^p in [LMS01]. Here we proved that model checking fragments of $\mathrm{CTL}^+(T)$ and $\mathrm{ECTL}^+(T)$ for $T \subseteq \{\mathsf{A}, \mathsf{E}, \mathsf{X}\}$ remains tractable, while the existential and the universal fragments of $\mathrm{CTL}_{\mathrm{pos}}^+(T)$ and $\mathrm{ECTL}_{\mathrm{pos}}^+(T)$ containing temporal operators other than X are complete for NP and coNP, respectively.

Instead of restricting only the use of negation as done in this paper, one might go one step further and restrict the allowed Boolean connectives in an arbitrary way. One might, *e.g.*, allow the exclusive-OR as the only propositional connective. This has been done for the case of linear temporal logic LTL in [BMS$^+$09], where the complexity of LTL-MC(T, B) for an arbitrary set T of temporal operators and B of propositional connectives was studied. For example, restricting the Boolean connectives to only one of the functions AND or OR leads to many NL-complete fragments in the presence of certain sets of temporal operators. However a full classification is still open.

Considering the CTL variants considered in this paper, plus $\mathrm{CTL}^\star$, over *arbitrary* sets of Boolean operators would be one way to generalise our results. In the case of CTL^+ and $\mathrm{CTL}^\star$, where model checking is intractable [EL87, Sch03, LMS01], such a more fine-grained complexity analysis could help draw a tighter border between fragments with tractable and intractable model checking problems. As for the corresponding satisfiability problems CTL-SAT(T, B) and CTL*-SAT(T, B), their complexity has been determined—with respect to the set of Boolean operators, completely—in [MMTV09].

Throughout this paper, we have assumed that the formula *and* the Kripke structure are part of the input and can vary in size. The case where the complexity is measured in terms of the size of the formula (or the Kripke structure), and the other component is assumed to be fixed, is usually referred to as *specification complexity* (or *system complexity*). Our approach measures the joint complexity. In applications, where usually the structure is significantly bigger than the specification, an analysis of the system complexity becomes

interesting. For system complexity, model checking for CTL and CTL* is already NL-complete [BVW94, KVW00]. Still, the hope for a significant drop of system complexity justifies a systematic analysis of fragments of these logics.

REFERENCES

[BMM⁺09] O. Beyersdorff, A. Meier, M. Mundhenk, T. Schneider, M. Thomas, and H. Vollmer. Model checking CTL is almost always inherently sequential. In *Proc. 16th International Symposium on Temporal Representation and Reasoning*. IEEE Computer Society Press, 2009.

[BMS⁺09] M. Bauland, M. Mundhenk, T. Schneider, H. Schnoor, I. Schnoor, and H. Vollmer. The tractability of model checking for LTL: the good, the bad, and the ugly fragments. In *Proc. 5th Methods for Modalities*, volume 231 of *Electronic Notes in Theoretical Computer Science*, pages 277–292, 2009.

[Bus87] S. R. Buss. The Boolean formula value problem is in ALOGTIME. In *Proc. 19th Symposium on Theory of Computing*, pages 123–131. ACM Press, 1987.

[BVW94] O. Bernholtz, M. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking (extended abstract). In *Proc. 6th International Conference on Computer Aided Verification*, volume 818 of *Lecture Notes in Computer Science*, pages 142–155. Springer, 1994.

[CES86] E. Clarke, E. A. Emerson, and A. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[EC82] E. A. Emerson and E. M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Science of Computer Programming*, 2(3):241–266, 1982.

[EH85] E. A. Emerson and J. Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences*, 30(1):1–24, 1985.

[EH86] E. A. Emerson and J. Y. Halpern. "Sometimes" and "not never" revisited: on branching versus linear time temporal logic. *Journal of the ACM*, 33(1):151–178, 1986.

[EL87] E. A. Emerson and C.-L. Lei. Modalities for model checking: Branching time logic strikes back. *Science of Computer Programming*, 8(3):275–306, 1987.

[KF09] L. Kuhtz and B. Finkbeiner. LTL Path Checking is Efficiently Parallelizable. *International Colloquium on Automata, Languages and Programming*, 2009.

[KVW00] O. Kupferman, M. Y. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–360, 2000.

[Lar95] F. Laroussinie. About the expressive power of CTL combinators. *Information Processing Letters*, 54(6):343–345, 1995.

[LMS01] F. Laroussinie, N. Markey, and P. Schnoebelen. Model checking CTL⁺ and FCTL is hard. In *Proc. 4th Foundations of Software Science and Computation Structure*, volume 2030 of *Lecture Notes in Computer Science*, pages 318–331. Springer Verlag, 2001.

[Mar04] N. Markey. Past is for free: on the complexity of verifying linear temporal properties with past. *Acta Informatica*, 40(6-7):431–458, 2004.

[MMTV09] A. Meier, M. Mundhenk, M. Thomas, and H. Vollmer. The complexity of satisfiability for fragments of CTL and CTL*. *International Journal of Foundations of Computer Science*, 20(5):901–918, 2009.

[Pap94] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, MA, 1994.

[Pnu77] A. Pnueli. The temporal logic of programs. In *Proc. 18th Symposium on Foundations of Computer Science*, pages 46–57. IEEE Computer Society Press, 1977.

[Ruz80] W. L. Ruzzo. Tree-size bounded alternation. *Journal of Computer and System Sciences*, 21:218–235, 1980.

[RV97] K. Regan and H. Vollmer. Gap-languages and log-time complexity classes. *Theoretical Computer Science*, 188:101–116, 1997.

[SC85] A. Sistla and E. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749, 1985.

[Sch03] Ph. Schnoebelen. *The Complexity of Temporal Logic Model Checking*, volume 4 of *Advances in Modal Logic*, pages 393–436. King's College Publications, 2003.

[Sch10]   H. Schnoor. The complexity of model checking for Boolean formulas. *Int. Journal on Foundations of Computer Science*, 21(3):289–309, 2010.

[Sel82]   A. L. Selman. Reductions on NP and P-selective sets. *Theoretical Computer Science*, 19:287–304, 1982.

[Vol99]   H. Vollmer. *Introduction to Circuit Complexity – A Uniform Approach*. Texts in Theoretical Computer Science. Springer Verlag, Berlin Heidelberg, 1999.