This is an author produced version of *Model Checking CTL is Almost Always Inherently Sequential*.

White Rose Research Online URL for this paper: http://eprints.whiterose.ac.uk/74437/

**Article:**

http://dx.doi.org/10.2168/LMCS-7(2:12)2011
MODEL CHECKING CTL IS ALMOST ALWAYS INHERENTLY
SEQUENTIAL

OLAF BEYERSDORFF\textsuperscript{a}, ARNE MEIER\textsuperscript{b}, MARTIN MUNDHENK\textsuperscript{c}, THOMAS SCHNEIDER\textsuperscript{d},
MICHAEL THOMAS\textsuperscript{e}, AND HERIBERT VOLLMER\textsuperscript{f}

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

\textsuperscript{c} Computer Science, University of Jena, Germany
e-mail address: martin.mundhenk@uni-jena.de

\textsuperscript{d} Computer Science, Saarland University, Germany
e-mail address: schneider@ps.uni-saarland.de

Revision Note. This is a revised and corrected version of the article originally published on May 17, 2011.

ABSTRACT. The model checking problem for CTL is known to be P-complete (Clarke, Emerson, and Sistla (1986), see Schobbelen (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\textsuperscript{+}, and ECTL\textsuperscript{++}.

1998 ACM Subject Classification: D.2.4, F.3.1, I.2.2, I.2.4.

Key words and phrases: Model checking, temporal logic, complexity.

\textsuperscript{*} A preliminary version of this paper appeared in the proceedings of the conference TIME’09 [BMM\textsuperscript{+}09].

Supported in part by grants DFG VO 630/6-1, VO 630/6-2, DAAD-ARC D/08/08881, and BC-ARC 1323.
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\(^{\star}\) [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\(^{\star}\) (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 $\Delta^p_2$-complete [LMS01]. We show that for $T \subseteq \{A, E, X\}$, both model checking problems restricted to operators from $T$ remain tractable, while for $T \not\subseteq \{A, E, X\}$, they become $\Delta^p_2$-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 $\Phi$ be a finite set of atomic propositions. The symbols used are the atomic propositions in $\Phi$, the constant symbols $\top$, $\bot$, the Boolean connectives $\neg$, $\land$, and $\lor$, and the temporal operator symbols $\forall$, $\exists$, $\X$, $\F$, $\G$, $\U$, $\R$.

$\forall$ and $\exists$ are called a path quantifiers, temporal operators aside from $\forall$ and $\exists$ 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 $\varphi, \psi$ are state formulae and $\chi, \pi$ are path formulae, then $\neg \varphi$, $\varphi \land \psi$, $\varphi \lor \psi$, $\forall \chi \in \Phi$, $\exists \chi \in \Phi$, $\X \chi$, $\F \chi$, $\G \chi$, $[\chi \U \pi]$, and $[\chi \R \pi]$ 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 $\eta: W \rightarrow \mathfrak{P}(\Phi)$ 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 $\varphi, \psi$ be state formulae and $\chi, \pi$ be path formulae. The truth of a CTL$^*$-formula w.r.t. $K$ is inductively defined as follows:

\begin{align*}
K, w &\models \top \quad \text{always}, \\
K, w &\models \bot \quad \text{never}, \\
K, w &\models p \quad \text{iff } p \in \Phi \text{ and } p \in \eta(w), \\
K, w &\models \neg \varphi \quad \text{iff } K, w \not\models \varphi, \\
K, w &\models (\varphi \land \psi) \quad \text{iff } K, w \models \varphi \text{ and } K, w \models \psi, \\
K, w &\models (\varphi \lor \psi) \quad \text{iff } K, w \models \varphi \text{ or } K, w \models \psi, \\
K, x &\models \forall \chi \quad \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 \quad \text{iff } K, x_1 \models \varphi, \\
K, x &\models \neg \chi \quad \text{iff } K, x \not\models \chi, \\
K, x &\models (\chi \land \pi) \quad \text{iff } K, x \models \chi \text{ and } K, x \models \pi,
\end{align*}
Figure 1: The expressive power of CTL($T$).

\[ K, x \models (\chi \lor \pi) \text{ iff } K, x \models \chi \text{ or } K, x \models \pi, \]
\[ K, x \models X\chi \text{ iff } K, x_2 \models \chi \]
\[ K, x \models [\chi U\pi] \text{ iff there is a } k \in \mathbb{N} \text{ such that } K, x^k \models \pi \text{ and } K, x^i \models \chi \text{ for } 1 \leq i < k. \]

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

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

A CTL-\textit{formula} is a CTL\(^\ast\)-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\(^\ast\)-formulae. For example, \( \text{AGEF}_p \) is a CTL-formula, but \( \text{A}(\text{GF}_p \land \text{F}_q) \) is not. CTL is less expressive than CTL\(^\ast\) [EH85, EH86].

Pairs of path quantifiers and pure temporal operators are called CTL-\textit{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 \( A[\psi U\chi] \equiv AF\chi \land \neg E[\neg \chi U(\neg \psi \land \neg \chi)], \) and thus \( E[\psi R\chi] \equiv EG\chi \lor E[\chi U(\psi \land \chi)]. \) Hence \( \{AX, AF, AR\} \) is a minimal set of operators for CTL (in presence of all Boolean connectives), whereas \( \{AX, AG, AU\} \) is not [Lar95].

By CTL($T$) we denote the set of CTL-formulae using the connectives \( \{\land, \lor, \neg\} \) 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).

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

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

- $\text{CTL}_{\text{mon}}(T)$ (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 $\bar{L}(T)$ for $\text{CTL}_{\text{a,n.}}(T)$ and $L^+(T)$ for $\text{CTL}_{\text{pos}}(T)$.

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

*Problem:* $\mathcal{L}$-MC($T$)

*Input:* A Kripke structure $K = (W,R,\eta)$, a state $w \in W$, and an $\mathcal{L}(T)$-formula $\varphi$.

*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 $\leq_{cd}$-reductions defined as follows: A language $A$ is constant-depth reducible to $B$, $A \leq_{cd} B$, 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 $\mathcal{C} = (C_1,C_2,C_3,\ldots)$ 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, i.e., given a tuple $\langle n,g,t,h \rangle$ where $n,g,h$ are binary numbers and $t \in \{\text{AND, OR, NOT, ORACLE}\}$, $M$ 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

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

Further, for model checking, atomic negation can be eluded, that is, $\text{CTL}_{\text{a,n.}}\text{-MC}(T) \leq_{cd} \text{CTL}_{\text{mon}}\text{-MC}(T)$. 
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 \( \varphi \) be a CTL\(_{a,n}(T)\)-formula over the propositions \( \Phi = \{p_1, \ldots, p_n\} \). Every negation in \( \varphi \) appears inside a negative literal. We obtain \( \varphi' \) 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.

In Section 3.3, we complete the picture by proving CTL\(_{pos}\)-MC(\( T \)) \( \leq_{cd} \) CTL\(_{mon}\)-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, CFL \( \subseteq \) 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, \( \leq_{cd} \)-reductions are used). Hence, LOGCFL \( \subseteq \) NC (even LOGCFL \( \subseteq \) 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:

\[ \text{NC}^1 \subseteq L \subseteq \text{NL} \subseteq \text{LOGCFL} = \text{SAC}^1 \subseteq \text{NC}^2; \]

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\(_{pos}\)

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

While model checking for CTL in general is known to be polynomial time solvable and in fact P-complete [NESS86, 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 CTL_{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 CTL_{pos}-MC(\( T \)) is
- NC\(^1\)-complete if \( T = \emptyset \),
- LOGCFL-complete if \( \emptyset \subsetneq T \subseteq \{EX, EF\} \) or \( \emptyset \subsetneq T \subseteq \{AX, AG\} \), 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 [NESS86]. While this upper bound also applies to CTL_{pos}-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 \( T \subseteq \{EX, EF\} \) or \( T \subseteq \{AX, AG\} \). Then CTL_{pos}-MC(\( T \)) is in LOGCFL.

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

Algorithm 1 always terminates because each subformula of \( \varphi \) 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 \( (\psi, w) \) 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 CTL_{pos}-MC(\( T \)) is in LOGCFL.

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

Finally, for the trivial case where no CTL-operators are present, model checking CTL(\( \emptyset \))-formulae is equivalent to the problem of evaluating a propositional formula. This problem is known to be solvable in NC\(^1\) [Bus87].
Algorithm 1 Determine whether $K, w_0 \models \varphi$.

Require: a Kripke structure $K = (W, R, \eta)$, $w_0 \in W$, $\varphi \in \text{CTL}_{\text{pos}}(T)$

1: push($S$, $(\varphi, w_0)$)
2: while $S$ is not empty do
3:     $(\varphi, w) \leftarrow \text{pop}(S)$
4:     if $\varphi$ is a propositional formula then
5:         if $\varphi$ evaluates to false in $w$ under $\eta$ then
6:             return false
7:         end if
8:     else if $\varphi = \alpha \land \beta$ then
9:         push($S$, $(\beta, w)$)
10:        push($S$, $(\alpha, w)$)
11:    else if $\varphi = \alpha \lor \beta$ then
12:        nondet. push($S$, $(\alpha, w)$) or push($S$, $(\beta, w)$)
13:    else if $\varphi = \text{EX} \alpha$ then
14:        nondet. choose $w' \in \{ w' \mid (w, w') \in R \}$
15:        push($S$, $(\alpha, w')$)
16:    else if $\varphi = \text{EF} \alpha$ then
17:        nondet. choose $w' \in \{ w' \mid (w, w') \in R^* \}$
18:        push($S$, $(\alpha, w')$)
19:    end if
20: end while
21: return true

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 $\text{CTL}_{\text{mon}}\text{-MC}(T)$ is P-hard if $T$ contains an existential and a universal CTL-operator.

Proof. First, assume that $T = \{ \text{AX}, \text{EX} \}$. 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 \( W := \{c^j_i | 1 \leq i \leq q(n), 0 \leq j \leq p(n)\} \) and the relation \( R \subseteq W \times W \) as
\[
R := \{(c^j_i, c^{j+1}_i) | M \text{ reaches configuration } c_k \text{ from } c_i \text{ in one step, } 0 \leq j < p(n)\}
\cup \{(c^j_i, c^j_i) | c^j_i \text{ has no successor, } 1 \leq i \leq q(n), 0 \leq j < p(n)\}
\cup \{(c^{p(n)}_i, c^{p(n)}_i) | 1 \leq i \leq q(n)\}.
\]

The labelling function \( \eta \) is defined for all \( c^j_i \in W \) as
\[
\eta(c^j_i) := \begin{cases} 
\{t\}, & \text{if } c_i \text{ is an accepting configuration and } j = p(n) \\
\emptyset, & \text{otherwise}
\end{cases}
\]
where \( t \) is the only atom used by this labelling. It then holds that
\[
M \text{ accepts } x \iff K, c^0_i \models \psi_1(\psi_2(\cdots \psi_{p(n)}(t)) \cdots),
\]
where \( \psi_i(x) := AX(x) \) if \( M \)'s configurations before the \( i \)th step are universal, and \( \psi_i(x) := EX(x) \) 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 \( \varphi \) is polynomial in the size of \( (M, x) \). Moreover, \( K \) and \( \varphi \) can be constructed from \( M \) and \( x \) using AC\(^0\)-circuits. Thus, \( A \leq_{cd} \text{CTL}_{\text{mon}}\text{-MC} \{\{AX, EX\}\} \) for all \( A \in \text{ALOGSPACE} = \text{P} \).

For \( T = \{\text{AF}, \text{EG}\} \) we modify the above reduction by defining the labelling function \( \eta \) and the formula \( \psi_i \) as follows:
\[
\eta(c^j_i) := \begin{cases} 
\{d_j, t\}, & \text{if } c_i \text{ is an accepting configuration and } j = p(n) \\
\{d_j\}, & \text{otherwise}
\end{cases}
\]
\[
\psi_i(x) := \begin{cases} 
\text{AF}(d_i \land x), & \text{if } M \text{’s configurations before step } i \text{ are universal,} \\
\text{EG}(D_i \lor x), & \text{otherwise,}
\end{cases}
\]
where \( d_j \) are atomic propositions encoding the ‘time stamps’ of the respective configurations and \( D_i = \bigvee_{i \neq j \in \{0, \ldots, p(n)\}} d_j \).

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

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

In the presence of arbitrary negation, universal operators are definable by existential operators and vice versa. Hence, from Proposition 3.3 we obtain the following corollary.

**Corollary 3.5.** The model checking problem \( \text{CTL-MC}(T) \) 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 \( \text{CTL}_{\text{mon}}\text{-MC}(T) \) is P-hard if \( T \) contains at least one of the operators \( \text{EG}, \text{EU}, \text{ER}, \text{AF}, \text{AU}, \) or \( \text{AR} \).

**Proof.** We modify the proof of Proposition 3.4 to work with \( \text{EG} \) only. The remaining fragments follow from the closure of \( P \) under complementation and \( F\chi \equiv \neg G\neg\chi \equiv [\top U\chi], \) \( [\chi U\pi] \equiv \neg[\neg\chi R\neg\pi] \).
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. The purpose of the introduced layers below is to ensure the uniqueness of the successors of universal configurations which is essential in the construction of $\psi_i$ later. We construct a Kripke structure $K' := (W', R, \eta)$ consisting of $q(n) + 1$ layers and a ‘trap’ as follows: let $W' := W \times \{1, \ldots, q(n) + 1\} \cup \{z\}$. The transition relation $R \subseteq W' \times W'$ is defined as

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

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 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 $\eta$ 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

$$\psi_i(x) := \begin{cases} 
\text{EG}(d_{i-1} \lor (d_i \land x) \lor z), \text{if } M\text{'s configurations before step } i \text{ are universal,} \\
\text{EG}(D_i \lor x), \text{if } M\text{'s configurations before step } i \text{ are existential,}
\end{cases}$$
As we essentially only duplicated the set of states in the trap. For each of the two successor states three disjuncts ranges through both series-connected successor configurations and ends in EG be satisfied the composition of the ψ holds. Thus AF(di ∧ x) is true in the state w in the structure K.

From this, it easily follows that for

\[ M \text{ accepts } x \iff K'(v_1, 1) \models \psi_1(\psi_2(\ldots \psi_n(t)) \ldots) \].

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 AC0 constructible. Concluding A ≤cd CTL_mon-MC(\{EG\}) for all A ∈ P.

By Lemma 2.1, CTL_mon-MC(T) ≤cd CTL_pos-MC(T) and hence the above results directly translate to model checking for CTL_pos: for any set T of temporal operators, CTL_pos-MC(T) is P-hard if T ⊆ {EX, EF} or if T ⊆ {AX, AG}. These results cannot be improved w.r.t. T, as for T ⊆ {EX, EF} and T ⊆ {AX, 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 CTL_mon-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 ∨-gates with unbounded fan-in and ∧-gates with fan-in 2. For every single CTL-operator O, we will show that CTL_mon-MC(T) is LOGCFL-hard for all T ⊆ \{O\} by giving a generic ≤cd-reduction f from the word problem for SAC\(^1\) circuits to CTL_mon-MC(T).

First, consider EX ∈ T. Let C be a logtime-uniform SAC\(^1\) circuit of depth \(\ell\) 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 ∧-gates and ∨-gates, and that the output gate of C is an ∨-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 \(\ell\). Denote the graph of C by \(G = (V, E)\), where \(V := V_\text{in} \uplus V_\wedge \uplus V_\lor\) is partitioned into the sets corresponding to the (possibly negated) input-gates, the ∧-gates, and the ∨-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 ∧-gate from each other. This will be required to model proof trees using CTL_mon({EX})-formulae.
For \( i \in \{1, 2\} \), let \( V_\text{in}^i := \{ v^i \mid v \in V_\text{in} \} \), \( V_\text{out}^i := \{ v^i \mid v \in V_\text{out} \} \) and define \( V_\text{in,\,or}^i := V_\text{in}^i \cup V_\text{out}^i \).

Further define
\[
E' := \{ (v, u^i) \in V_\land \times V_\text{in,\,or}^i \mid (u, v) \in E \text{ and } u \text{ is the } \text{ith} \text{ predecessor of } v \} \\
\cup \{ (v, v) \mid v \in V_\text{in}^i \cup V_\text{out}^2 \} \cup \bigcup_{i \in \{1, 2\}} \{ (v^i, u) \in V_\text{in,\,or}^i \times V_\land \mid (u, v) \in E \},
\]
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 \( V' := V_\text{in,\,or}^1 \cup V_\text{in,\,or}^2 \cup V_\land \), transition relation \( E' \), and labelling function \( \eta : V' \to \mathfrak{P}(\{1, 2, t\}) \),
\[
\eta(v) := \begin{cases} 
\{i, t\}, & \text{if } (v = v^i_{\text{in}} \in V_\text{in}^i \text{ and } x_j = 1) \text{ or } (v = \overline{v}_\text{in}^j \in V_\text{in}^j \text{ and } x_j = 0), \\
\{i\}, & \text{if } (v = v^i_{\text{in}} \in V_\text{in}^i \text{ and } x_j = 0) \text{ or } (v = \overline{v}_\text{in}^j \in V_\text{in}^j \text{ and } x_j = 1) \text{ or } v \in V_\text{out}^j, \\
\emptyset, & \text{otherwise,}
\end{cases}
\]
where \( i = 1, 2, j = 1, \ldots, n \) and \( v_{\text{in}1}, \ldots, v_{\text{in}n}, \overline{v}_{\text{in}1}, \ldots, \overline{v}_{\text{in}n} \) enumerate the input gates and their negations. The formula \( \varphi \) that is to be evaluated on \( K \) will consist of atomic propositions \( 1, 2 \) and \( t \), Boolean connectives \( \land \) and \( \lor \), and the CTL-operator \( \text{EX} \). To construct \( \varphi \) we recursively define formulae \( \langle \varphi_i \rangle_{0 \leq i \leq \ell} \) by
\[
\varphi_i := \begin{cases} 
t, & \text{if } i = \ell, \\
\text{EX}\varphi_{i+1}, & \text{if } i \text{ is even (\lor-layers),} \\
\land_{i=1,2} \text{EX}(i \land \varphi_{i+1}), & \text{if } i \text{ is odd (\land-layers).}
\end{cases}
\]

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 \( \varphi \) 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 \( \lor \)-gate, and
3. \( K' \) includes the two successors of every node corresponding to an \( \land \)-gate.

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

Second, consider \( \text{EF} \in T \). 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 \( \eta \) to encode this distance \( i, 1 \leq i \leq \ell \), into the “depth-propositions” \( d_i \) as in the proof of Proposition 4.1. Denote this modified Kripke structure by \( K' \). Further, we define \( \langle \varphi'_i \rangle_{0 \leq i \leq \ell} \) as
\[
\varphi'_i := \begin{cases} 
t, & \text{if } i = \ell, \\
\text{EF}(d_{i+1} \land \varphi'_{i+1}), & \text{if } i \text{ is even,} \\
\land_{i=1,2} \text{EF}(d_{i+1} \land i \land \varphi'_{i+1}), & \text{if } i \text{ is odd.}
\end{cases}
\]
Redefining the reduction \( f \) as \( (C, x) \mapsto (K', v_0, \varphi'_0) \) hence yields the LOGCFL-hardness of \( \text{CTL}_{\text{mon}}\text{-MC}(T) \) for \( \text{EF} \in T \).

Third, consider \( \text{AX} \in T \). Consider the reduction in case 1 for \( \text{CTL}_{\text{mon}}(\{\text{EX}\}) \)-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 \(\varphi'\) be the formula obtained from \(\neg \varphi\) by multiplying the negation into the formula. Then \(\varphi'\) is a CTL_{a,n}(\{AX\})-formula. Since LOGCFL is closed under complement, it follows that CTL_{a,n}-MC(\{AX\}) is LOGCFL-hard. Using Lemma 2.1, we obtain that CTL_{mon}-MC(\{AX\}) is LOGCFL-hard, too. An analogous argument works for the case AG \(\in T\). The remaining fragments are even P-complete by Proposition 3.6.

Using Lemma 2.1 we obtain LOGCFL-hardness of CTL_{pos}-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 CTL_{a,n} and CTL_{pos} is computationally equivalent to model checking for CTL_{mon}, for any set \(T\) of CTL-operators. Since we consider \(\leq_{cd}\)-reductions, this is not immediate.

From Lemma 2.1 it follows that the hardness results for CTL_{mon}-MC(\(T\)) also hold for CTL_{a,n}-MC(\(T\)) and CTL_{pos}-MC(\(T\)). Moreover, the algorithms for CTL_{pos}-MC(\(T\)) also work for CTL_{mon}-MC(\(T\)) and CTL_{a,n}-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 CTL_{mon}-MC(\(T\)), CTL_{a,n}-MC(\(T\)), and CTL_{pos}-MC(\(T\)) are

- NC\(^1\) complete if \(T\) is empty,
- LOGCFL-complete if \(\emptyset \not\subseteq T \subseteq \{EX, EF\}\) or \(\emptyset \not\subseteq T \subseteq \{AX, AG\}\),
- P-complete otherwise.

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

This equivalence extends Lemma 2.1. We remark that this equivalence is not straightforward. Simply applying de Morgan’s laws to transform one problem into another requires counting the number of negations on top of \(\land\) and \(\lor\)-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.

### 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 [EHS86]. ECTL extends CTL with the \(\bar{F}\)-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 \(\chi\)

\[K, x \models \bar{F} \chi \iff K, x^i \models \chi \text{ for all } i \in \mathbb{N}.\]

The dual operator \(\bar{G}\) 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 $\text{ECTL-MC}(T) \equiv_{cd} \text{CTL-MC}(T')$ and $\text{ECTL}_{\text{pos}}\text{-MC}(T) \equiv_{cd} \text{CTL}_{\text{pos}}\text{-MC}(T')$, where $T'$ is obtained from $T$ by substituting $\text{EF}$ with $\text{F}$ and $\text{G}$ with $\text{G}$.

Proof. For the upper bounds, notice that $\text{ECTL-MC}(\text{ALL} \cup \{\text{EF}, \text{AF}\}) \in \text{P}$. It thus remains to show that $\text{ECTL}_{\text{pos}}\text{-MC}(T) \in \text{LOGCFL}$ for $T \subseteq \{\text{EX}, \text{EF}, \text{EF}\}$ and $T \subseteq \{\text{AX}, \text{AG}, \text{AG}\}$.

First, consider the case that $T \subseteq \{\text{EX}, \text{EF}, \text{EF}\}$. We modify Algorithm 1 to handle $\text{EF}$ by extending the case distinction in lines 4–19 with the code fragment given in Algorithm 1. The algorithm for $T \subseteq \{\text{AX}, \text{AG}, \text{AG}\}$ is analogous and membership in $\text{LOGCFL}$ 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 $\text{AF}$, $\text{AG}$, $\text{EF}$, and $\text{EG}$. Therefore, we only need modify the accessibility relation $R$ of respective Kripke structure $K$ to be reflexive. The hardness results follow by replacing $\text{F}$ with $\text{F}$ and $\text{G}$ with $\text{G}$ in the respective reductions.

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

$$
\psi_i(x) := \begin{cases}
\text{AF}(d_i \land x), & \text{if } M \text{’s configurations in step } i \text{ are universal}, \\
\text{EG}(D_i \lor x), & \text{otherwise},
\end{cases}
$$

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 $1 \leq i \leq p(|x|)$, $\tilde{K}, w \models \text{AF}(d_i \land x)$ iff $K, w \models \text{AF}(d_i \land x)$, and $\tilde{K}, w \models \text{EG}(\bigvee_{i \neq j \in \{0, \ldots, p(n)\}} d_j \lor x)$ iff $K, w \models \text{EG}(\bigvee_{i \neq j \in \{0, \ldots, p(n)\}} d_j \lor x)$. From this, correctness of the reduction follows. The P-hardness of $\text{CTL-MC}(T)$ for the remaining fragments follows analogously.

As for $T \subseteq \{\text{EX}, \text{EF}, \text{EF}\}$, we will show that $\text{ECTL}_{\text{mon}}\text{-MC}(T)$ is LOGCFL-hard under $\equiv_{cd}$-reductions for $T = \{\text{EF}\}$. Let $C$, $x$, and $\ell$ 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 $\text{CTL-MC}(\{\text{EF}\})$, $v_0 \in V'$, and $\varphi_0$ is recursively defined via $(\varphi_i')_{0 \leq i \leq \ell}$ as

$$
\varphi_i := \begin{cases}
t, & \text{if } i = \ell, \\
\text{EF}(d_i \land \varphi_{i+1}), & \text{if } i \text{ is even}, \\
\land_{j=1,2,3} \text{EF}(d_i \land \varphi_{i+1}), & \text{if } i \text{ is odd}.
\end{cases}
$$

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 $\tilde{K}', v \models \text{EF}(d_i \land \varphi_{i+1})$ iff $K', v \models \text{EF}(d_i \land \varphi_i)$, for all $v \in V'$ and
1 ≤ i ≤ ℓ. We conclude that ECTL_{mon}'-MC(\{EF\}) is LOGCFL-hard. The case T = \{AG\} follows analogously.

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, AGEFp and A(Gp ∧ Fq) are CTL^-formulae, but AGFp is not. However, CTL is as expressive as CTL^+ [EH85].

By CTL^+(T) we denote the set of CTL^+-formulae using the connectives \{∧, ∨, ¬\} and temporal operators in T only. Analogous to the fragments CTL_{pos}(T), CTL_{a.n.}(T), and CTL_{mon}(T), we define CTL^+_{pos}(T), CTL^+_{a.n.}(T), and CTL^+_{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 \Delta^p_2-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^+_{mon}-MC(T) is

- NC^1-complete if \{A, E\} ⊆ T,
- P-complete if \{X\} ⊆ T ⊆ \{A, E, X\}, and
- \Delta^p_2-complete otherwise.

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

If \{X\} ⊆ T ⊆ \{A, E, X\}, then CTL^+_{mon}-MC(T) can be solved using a labelling algorithm: Let \(K = (W, R, η)\) be a Kripke structure, and \(φ\) be a CTL^+({A, E, X})-formula. Assume w.l.o.g. that \(φ\) starts with an E and that it does not contain any A’s. Compute \(K, w ⊨ ψ\) for all \(w ∈ W\) and all subformulae Eψ of \(φ\) such that \(ψ\) is free of path quantifiers, and replace Eψ in \(φ\) with a new proposition \(p_ψ\) while extending the labelling function \(η\) such that \(p_ψ ∈ η(w) ↔ K, w ⊨ ψ\). Repeat this step until \(φ\) is free of path quantifiers and denote the resulting (propositional) formula by \(φ'\). To decide whether \(K, w ⊨ ψ\) for some \(w ∈ W\), it now suffices to check whether \(φ'\) is satisfied by the assignment implied by \(η(w)\). As for all of the above subformulae Eψ of \(φ, ψ ∈ CTL^+(\{X\})\), it follows that \(K, w ⊨ ψ\) can be determined in polynomial time in the size of \(K\) and \(ψ\). Considering that the number of labelling steps is at most \(O(|φ| ∗ |W|)\) it follows that CTL^+_{mon}-MC(T) is in P. The P-hardness follows from CTL-MC({EX}) \(⊆_{cd}\) CTL^+_{mon}-MC({E, X}) resp. CTL-MC({AX}) \(⊆_{cd}\) CTL^+_{mon}-MC({A, X}).

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

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 CTL^+_{pos}-MC(T) is

- NC^1-complete if \{A, E\} ⊆ T,
- LOGCFL-complete if \(T = \{A, X\}\) or \(T = \{E, X\}\),
The NP-hardness of CTL

formed into a CTL

+ reduction from CTL

A a set

∈ over to CTL

+ using an oracle

∈ arguments hold.

The fourth claim can be solved with a labelling algorithm analogously to the algorithm for CTL

+ -MC(⟨A, E, X⟩). In this case, however, whole paths need to be guessed in the Kripke structures. Hence, we obtain a polynomial time algorithm deciding CTL

+ -MC using an oracle

∈ that has been read and decompose the formula w.r.t. w'. For T = {A, X} analogous arguments hold.

Proof. The first and third claim follow from Theorem 4.2 and the monotone formula value problem being NC1-complete [Sch10].

For the second claim, consider the case T = {E, 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 = {A, X} analogous arguments hold.

As for the NP-hardness of CTL

+ -MC(T), note that the reduction from 3SAT to LTL-MC(⟨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 CTL

+ -MC(⟨E, F⟩) indeed. The NP-hardness of CTL

+ -MC(⟨E, G⟩) is obtained by a similar reduction: let \( \varphi \) be a propositional formula in 3CNF, i.e., \( \varphi = \bigwedge_{i=1}^{n} C_i \) with \( C_i = \ell_{i1} \lor \ell_{i2} \lor \ell_{i3} \) 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, \( \lor A \) denotes the disjunction \( \bigvee_{a \in A} a \). We map \( \varphi \) to the triple \( (K, y_0, \psi) \), where \( K = (W, R, \eta) \) is the Kripke structure given in (4.1) and \( \psi := E \bigwedge_{i=1}^{n} \bigvee_{j=1}^{3} G \bigvee (\Phi \setminus \{\neg \ell_{ij}\}) \) with \( \Phi := \{y_0, y_i, x_i, \overline{x}_i \mid 1 \leq i \leq m\} \) and \( \neg \ell_{ij} \) denoting the complementary literal of \( \ell_{ij} \).

\[
W := \{y_0\} \cup \{x_i, \overline{x}_i, y_i \mid 1 \leq i \leq m\},
\]

\[
R := \{(y_{i-1}, x_i), (x_i, y_i), (y_{i-1}, \overline{x}_i), (\overline{x}_i, y_i) \mid 1 \leq i \leq m\} \cup \{(y_m, y_m)\}, \tag{4.1}
\]

\( \eta(w) := \{w\} \) for all \( w \in W \).

Note that the above reductions prove hardness for CTL

+ -MC(T) already. The coNP-hardness of CTL

+ -MC(⟨A, G⟩) and CTL

+ -MC(⟨A, F⟩) follows from the same reductions.

As for the the last claim, note that the \( \Delta^p_2 \)-hardness of CTL

+ -MC(⟨A, E, F, G⟩) carries over to CTL

+ -MC(⟨A, E, F, G⟩), because any CTL

+ (⟨A, E, F, G⟩)-formula can be transformed into a CTL

+ -MC(⟨A, E, F, 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 \( \Delta^p_2 \)-hardness of CTL

+ -MC(⟨A, E, F⟩) and CTL

+ -MC(⟨A, E, G⟩). Consider CTL

+ -MC(⟨A, E, 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
We will therefore consider the formulae $\psi$ of conjunctive normal form, whether $x_n$ holds in the unique valuation $\sigma$ defined by $\sigma(x_i) = \top \iff \varphi_i(x_1, \ldots, x_{i-1}, Z_i)$ is satisfiable. (4.2)

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

$$\psi_k := \textstyle E \left[ G \left( \bigvee_{i=1}^n \varphi_i \rightarrow E \left( \bigwedge_{i=1}^n (s_{i,0}^0 \lor s_{i,0}^1 \lor s_{i,0}^2) \land F(\bigvee_{i=1}^n x_i \not\rightarrow \psi_{k-1}) \right) \right) \right]$$

$$\quad \land G \left( \bigwedge_{i=1}^n \neg c_i \right) \land \left[ \bigwedge_{i=1}^n \left( F x_i \rightarrow \bigvee_{j,m} F \ell_{i,j,m} \right) \right],$$

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 $s_{i,0}^l$ 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 $\sigma$ on all propositions that are set to $\top$. Lastly (A) expresses the recursion inherent in the definition of SNSAT. It holds that $I \in \text{SNSAT} \iff K, x_n = \psi_{2n-1}$ (see [LMS01] for the correctness of this argument).

We modify the given reduction to not use $F$. First note that $\psi_{k-1}$ occurs negatively in $\psi_k$. 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 $G \left( \bigvee_{i=1}^n \varphi_i \rightarrow E \left( G \bigwedge_{i=1}^n (s_{i,0}^0 \land s_{i,0}^1 \land s_{i,0}^2) \land \bigvee_{i=1}^n \bigvee_{c_i, \neg \psi_{k-1}} \right) \right)$,

- (C) with $\bigwedge_{i=1}^n \left( G \land \bigwedge_{j,m} F \ell_{i,j,m} \right)$;

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

- (A) with $\bigvee_{1 \leq i \leq n} G \left( \bigvee_{j,m} (\Phi \setminus \{\varphi_i\}) \lor A \left( G(\bigcup_{t \in T} \{c_t\}) \lor G(c_i \lor \psi_{k-1}) \right) \right)$.
(B) with \( \bigvee_{i=2}^{n} G \bigvee (\Phi \setminus \{s_{i}^{00}, s_{i}^{01}, s_{i-1}^{01}, s_{i-1}^{11}\}) \), and

(C) with \( \bigvee_{i=1}^{n} \left( G \bigvee (\Phi \setminus \{\pi_{i}\}) \right) \land \bigwedge_{j,m} G^{\neg \ell_{i,j,m}} \),

where \( \Phi := \{x_{i}, \pi_{i}, c_{i}, s_{i}^{00}, s_{i}^{01}, s_{i}^{10}, s_{i}^{11} | 1 \leq i \leq n \} \cup \{z_{i}, \pi_{i} | 1 \leq i \leq p \} \) is the set of all propositions used in \( K \). Denote the resulting formulae by \( \psi'_{k} \), \( k \geq 0 \). In \( \psi'_{k} \), all negations are atomic and only the temporal operators \( E, A \) and \( G \) are used.

To verify that \( K, x_{k} |\equiv \psi_{k} \iff K, x_{k} |\equiv \psi'_{k} \) for all \( 0 \leq k < 2n \), consider \( \psi_{k} \) with \( k \) odd first. Suppose \( K, x_{k} |\equiv \psi_{k} \). Then, by (A), there exists a path \( \pi \) in \( K \) such that whenever some \( \pi_{i} \) is labelled in the current state \( \pi_{p} \), then there exists a path \( \pi' \) starting in \( \pi_{p} \) that never visits any state labelled with \( s_{i}^{j} \), \( 1 \leq i \leq n \), \( j \in \{00, 01, 10, 11\} \), and eventually falsifies \( \psi_{k-1} \) because it reaches a state where neither \( \pi_{i} \) nor \( c_{i} \) holds for all \( 1 \leq i \leq n \). Hence, by construction of \( K \), \( \pi' \) has to visit the states labelled with \( c_{i} \) and \( x_{i} \) for \( i \) such that \( \pi_{i} \in \eta(\pi_{p}) \). This is equivalent to the existence of a path \( \pi' \) starting in \( \pi_{p} \) which never visits any state labelled with \( s_{i}^{j} \), \( 1 \leq i \leq n \), \( j \in \{00, 01, 10, 11\} \), and that falsifies \( \psi_{k-1} \) if the current state is not labelled with \( c_{i} \) or \( \pi_{i} \), for all \( 1 \leq i \leq n \). 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 \( \pi \) quantified by the outmost \( E \) in \( \psi_{k} \) visits the state labelled \( x_{i} \), then for every clause \( j \) in the \( i \)th formula \( \varphi_{i} \) of given SNSAT instance at least one literal \( \ell_{i,j,m} \) occurs in the labels on \( \pi \) (i.e., \( \varphi_{i} \) is satisfied by the assignment induced by \( \pi \)). The path \( \pi \) is guaranteed to visit either a state labelled \( x_{i} \) or a state labelled \( \pi_{i} \) 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 \( \varphi_{i} \) can be asserted by requiring that for any clause some literal is globally absent from the labels on \( \pi \). Thus the substitution performed on (C) does not alter the set of states on which the formula is satisfied either. Concluding, \( K, x_{k} |\equiv \psi_{k} \iff K, x_{k} |\equiv \psi'_{k} \) for all odd \( 0 \leq k < 2n \).

Now, if \( k \) is even, then

\[
\neg \psi_{k} = \left[ F \left( \bigvee_{i=1}^{n} \pi_{i} \land A \left( F \left( \bigvee_{i=1}^{n} \left( s_{i}^{00} \lor s_{i}^{01} \lor s_{i}^{10} \lor s_{i}^{11} \right) \right) \lor G \left( \bigvee_{i=1}^{n} x_{i} \rightarrow \psi_{k-1} \right) \right) \right) \right] \\
\lor \left[ F \left( \bigvee_{i=1}^{n} c_{i} \right) \lor \left( \bigvee_{i=1}^{n} \left( F x_{i} \land \bigwedge_{j,m} G^{\neg \ell_{i,j,m}} \right) \right) \right].
\]

Here, (A) asserts that on all paths \( \pi \) there is a state \( \pi_{p} \) such that \( \pi_{i} \in \eta(\pi_{p}) \) for some \( 1 \leq i \leq n \) and all paths \( \pi' \) starting in \( \pi_{p} \) eventually visit a state labelled with \( s_{i}^{j} \), \( 1 \leq i \leq n \), \( j \in \{00, 01, 10, 11\} \), or satisfy \( \psi_{k-1} \) 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 \( \pi' \) either pass the state labelled \( c_{i} \) and globally satisfy \( c_{i} \lor \psi_{k-1} \) or do not pass the state labelled \( c_{i} \). As for the states in \( K \) the formula

\[
F \left( \bigvee_{i=1}^{n} \pi_{i} \land \chi \right) \equiv \bigvee_{i=1}^{n} F \left( \pi_{i} \land \chi \right)
\]

is satisfied iff \( \bigvee_{i=1}^{n} G \left( \bigvee \left( \phi_{i} \setminus \{\pi_{i}\} \right) \right) \land \chi \) is satisfied, the set of states in \( K \) on which the \( \psi_{k} \) is satisfied remains unaltered when substituting (A) with

\[
\bigvee_{1 \leq i \leq n} G \left( \bigvee \left( \phi_{i} \setminus \{\pi_{i}\} \right) \right) \lor A \left( G \left( \bigvee \phi_{i} \setminus \{c_{i}\} \right) \lor G \left( c_{i} \lor \psi_{k-1} \right) \right).\]
Similarly, the set of states in \( K \) on which the \( \psi_k \) is satisfied remains unaltered when substituting \( (B) \) with \( \bigvee_{i=2}^{n} G \bar{\forall} \left( F \setminus \left\{ s'^{00}_i, s'^{01}_i, s'^{11}_i \right\} \right) \), as any path in \( K \) that visits a state labelled with some \( c_i \) cannot pass via states labelled with \( s'^{00}_i, s'^{01}_i, s'^{11}_i \), or \( s^{11}_i \).

Finally, the equivalence of \( \psi_k \) with \( \bigvee_{i=1}^{n} \left( G \bar{\forall} \left( F \setminus \left\{ \bar{f}_i \right\} \right) \right) \wedge \bigvee_{j=1}^{m} G \bar{\forall} \left( \bar{f}_i,j,m \right) \) 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, \( \text{CTL}_{\text{pos}}^+\text{-MC}(\{A, E, G\}) \) is \( \Delta_2^P \)-hard.

For \( T = \{A, E, F\} \) similar modifications show that \( \text{CTL}_{\text{pos}}^+\text{-MC}(T) \) is \( \Delta_2^P \)-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 \( \mathcal{F} \) and \( \mathcal{G} \) instead of \( F \) and \( G \): For example, to establish the \( \Delta_2^P \)-hardness of \( \text{ECTL}_{\text{pos}}^+\text{-MC}(T) \) in case \( T = \{A, E, G\} \) we modify \( K \) such that the states labelled \( x_n \) and \( w_n \) are reachable from \( z_p \) and \( \bar{w}_p \) and assert that (a) the path quantified by the outmost path quantifier in \( \psi_k \), \( 1 \leq i < 2n \), additionally satisfies \( \bigwedge_{i=1}^{n} (G \bar{\forall} x_i \vee \mathcal{G} \bar{\forall} \bar{w}_i) \) and (b) whenever \( \bar{w}_i \) is labelled, then there exists a path that all but a finite number of times satisfies \( x_i \). The changes if \( \mathcal{F} \) is available instead of \( G \) 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 \( T' \) be obtained from \( T \) by substituting \( \mathcal{F} \) with \( F \) and \( \mathcal{G} \) with \( G \). Then \( \text{ECTL}_{\text{pos}}^+\text{-MC}(T') \equiv_{\text{cd}} \text{CTL}_{\text{pos}}^+\text{-MC}(T) \) and \( \text{ECTL}_{\text{pos}}^+\text{-MC}(T) \equiv_{\text{cd}} \text{CTL}_{\text{pos}}^+\text{-MC}(T) \).

5. Conclusion

We have shown (Theorem 3.2) that model checking for \( \text{CTL}_{\text{pos}}(T) \) is already \( P \)-complete for most fragments of CTL. Only for some weak fragments, model checking becomes easier: if \( T \subseteq \{EX, EF\} \) or \( T \subseteq \{AX, AG\} \), then \( \text{CTL}_{\text{pos}}\text{-MC}(T) \) is \( \text{LOGCFL} \)-complete. In the case that no CTL-operators are used, \( \text{NC}^1 \)-completeness of evaluating propositional formulae applies. As a direct consequence (Theorem 4.1), model checking for \( \text{CTL}(T) \) is \( P \)-complete for every nonempty \( T \). This shows that for the majority of interesting fragments, model checking \( \text{CTL}(T) \) 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 \( \Delta_1^P \) in [LMS01]. Here we proved that model checking fragments of \( \text{CTL}_{\text{pos}}^+(T) \) and \( \text{ECTL}_{\text{pos}}^+(T) \) for \( T \subseteq \{A, E, X\} \) remains tractable, while the existential and the universal fragments of \( \text{CTL}_{\text{pos}}^+(T) \) and \( \text{ECTL}_{\text{pos}}^+(T) \) containing temporal operators other than \( X \) are complete for \( \text{NP} \) and \( \text{coNP} \), 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 [BMS00], where the complexity of \( \text{LTL}\text{-MC}(T, B) \) 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 \( \text{NL} \)-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 \( \text{CTL}^* \), over arbitrary sets of Boolean operators would be one way to generalise our results. In the case of \( \text{CTL}^+ \) and
Figure 4: Complexity of $\text{CTL}_{\text{pos}}\text{-MC}(T)$ 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).

CTL$^*$, where model checking is intractable [ELS7, 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 $\text{CTL-SAT}(T,B)$ and $\text{CTL}^*-\text{SAT}(T,B)$, their complexity has been determined—with respect to the set of Boolean operators, completely—in [MNTV09].

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


MODEL CHECKING CTL IS ALMOST ALWAYS INHERENTLY SEQUENTIAL


