A

Parametrised Complexity of Satisfiability in Temporal Logic

MARTIN LÜCK, Leibniz Universität Hannover
ARNE MEIER, Leibniz Universität Hannover
IRENA SCHINDLER, Leibniz Universität Hannover

We apply the concept of formula treewidth and pathwidth to computation tree logic CTL, linear temporal logic LTL, and the full branching time logic CTL*. Several representations of formulas as graph-like structures are discussed, and corresponding notions of treewidth and pathwidth are introduced. As an application for such structures, we present a classification in terms of parametrised complexity of the satisfiability problem, where we make use of Courcelle's famous theorem for recognition of certain classes of structures. Our classification shows a dichotomy between W[1]-hard and fixed-parameter tractable operator fragments almost independently of the chosen graph representation. The only fragments that are proven to be in FPT are those that are restricted to the X operator. By investigating Boolean operator fragments in the sense of Post's lattice we achieve the same complexity as in the unrestricted case if the set of available Boolean functions can express the function "negation of the implication". Conversely, we show containment in FPT for almost all other clones.

CCS Concepts: •**Theory of computation** → **Complexity theory and logic;** *Complexity classes;*

General Terms: Complexity, Logic

Additional Key Words and Phrases: Parametrised complexity, temporal logic, linear temporal logic, computation tree logic, treewidth, pathwidth, temporal depth, Post's lattice

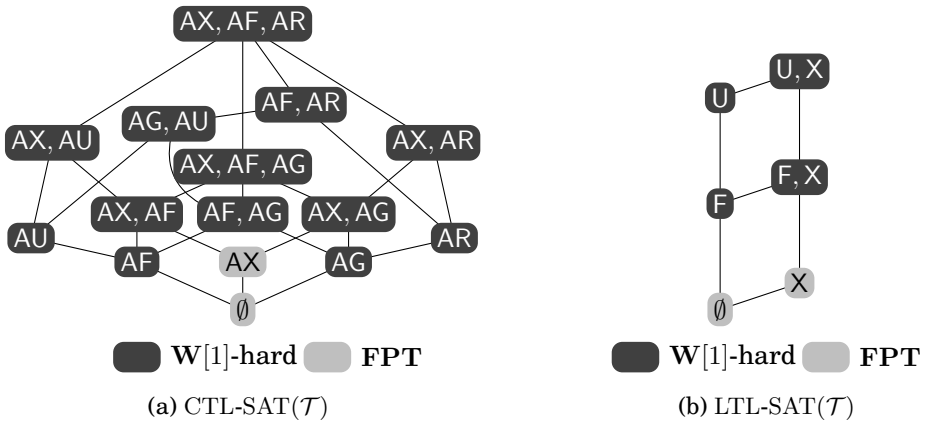
ACM Reference Format:
Martin Lück, Arne Meier, Irena Schindler, 2015. Parametrised Complexity of Satisfiability in Temporal Logic. *ACM Trans. Comput. Logic* V, N, Article A (January YYYY), 31 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Temporal logic is a very important concept in computer science in the area of program verification and is widely used to express specifications of programs. This type of logic can be traced back to the late 1950s where a seminal contribution by Prior [1957] established the foundations of this field. Since then a large area of research has evolved and the most notable contributions have been made by Kripke [1963], Pnueli [1977], Allen Emerson and Halpern [1985], as well as Allen Emerson and Clarke [1981]. Computation tree logic CTL is arguably the most important temporal logic due to its polynomial time solvable model checking problem. Its satisfiability problem—the question whether a given specification is consistent—is complete for deterministic exponential time and therefore beyond tractability. The linear temporal logic LTL has a satisfiability problem that is "only" complete for polynomial space but that, on the other hand, is equivalent to its model checking problem. Despite the intractability of the model checking for LTL, efficient verification tools have been constructed and the logic itself has proven its relevance in practice. The full branching time logic CTL*

This work is supported by DFG grant ME 4279/1-1. Part of this work has been published in a preliminary form in: M. Lück and A. Meier and I. Schindler, Parameterized Complexity of CTL, Proc. LATA 2015, pp. 549–560, vol. 8977 LNCS. [Lück et al. 2015].
Author's addresses: L. Lück and A. Meier and I. Schindler, Leibniz Universität Hannover, Institut für Theoretische Informatik, Appelstrasse 4, 30167 Hannover, Germany.

DOI: http://dx.doi.org/10.1145/0000000.0000000



(a) CTL-SAT($\mathcal{T}$) (b) LTL-SAT($\mathcal{T}$)

Fig. 1: Parametrised complexity of satisfiability problem of CTL and LTL, parametrised by circuit pathwidth or treewidth, and temporal depth.

has a polynomial space complete model checking problem whereas its satisfiability problem is complete for doubly exponential time and therefore even much harder than the corresponding problem for the previous two logics.

One way to attack the intrinsic hardness of temporal problems in logics on a theoretical level is to consider restrictions of the problem by means of operator fragments, Boolean connectives, or bounded temporal depth. Another very prominent theory, which by now is more than a decade old, allows us to better understand the structure of intractability: Downey and Fellows [1999] started the area of parametrised complexity and up to today this field has shown a tremendous growth. Informally, the main idea is to detect a specific part of the problem, the *parameter*, such that the intractability of the problem vanishes if the parameter is assumed to be small. Through this approach the notion of *fixed-parameter tractability* has emerged: a problem is said to be fixed-parameter tractable (or short, FPT) if for some recursive function f there exists a deterministic algorithm running in time $f(k) \cdot \mathit{poly}(n)$ for all input lengths n and corresponding parameter values k. As an example, the satisfiability problem for propositional logic SAT (which is well-known to be NP-complete) becomes fixed-parameter tractable under the parameter that counts the number of variables of the given formula.

In the last decades, the concept of *treewidth* of graphs was carried over from graph theory into logic. The definition of *formula treewidth* as the treewidth of a formula represented by an underlying relational structure has been used for various satisfiability problems, including propositional logic with counting [Samer and Szeider 2006], conjunctive query containment [Chekuri and Rajaraman 1997; Kolaitis and Vardi 2000], constraint satisfaction [Samer and Szeider 2010], and modal logic [Praveen 2013]. This parameter has proven to often yield fixed-parameter tractability.

Results. In this work, we extend the concept of formula treewidth to temporal logics, and completely classify the satisfiability problem of almost all temporal and Boolean operator fragments with respect to its parametrised complexity. We show a dichotomy consisting of the X fragments being fixed-parameter tractable and the remainder being hard for the complexity class $\mathbf{W}[1]$ under fpt-reductions. In the context of parametrised complexity theory, $\mathbf{W}[1]$ is a class of intractable problems and can be seen as an analogue to **NP**. To obtain this classification, we use Courcelle's theorem [Courcelle and Engelfriet 2012].

Related work. Praveen's [2013] analogous research for modal logic influenced the present work in some parts. His **FPT** results for modal CNF logic comply with the results for the X fragments presented here, as well as his hardness results for transitive modal logic do with our results for other temporal operators. Further applications of Courcelle's theorem have been investigated by Gottlob et al. [2010] and Meier et al. [2012]. Elberfeld et al. [2010] enriched Courcelle's theorem to provide upper bounds for the complexity class XL, wherefore Corollary 4.6 can be extended to this class, too.

Organisation. In this paper we first give an introduction into basic notions of parametrised complexity theory, treewidth, and logic. In particular we will introduce the family of temporal logics CTL, LTL, and CTL* (Section 2). The representation of formulas as different relational structures is discussed in Section 3 where we also motivate the introduction of two new types of representations. With these representations of formulas we use Courcelle's theorem and apply it to the tractable cases of temporal satisfiability (Section 4) to obtain **FPT** results, while the other cases are covered by the hardness results in Section 5. We conclude with Section 6 where we extend the results to most Boolean clones of Post's lattice.

2. PRELIMINARIES

We assume familiarity with standard notions of complexity theory such as Turing machines, polynomial time reductions, and the classes **P** and **NP**. For a much deeper introduction into this field, we refer the reader to the very good textbook of Pippenger [1997].

2.1. Complexity Theory

Let Σ be an alphabet. A pair $\Pi = (Q, \kappa)$ is a *parametrised problem* if $Q \subseteq \Sigma^*$ and $\kappa\colon \Sigma^* \to \mathbb{N}$ is a function. For a given instance $x \in \Sigma^*$ we refer to x as the *input*. A function $\kappa\colon \Sigma^* \to \mathbb{N}$ is said to be the *parametrisation* or *parameter of* Π. We say a parametrised problem Π is *fixed-parameter tractable* (or "in the class **FPT**", or just "**FPT**") if there exists a deterministic algorithm deciding Π in time $f(\kappa(x)) \cdot |x|^{O(1)}$ for every $x \in \Sigma^*$ and some recursive function f. Observe that the notion of fixed-parameter tractability is easily extended beyond decision problems.

If $\Pi = (Q, \kappa)$, $\Pi' = (Q', \kappa')$ are parametrised problems over alphabets Σ, Δ then an *fpt-reduction from* Π *to* Π' is a mapping $r\colon \Sigma^* \to \Delta^*$ with the following three properties:

(1) For all $x \in \Sigma^*$ it holds that $x \in Q$ iff $r(x) \in Q'$.
(2) r is fixed-parameter tractable, i.e., r is computable in time $f(\kappa(x)) \cdot |x|^{O(1)}$ for a recursive function $f\colon \mathbb{N} \to \mathbb{N}$.
(3) There exists a recursive function $g\colon \mathbb{N} \to \mathbb{N}$ such that for all $x \in \Sigma^*$ it holds $\kappa'(r(x)) \leq g(\kappa(x))$.

If there exists an fpt-reduction from Π to Π' then we say that Π *is fpt-reducible to* Π', or in symbols $\Pi \leq^{fpt} \Pi'$. If $\Pi \leq^{fpt} \Pi'$ and $\Pi' \leq^{fpt} \Pi$, then write $\Pi \equiv^{fpt} \Pi'$.

The class $\mathbf{W}[1]$ is a parametrised complexity class which plays a similar role as **NP** in the sense of intractability in the parametrised world. The class $\mathbf{W}[1]$ is a superset of **FPT** and a hierarchy of other **W**-classes are built above it: $\mathbf{FPT} \subseteq \mathbf{W}[1] \subseteq \mathbf{W}[2] \subseteq \cdots \subseteq \mathbf{W}[\mathbf{P}]$. All these classes are closed under fpt-reductions. It is not known whether any of these inclusions is strict. For further information on this topic we refer the reader to the textbook of Flum and Grohe [2006].

Similarly to **FPT** the following classes are defined: The class para-**NP** contains the parametrised problems (Q, κ) for which there is a computable function f and an NTM deciding if $x \in Q$ in time $f(\kappa(x)) \cdot |x|^{O(1)}$. The class para-**PSPACE** contains the

parametrised problems (Q, κ) for which there is a computable function f and a DTM deciding if $x \in Q$ in space $f(\kappa(x)) \cdot |x|^{O(1)}$. The class para-**EXPTIME** contains the parametrised problems (Q, κ) for which there is a computable function f and a DTM deciding if $x \in Q$ in time $f(\kappa(x)) \cdot 2^{|x|^{O(1)}}$.

The ℓ-th *slice* of a parametrised problem (Q, κ) is denoted $(Q, \kappa)_\ell$ and defined as:

$$(Q, \kappa)_\ell := \{\, x \mid x \in Q \text{ and } \kappa(x) = \ell \,\}$$

THEOREM 2.1 ([FLUM AND GROHE 2006]). *Let $\mathcal{C}$ be a complexity class in $\{ \mathbf{NP}, \mathbf{PSPACE}, \mathbf{EXPTIME} \}$. Let (Q, κ) be a parametrised problem, $\emptyset \subsetneq Q \subsetneq \Sigma^*$. Then (Q, κ) is para-$\mathcal{C}$-hard if and only if a union of finitely many slices of (Q, κ) is $\mathcal{C}$-hard.*

2.2. Tree- and Pathwidth

Given a finite structure $\mathcal{A}$ (with universe A) we define a *tree decomposition of* $\mathcal{A}$ to be a tuple $\mathcal{T} = (T, X)$ where $T = (V, E)$ is a finite tree and $X = (\mathcal{B}_v)_{v \in V}$ is a family of subsets of A (the set of *bags*), satisfying the following conditions:

(1) Every element of the universe appears in at least one bag: $\bigcup_{v \in V} \mathcal{B}_v = A$.
(2) Every tuple is contained in a bag: for each $(a_1, \ldots, a_k) \in R$ where R is a relation in $\mathcal{A}$, there exists a $v \in V$ such that $\{ a_1, \ldots, a_k \} \subseteq \mathcal{B}_v$.
(3) For every element a the set of bags containing a is connected, i.e., for all $a \in A$ the set $\{ v \mid a \in \mathcal{B}_v \}$ induces a connected subtree in T.

The *width* of a decomposition (T, X) is $\text{width}(T, X) := \max \{ |\mathcal{B}_v| \mid v \in V \} - 1$ which is the size of the largest bag minus 1. The *treewidth* of a structure $\mathcal{A}$ is the minimum of the widths of all tree decompositions of $\mathcal{A}$. Informally the treewidth of a structure describes its tree-alikeness. The closer the value is to 1 the more the structure resembles a tree.

A *path decomposition* (of a given structure $\mathcal{A}$) is defined in a way similar to tree decompositions, except that for the corresponding tuple (T, X), T has to be a path graph. Here $\text{pw}(\mathcal{A})$ denotes the *pathwidth* of $\mathcal{A}$. Likewise, pathwidth captures the similarity of a structure to a path. Observe that pathwidth is an upper bound for treewidth, i.e., $\text{tw} \leq \text{pw}$.

2.3. Post's Lattice

Post [1941] defined the lattice of all *Boolean clones*. A *clone* is defined as follows. Let B be a finite set of Boolean functions. Then any superset of B is called a clone if it contains all projections and is closed under arbitrary compositions of functions from B. With $[B]$ we denote the smallest clone containing B and call B also a *base* of $[B]$. In order to describe all possible clones of Boolean functions we have to introduce a list of properties of functions. We say an n-ary Boolean function f is

— *c-reproducing* if $f(c, \ldots, c) = c$,
— *monotone* if $a_1 \leq b_1, \ldots, a_n \leq b_n$ implies $f(a_1, \ldots, a_n) \leq f(b_1, \ldots, b_n)$,
— *c-separating* if there is an $i \in \{ 1, \ldots, n \}$ s.t. $f(a_1, \ldots, a_n) = 1$ implies $a_i = c$,
— *c-separating of degree n* if all $A \subseteq f^{-1}(c)$ with $|A| = n$ are c-separating, where $A \subseteq \{0,1\}^m$ is c-separating if there is an $i \in \{1, \ldots, m\}$ with $(b_1, \ldots, b_m) \in A$ implies $b_i = c$,
— *self-dual* if $f \equiv \text{dual}(f)$, where $\text{dual}(f)(x_1, \ldots, x_n) = \neg f(\neg x_1, \ldots, \neg x_n)$, and
— *linear* (or *affine*) if $f(x_1, \ldots, x_n) \equiv x_1 \oplus \cdots \oplus x_n \oplus c$ for some c.

The list of all clones with their finite bases is shown in the appendix on Table II, where id is the identity function (i.e., $\text{id}(x) = x$), and $T_n^{n+1} := \bigvee_{i=0}^{n} (x_0 \wedge \cdots \wedge x_{i-1} \wedge x_{i+1} \wedge \cdots \wedge x_n)$

is a threshold function requiring n bits out of $n+1$ set to $\top$. The lattice is depicted in the appendix on page 31.

Remarkably, this lattice has been extensively used as a tool to investigate on a deeper level the inherent computational complexity of problems on Boolean functions. A major result to point out is due to Lewis [Lewis 1979] who used this lattice to fragmentise the propositional satisfiability problem $\mathrm{SAT}(\mathsf{B})$ with respect to possible clones B. He was able to prove in the late 1970s that $\mathrm{SAT}(\mathsf{B})$ is already **NP**-complete if $\nrightarrow \in [\mathsf{B}]$ holds, where $x \nrightarrow y \equiv x \wedge \neg y$ is the negation of implication, and otherwise is in **P**. Further this lattice has been used to investigate a plethora of different types of logics, e.g., temporal logics [Meier et al. 2009; Beyersdorff et al. 2011], circuits [Böhler et al. 2012], constraints [Bauland et al. 2010], non-monotonic logics [Beyersdorff et al. 2010; Creignou et al. 2012], description logics [Meier and Schneider 2013], hybrid logics [Meier et al. 2010], and modal logic [Hemaspaandra et al. 2010] to name only a few.

2.4. Logics

Let Φ be a finite set of propositional letters, and let B be a finite set of Boolean functions. A *propositional formula* ($\mathcal{PL}$ formula) is inductively defined as follows. Any *propositional letter* (or *proposition*) $p \in \Phi$ is a $\mathcal{PL}$ formula. If $f \in B$ is an n-ary function, and $\phi_1, \ldots, \phi_n$ are $\mathcal{PL}$ formulas then so is $f(\phi_1, \ldots, \phi_n)$. Temporal logic extends propositional logic by introducing five *temporal operators*: *next* X, *future* F, *globally* G, *until* U, and *release* R. Together with the two *path quantifiers*, *exists* E and *all* A, they fix the set of the *full branching time logic* (CTL*) formulas as follows. Every proposition p is a *state formula*, and if $\phi_1, \ldots, \phi_n$ are state formulas, so are $\mathsf{A}\phi_1$, $\mathsf{E}\phi_1$, and $f(\phi_1, \ldots, \phi_n)$, where $f \in B$ is an n-ary Boolean function. Every state formula is also a *path formula*. If $\phi_1, \ldots, \phi_n$ are path formulas, so are $\mathsf{O}\phi_1, \phi_1\mathsf{U}\phi_2, \phi_1\mathsf{R}\phi_2, f(\phi_1, \ldots, \phi_n)$, where $\mathsf{O} \in \{ \mathsf{X}, \mathsf{F}, \mathsf{G} \}$ is a unary temporal operator and f is as above. Finally if ϕ is a path formula, then $\mathsf{A}\phi$ and $\mathsf{E}\phi$ are state formulas. Write $\mathcal{CTL}^\star(B,T)$, where $T \subseteq \{ \mathsf{X}, \mathsf{F}, \mathsf{G}, \mathsf{U}, \mathsf{R} \}$, for the set of state formulas that use only Boolean functions from B and only temporal operators O such that O or its dual $\overline{O}$ is in T. The dual operators are defined as $\overline{\mathsf{X}} := \mathsf{X}$, $\overline{\mathsf{F}} := \mathsf{G}$, $\overline{\mathsf{G}} := \mathsf{F}$, $\overline{\mathsf{R}} := \mathsf{U}$, and $\overline{\mathsf{U}} := \mathsf{R}$. Analogously, $\mathcal{LTL}(B,T)$ is the set of all path formulas without occurrences of A or E. Finally, $\mathcal{CTL}(B,T)$ is the set of all state formulas where every path quantifier is immediately followed by a temporal operator and, vice versa, every temporal operator directly occurs after a path quantifier. Here, the dual operators are defined analogously as $\overline{\mathsf{AX}} := \mathsf{EX}$, $\overline{\mathsf{AF}} := \mathsf{EG}$, and so on. We will omit T if it is the full set of temporal operators, and B if it can express all Boolean formulas, e.g., $[B] = \mathsf{BF}$.

Let us turn to the notion of Kripke semantics. Let Φ be a finite set of propositions. A *Kripke structure* $K = (W, R, V)$ is a finite set of *worlds* W, a *successor relation* $R \subseteq W \times W$, and an *evaluation function* $V \colon W \to 2^\Phi$ labelling sets of propositions to worlds. A *path* π in a Kripke structure $K = (W, R, V)$ is an infinite sequence of worlds $w_0, w_1, \ldots$ such that for every $i \in \mathbb{N}$ $w_i R w_{i+1}$. With $\pi(i)$ we refer to the i-th world w_i in π. Denote with $\Pi(w)$ the set of all paths starting at w. For temporal formulas we define the semantics for a given Kripke structure $K = (W, R, V)$, a world $w \in W$, temporal formulas $\phi, \phi_1, \ldots, \phi_n$, a path π, a proposition $p \in \Phi$, and a Boolean function f (note that for simplicity we identify the operator corresponding to the function f with the same symbol) as

$$
\begin{array}{lcl}
K, w \models p & \Leftrightarrow & p \in V(w), \\
K, w \models f(\phi_1, \phi_2, \ldots, \phi_n) & \Leftrightarrow & \text{there is an assignment } \theta \colon \{ x_1, \ldots, x_n \} \to \{ 0, 1 \} \\
 & & \text{such that } \theta \models f(x_1, \ldots, x_n), \text{ and} \\
 & & \text{for all } 1 \leq i \leq n \text{ it holds } \theta(i) = 1 \text{ iff } K, w \models \phi_i, \\
K, w \models \mathsf{A}\phi & \Leftrightarrow & \text{for all } \pi \in \Pi(w) \text{ it holds } K, \pi \models \phi,
\end{array}
$$

$$
\begin{array}{lcl}
K, w \models \mathsf{E}\phi & \Leftrightarrow & \text{there exists a } \pi \in \Pi(w) \text{ such that } K, \pi \models \phi, \\
K, \pi \models p & \Leftrightarrow & p \in V(\pi(0)), \\
K, \pi \models f(\phi_1, \phi_2, \ldots, \phi_n) & \Leftrightarrow & \text{there is an assignment } \theta\colon \{ x_1, \ldots, x_n \} \to \{ 0, 1 \} \\
 & & \text{such that } \theta \models f(x_1, \ldots, x_n), \text{ and} \\
 & & \text{for all } 1 \leq i \leq n \text{ it holds } \theta(x_i) = 1 \text{ iff } K, \pi \models \phi_i, \\
K, \pi \models \mathsf{A}\phi & \Leftrightarrow & K, \pi(0) \models \mathsf{A}\phi, \\
K, \pi \models \mathsf{E}\phi & \Leftrightarrow & K, \pi(0) \models \mathsf{E}\phi, \\
K, \pi \models \mathsf{X}\phi & \Leftrightarrow & K, \pi(1) \models \phi, \\
K, \pi \models \mathsf{F}\phi & \Leftrightarrow & \text{there exists an } i \geq 0 \text{ such that } K, \pi(i) \models \phi, \\
K, \pi \models \mathsf{G}\phi & \Leftrightarrow & \text{for all } i \geq 0 : K, \pi(i) \models \phi, \\
K, \pi \models \phi \mathsf{U} \psi & \Leftrightarrow & \exists\, i \geq 0\, \forall\, j < i : K, \pi(j) \models \phi \text{ and } K, \pi(i) \models \psi, \\
K, \pi \models \phi \mathsf{R} \psi & \Leftrightarrow & \forall\, i \geq 0\, \exists\, j < i : K, \pi(j) \models \phi \text{ or } K, \pi(i) \models \psi.
\end{array}
$$

For a formula $\phi \in \mathcal{CTL}$ (resp., $\mathcal{CTL}^\star$) we define the satisfiability problem CTL-SAT (resp., CTL*-SAT) asking if there exists a Kripke structure $K = (W, R, V)$ and $w \in W$ such that $K, w \models \phi$ and K is *serial* (i.e., for every $w \in W$ there exists a $w' \in W$ with wRw'). The problem LTL-SAT is analogously asking if for a given formula $\phi \in \mathcal{LTL}$ there is a serial Kripke structure $K = (W, R, V)$ and a path π in K such that $K, \pi \models \phi$. If $K \models \phi$ then we also say that K is a *model of* ϕ.

For a set T of temporal operators, and a finite set of Boolean functions B, the problems CTL-SAT(B, T), LTL-SAT(B, T) and CTL*-SAT(B, T) are the restrictions of CTL-SAT, LTL-SAT and CTL*-SAT to formulas in $\mathcal{CTL}(B, T)$, $\mathcal{LTL}(B, T)$ and $\mathcal{CTL}^\star(B, T)$.

Given ϕ, we define $\mathrm{SF}(\phi)$ as the *set of all subformulas of* ϕ according to the inductive definition (containing ϕ itself). The *temporal depth of* ϕ, in symbols $\mathrm{td}(\phi)$, is defined inductively as follows.

$$
\begin{array}{llll}
\mathrm{td}(p) & := 0, & \mathrm{td}(f(\phi_1, \ldots, \phi_n)) & := \max\{ \mathrm{td}(\phi_1), \ldots, \mathrm{td}(\phi_n), 0 \}, \\
\mathrm{td}(\mathsf{P}\phi) & := \mathrm{td}(\phi), & \mathrm{td}(\phi \mathsf{U} \psi) & := \max\{ \mathrm{td}(\phi), \mathrm{td}(\psi) \} + 1, \\
\mathrm{td}(\mathsf{T}\phi) & := \mathrm{td}(\phi) + 1, & \mathrm{td}(\phi \mathsf{R} \psi) & := \max\{ \mathrm{td}(\phi), \mathrm{td}(\psi) \} + 1,
\end{array}
$$

where f is Boolean function, $\mathsf{P} \in \{ \mathsf{A}, \mathsf{E} \}$, $\mathsf{T} \in \{ \mathsf{X}, \mathsf{F}, \mathsf{G} \}$, and p is a propositional symbol.

Vocabularies are tuples of *relational symbols* (or *predicates*) of finite arity $k \geq 1$ (if $k = 1$ then we say the predicate is *unary*) that are usually denoted with the symbol τ. A *structure* $\mathcal{A}$ over a vocabulary τ consists of a *universe* A that is a non-empty set, and a relation $P^{\mathcal{A}} \subseteq A^k$ for each predicate P in τ of arity k. Monadic second order logic ($\mathcal{MSO}$) is the restriction of second order logic ($\mathcal{SO}$) that allows only quantification over unary relations (elements of the universe can still be quantified arbitrarily).

3. STRUCTURAL REPRESENTATIONS OF FORMULAS

In this section we want to define several structural representations of temporal formulas. Several concepts are already known from propositional logic, the simplest is perhaps the *primal graph* also known as *constraint graph* or *Gaifman graph*. The primal graph generally is used to represent arbitrary relations as a binary edge relation [Gaifman 1982]. When applied to propositional logic in conjunctive normal form, it contains a vertex for every propositional variable x in ϕ, and an edge for every pair (x, y) of variables that occur together in some clause C (regardless of whether the literals are negated or not). The rationale behind this concept is clear: If two variables are not connected, then the clauses they occur in are independent of each other, and therefore a Boolean assignment of the first variable can be chosen independently of the other.

Obviously this concept makes sense only for formulas in CNF. A slight variation of this definition gives the *dual primal graph* or just *dual graph*. The dual graph contains the clauses of a CNF as its vertices, and two clauses are connected if they share any variable. Of course unconnected clauses can again be considered independently.

More sophisticated is the concept of the *incidence graph* [Chekuri and Rajaraman 1997]. Formally an incidence graph models some relation by representing both the tuples in the relation and the individuals of the universe as vertices. The edges in such a graph then connect individuals and tuples that contain said individuals, hence the incidence graph is always bipartite. The advantage is that the exact relation can efficiently be reconstructed from the graph representation. To apply the incidence graph to propositional logic, the clauses in a formula are considered as the tuples and the variables as individuals.

The next step is to remove the restriction to CNF formulas. For his fixed-parameter tractability result on modal logic, Praveen uses an extended CNF which also contains modal operators. We still want to circumvent this restricted syntax, in particular to support arbitrary bases of Boolean clones of Post's lattice. Therefore, in this section more possible graph representations of formulas are discussed. Note that the inductive definition of temporal formulas already gives rise to a "natural" representation in form of a *syntax tree*: One vertex v is the root and represents the formula ϕ, and as the children of each vertex the vertices representing direct subformulas are appended.

However, this approach is not yet sufficient to provide a base for algorithmic reasoning. An algorithm that processes a syntax tree, whether top-down or bottom-up, has to respect logical implications between nodes. Therefore we change the definition in a straight-forward way: Instead of a syntax tree we consider a *syntax circuit* in which identical subformulas are identified as a single vertex. The formal definition of such a syntax circuit as a relational structure is as follows.

Definition 3.1 (*Syntax circuit*). Let B be a finite set of Boolean functions, and let T be a set of temporal operators. The vocabulary of our interest is τ, being defined as $\tau := \{\, \mathrm{repr}^1_{\mathsf{O}}, \mathrm{body}^{n+1}_{\mathsf{O}} \mid \mathsf{O} \in T, \mathrm{ar}(\mathsf{O}) = n \,\} \cup \{\, \mathrm{repr}^1_f, \mathrm{body}^{n+1}_f \mid f \in B, \mathrm{ar}(f) = n \} \cup \{\, \mathsf{var}^1 \,\}$.

The *syntax circuit* of ϕ, denoted $\mathcal{C}_\phi$, is defined as the structure $(\mathrm{SF}(\phi), \tau^{\mathcal{C}_\phi})$, i.e., the universe consists of all subformulas of ϕ including ϕ itself, and the relations are interpreted as follows:

— $\mathsf{var}^1(x)$ holds iff x represents a variable,
— $\mathrm{repr}^1_{\mathsf{O}}(x)$ holds iff x represents a formula $\mathsf{O}(\psi_1, \ldots, \psi_n)$ for some n and $\mathsf{O} \in T$,
— $\mathrm{body}^{n+1}_{\mathsf{O}}(x, y_1, \ldots, y_n)$ holds iff x represents the formula $\mathsf{O}(\psi_1, \ldots, \psi_n)$ and $\psi_1, \ldots, \psi_n$ are represented by $y_1, \ldots, y_n$ and $\mathsf{O} \in T$ is a temporal operator of arity n,
— $\mathrm{repr}^1_f(x)$ holds iff x represents a formula $f(\psi_1, \ldots, \psi_n)$ for some n and $f \in B$,
— $\mathrm{body}^{n+1}_f(x, y_1, \ldots, y_n)$ holds iff x represents the formula $f(\psi_1, \ldots, \psi_n)$ and $\psi_1, \ldots, \psi_n$ are represented by $y_1, \ldots, y_n$ and $f \in B$ is a Boolean function of arity n.

When we want to express some specific properties of syntax circuits the following shortcuts are useful:

— y is the i-th argument of x:
$\mathrm{body}^2_{R,i}(x, y) := \exists y_1 \ldots \exists y_{\mathrm{ar}(R)} \mathrm{body}^{\mathrm{ar}(R)+1}_R(x, y_1, \ldots, y_n) \wedge y_i = y$
— y is some argument of x:
$\mathrm{child}^2(x, y) := \bigvee_{R \in T \cup B} \bigvee_{1 \le i \le \mathrm{ar}(R)} \mathrm{body}^2_{R,i}(x, y)$
— x is the root of the structure:
$\mathrm{repr}^1(x) := \forall y \neg \mathrm{child}^2(y, x)$

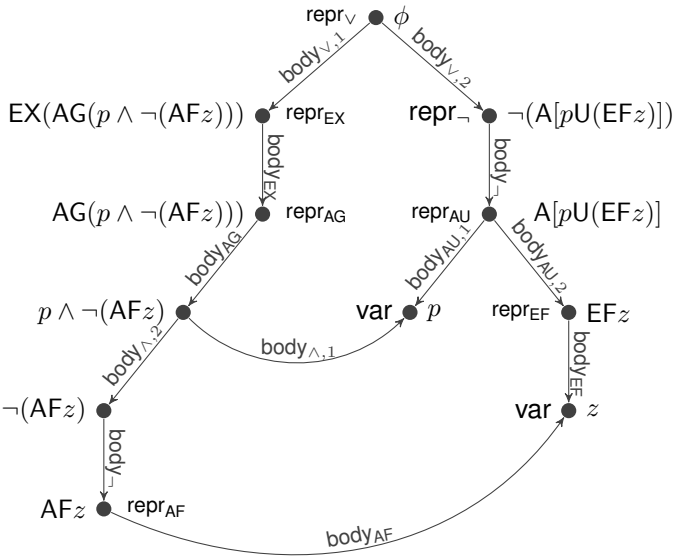


Fig. 2: Example syntactical circuit $\mathcal{C}_\phi$ as relational structure.

As an example, Figure 2 shows the corresponding structure $\mathcal{A}_\phi$ of the CTL formula $\phi := \mathsf{EX}(\mathsf{AG}(p \wedge \neg(\mathsf{AF}z))) \vee (\neg(\mathsf{A}[p\mathsf{U}(\mathsf{EF}z)]))$.

Representing a formula succinctly as a circuit is an old concept. One could object that this approach is unfair as this more succinct encoding can lead to a higher computational complexity (e.g., from $\mathbf{W}[\mathrm{SAT}]$-completeness to $\mathbf{W}[\mathbf{P}]$-completeness for propositional satisfiability). We will nevertheless contrast the circuit encoding with a more formula-like encoding; as it is open whether the treewidths and pathwidths of both representations are comparable, they both have to be considered in terms of parametrised complexity.

Definition 3.2 (*Syntax tree*). The *syntax tree* $\mathcal{S}_\phi$ of a formula ϕ is defined like its syntax circuit, but instead of the subformulas $\mathrm{SF}(\phi)$ the set of *ordered subformulas* $\mathrm{SF}_o(\phi)$ forms the individuals. They are defined like subformulas but distinct occurrences of the same non-atomic subformula are contained multiple times. Formally this can be achieved by replacing occurrences of each symbol $\mathsf{O} \in T \cup B$ with indexed variants $\mathsf{O}', \mathsf{O}'', \ldots$.

Note that the example in Figure 2 shows a syntax circuit which is also a syntax tree.
The *treewidth* of a relational structure is explained in Section 2.

Definition 3.3 (*Formula treewidth / pathwidth*). For a propositional formula ϕ in CNF, its *primal treewidth* $\mathrm{tw}^*(\phi)$ is the treewidth of its primal graph and its *incidence treewidth* $\mathrm{tw}_I(\phi)$ is the treewidth of its incidence graph. For a temporal formula ϕ, its *circuit treewidth* $\mathrm{tw}_\mathcal{C}(\phi)$ is the treewidth of its syntax circuit, and its *syntax treewidth* $\mathrm{tw}_\mathcal{S}(\phi)$ is the treewidth of its syntax tree. The terms primal, incidence, circuit and syntax pathwidth are defined analogously.

Now we consider the problem of temporal satisfiability parametrised by several notions of treewidth and pathwidth as well as the temporal depth of the formula. Hence the parametrisation function κ maps, given a formula ϕ, to the pathwidth of

the structures $\mathcal{A}_\phi \in \{ \mathcal{S}_\phi, \mathcal{C}_\phi \}$ plus the temporal depth of ϕ, i.e., $\kappa(\phi) = \mathrm{pw}(\mathcal{A}_\phi) + \mathrm{td}(\phi)$, resp., $\kappa(\phi) = \mathrm{tw}(\mathcal{A}_\phi) + \mathrm{td}(\phi)$.

The next sections present the collection of results regarding fixed-parameter tractability.

4. FIXED-PARAMETER TRACTABLE FRAGMENTS

One way to prove the membership of a parametrised problem in the class **FPT** is to use Courcelle's theorem [Courcelle and Engelfriet 2012, Thm. 6.3 (1)]:

THEOREM 4.1. *Model checking for $\mathcal{MSO}$ is in* **FPT** *when parameterized by the treewidth of the input structure and the length of the input formula.*

To apply this in temporal logics, we first introduce the notion of *quasi-models*. The crucial difference to a (Kripke) model is that we do not need to talk about *truth* of a subformula, but rather only whether a subformula or its negation is *necessitated* in a specific world at all. This approach is well-known in literature for establishing upper bounds for model size. Related notions are *Hintikka structures*, *pseudo-models*, or *tableaux*.

Definition 4.2 (*Closure*). Let B be a finite set of Boolean functions, T be a set of temporal operators, and ϕ be a $\mathcal{CTL}^\star(B,T)$ formula. Then define $\sim\psi := \xi$ if $\psi = \neg\xi$ for some ξ, and $\sim\psi := \neg\psi$ otherwise. Further define $\overline{\mathsf{A}} := \mathsf{E}$, $\overline{\mathsf{E}} := \mathsf{A}$, $\overline{\mathsf{F}} := \mathsf{G}$, $\overline{\mathsf{G}} := \mathsf{F}$, $\overline{\mathsf{U}} := \mathsf{R}$, $\overline{\mathsf{R}} := \mathsf{U}$, and $\overline{\mathsf{X}} := \mathsf{X}$. Set $P := \{ \mathsf{A}, \mathsf{E} \}$. Now the *closure* $\mathrm{cl}(\phi)$ of ϕ is the smallest set for which the following holds:

— $\phi \in \mathrm{cl}(\phi)$.
— If $O \in T \cup P$, $O\psi \in \mathrm{cl}(\phi)$, then $\overline{O}\sim\psi, \psi \in \mathrm{cl}(\phi)$.
— If $O \in T$, $\psi O\xi \in \mathrm{cl}(\phi)$, then $\sim\psi\overline{O}\sim\xi, \psi, \xi \in \mathrm{cl}(\phi)$.
— If $f(\psi_1, \ldots, \psi_n) \in \mathrm{cl}(\phi)$, $f \in C$, then $\psi_1, \ldots, \psi_n \in \mathrm{cl}(\phi)$.
— $\psi \in \mathrm{cl}(\phi)$ iff $\sim\psi \in \mathrm{cl}(\phi)$.

The closure cl is related to the *Ladner-Fischer closure* defined for PDL [Fischer and Ladner 1979]. Note that $\neg$ is used independently of whether there is $\neg \in B$, hence $\mathrm{cl}(\phi)$ is not necessarily a subset of $\mathcal{CTL}^\star(B,T)$.

Definition 4.3 (*Quasi-models*). Let $\phi \in \mathcal{CTL}^\star(B, \{ \mathsf{A}, \mathsf{E}, \mathsf{X} \})$. A *quasi-model* of ϕ then is a tuple $K = (W, R, L, L_\mathsf{A}, L_\mathsf{E})$ where (W, R) is a Kripke frame and $L, L_\eth \colon W \to 2^{\mathrm{cl}(\phi)}$ for $\eth \in \{ \mathsf{A}, \mathsf{E} \}$ are *extended labelling functions* with the following *quasi-label conditions*:

(1) If $L(w)$ (resp., $L_\eth(w)$) contains $f(\psi_1, \ldots, \psi_n)$, resp., $\sim f(\psi_1, \ldots, \psi_n)$ for some n-ary $f \in B$, then there is a Boolean assignment θ s.t. $\theta \models f$, resp., $\theta \not\models f$, and f.a. $1 \leq i \leq n$ it holds that $\theta(i) = 1$, in fact, implies $\psi_i \in L(w)$ (resp., in $L_\eth(w)$), and conversely $\theta(i) = 0$ implies $\sim\psi_i \in L(w)$ (resp., in $L_\eth(w)$).
(2) If $\psi, \neg\xi \in L(w)$ ($L_\eth(w)$), then $\psi \neq \xi$.
(3) If $\psi \in L_\eth(w)$ is a state formula, then $\psi \in L(w)$.
(4) If $\mathsf{E}\psi \in L(w)$ (resp., $\mathsf{A}\psi \in L(w)$), then $\psi \in L_\mathsf{E}(w)$ (resp., $\psi \in L_\mathsf{A}(w)$).
(5) If $\neg\mathsf{X}\psi \in L_\eth(w)$, then $\mathsf{X}\sim\psi \in L_\eth(w)$.
(6) If $\mathsf{X}\psi \in L_\mathsf{A}(w)$ (resp., in $L_\mathsf{E}(w)$), then for every (resp., some) successor w' of w it is $\psi \in L_\mathsf{A}(w')$ (resp., in $L_\mathsf{E}(w')$).
(7) $\phi \in L(w)$ for some $w \in W$.

$L_\mathsf{A}(w)$ is the *universal quasi-label* of a world w and $L_\mathsf{E}(w)$ the *existential quasi-label*. The items (1) to (5) are the *local* quasi-label conditions. L is the *local quasi-label* which contains only state formulas.

If (K, w_0) is a quasi-model with root w_0, define its *depth* as the maximal distance of a world from w_0. Call a quasi-model *tree-like* if its root has no predecessor and every other world has exactly one predecessor or is a *leaf*, where a leaf has itself as the only successor and exactly one other predecessor.

LEMMA 4.4. *Let $\phi \in \mathcal{CTL}^\star(B, \{ \mathsf{A}, \mathsf{E}, \mathsf{X} \})$ for a finite set B of Boolean functions. Then ϕ is satisfiable if and only if it has a serial tree-like quasi-model of depth $\mathrm{td}(\phi)$.*

PROOF. This can be proven analogously to the *tree model property* of modal logic, i.e., a model can always be "unravelled" to an (infinite) tree [Blackburn et al. 2001, p. 269, Lemma 35]. The inductive conditions of the quasi-labels reflect exactly the semantics of truth for CTL*. The set $L_{\mathsf{A}}(w)$ corresponds to the formulas that should hold in all paths π starting at w, and the set $L_{\mathsf{E}}(w)$ are the formulas that must hold in at least one path starting at w. If such a formula again starts with A or E, i.e., is a state formula and therefore in $L(w)$, then it is also contained in the proper set $L_{\mathsf{A}}(w)$, resp., $L_{\mathsf{E}}(w)$, but has to fulfil no further conditions. If a formula starts with X, then it is inherited to one or more successors of w, and its type (universal or existential) does not change. Boolean functions are also decomposed without changing the type. Finally ϕ itself has to be contained in some quasi-label. By induction over the nesting depth of X it can be easily shown that in minimal quasi-models the sets $L(w)$, $L_{\mathsf{A}}(w)$, and $L_{\mathsf{E}}(w)$ of worlds w of depth $\mathrm{td}(\phi)$ can only contain X-free formulas and therefore the successors can be replaced by self-loops. To obtain a model from a quasi-model and vice versa is straightforward: If the labels L_{A} and L_{E} are removed and from L all labeled formulas except propositions are dropped, then we obtain a model from a quasi-model. The other way around, to enrich a model K to a quasi-model, set $L_\eth(w) := \{ \psi \mid \psi \in \mathrm{cl}(\phi)$ and $K, w \models \eth\psi \}$ for $\eth \in \{ \mathsf{A}, \mathsf{E} \}$ and $L(w) := \{ \psi \mid \psi \in \mathrm{cl}(\phi)$ and $K, w \models \psi \}$. □

LEMMA 4.5. *Let B be a finite set of Boolean functions. There is a computable mapping from $n \in \mathbb{N}$ to an $\mathcal{MSO}$ formula θ_n such that the following holds: For all $\phi \in \mathcal{CTL}^\star(B, \{ \mathsf{A}, \mathsf{E}, \mathsf{X} \})$ with $\mathrm{td}(\phi) \leq n$, ϕ is satisfiable iff $\mathcal{S}_\phi \models \theta_n$ iff $\mathcal{C}_\phi \models \theta_n$.*

PROOF. In the following, we give an algorithm that constructs the $\mathcal{MSO}$ formula θ_n from n. As the formula speaks about structures $\mathcal{S}_\phi$ for $\mathcal{CTL}^\star$ formulas ϕ, the domain of each structure is the set $\mathrm{SF}(\phi)$ of subformulas of ϕ. Every relation symbol therefore is a relation between subformulas (see Definition 3.1 and 3.2), and every quantified monadic relation is a subset of $\mathrm{SF}(\phi)$. To check the satisfiability of ϕ, the formula θ_n expresses, roughly speaking, the existence of a serial tree-like quasi-model of depth n, by repeatedly quantifying sets representing the quasi-label of the root world, its successor worlds, and so on. This is sufficient according to the previous lemma. We have to be careful here as the formula would grow depending on $|\phi|$ in the naïve approach of quantifying worlds for every occurring E operator. Thus we express the branching to successor worlds with a universal quantifier, and hence construct a formula θ_n that depends only on n. The subformulas that verify the quasi-label conditions (1)–(6) of Definition 4.3 follow.

(1)+(2)	Consistency w.r.t. Boolean functions and negation	θ_{local}
(3)	State formulas of $L_\eth(w)$ are in $L(w)$	θ_{state}
(4)	Path formulas ϕ s. t. $\eth\phi \in L(w)$, $\eth \in \{\mathsf{A}, \mathsf{E}\}$, are in $L_\eth(w)$	$\theta_{\mathsf{A}}, \theta_{\mathsf{E}}$
(5)+(6)	Quantify a tree and forward X-preceded path formulas to successors	θ^n_{tree}

It follows the definition of θ_{local}. A minor technical obstacle is that negated formulas are not necessarily available in the domain of the structure, hence we separate each quasi-label L into two sets $H, \overline{H}$, in which the formulas of H must be true in L and

those in $\overline{H}$ must be false in L.

$$\begin{aligned}\theta_{\text{local}}(H,\overline{H}) := \forall x\,(\neg H(x) \vee \neg\overline{H}(x)) \wedge \\ \bigwedge_{\substack{f\in B\\ \text{ar}(f)=k}} \forall x\forall y_1 \ldots \forall y_k\ \text{body}_f(x,y_1,\ldots,y_k) \rightarrow \\ \Big(H(x) \rightarrow \bigvee_{\alpha\models f}\ \bigwedge_{\substack{1\le i\le k,\\ \alpha(i)=1}} H(y_i) \wedge \bigwedge_{\substack{1\le i\le k,\\ \alpha(i)=0}} \overline{H}(y_i)\Big) \\ \wedge \Big(\overline{H}(x) \rightarrow \bigvee_{\alpha\not\models f}\ \bigwedge_{\substack{1\le i\le k,\\ \alpha(i)=1}} H(y_i) \wedge \bigwedge_{\substack{1\le i\le k,\\ \alpha(i)=0}} \overline{H}(y_i)\Big)\end{aligned}$$

The next formula $\theta_{\text{inherit O}}(H,\overline{H},H',\overline{H}')$ is used as a shortcut and states that for the given unary operator O and all formulas $\mathsf{O}\gamma \in H$ ($\overline{H}$) it also holds $\gamma \in H'$ ($\overline{H}'$). This applies for the operators $\mathsf{O} \in \{\mathsf{A},\mathsf{E}\}$ as their semantics can be formulated as a relation between two quasi-labels.

$$\begin{aligned}\theta_{\text{inherit O}}(H,\overline{H},H',\overline{H}') :=\ & \forall x\forall y\ (H(x) \wedge \text{body}_O(x,y) \rightarrow H'(y)) \wedge \\ & (\overline{H}(x) \wedge \text{body}_O(x,y) \rightarrow \overline{H}'(y))\end{aligned}$$

From this we can easily define the remaining formulas. Note that our implementation of θ_{state} requires all formulas of $L_{\mathsf{A}} \cup L_{\mathsf{E}}$ to be in L, not just state formulas, since state formulas cannot easily be detected in the given vocabulary. But this is no real restriction as (non-state) path formulas in L are not checked further by any of the other subformulas.

$$\begin{aligned}\theta_{\text{state}}(H,\overline{H},H_{\mathsf{A}},\overline{H}_{\mathsf{A}},H_{\mathsf{E}},\overline{H}_{\mathsf{E}}) := \forall x((H_{\mathsf{A}}(x) \vee H_{\mathsf{E}}(x)) \rightarrow H(x)) \wedge \\ \forall x((\overline{H}_{\mathsf{A}}(x) \vee \overline{H}_{\mathsf{E}}(x)) \rightarrow \overline{H}(x))\end{aligned}$$

The formulas θ_{O}, $\mathsf{O} \in \{\mathsf{A},\mathsf{E}\}$, from the table above can then be defined as $\theta_{\text{inherit O}}$. Combining the above expressions yields the formula θ_{world} that verifies that the quasi-label conditions (1)–(4) (i.e., the conditions not regarding X) are all satisfied.

$$\begin{aligned}\theta_{\text{world}}(H,\overline{H},H_{\mathsf{A}},\overline{H}_{\mathsf{A}},H_{\mathsf{E}},\overline{H}_{\mathsf{E}}) :=\ & \theta_{\text{local}}(H,\overline{H}) \wedge \theta_{\text{local}}(H_{\mathsf{A}},\overline{H}_{\mathsf{A}}) \wedge \theta_{\text{local}}(H_{\mathsf{E}},\overline{H}_{\mathsf{E}}) \wedge \\ & \theta_{\text{state}}(H,\overline{H},H_{\mathsf{A}},\overline{H}_{\mathsf{A}},H_{\mathsf{E}},\overline{H}_{\mathsf{E}}) \wedge \theta_{\mathsf{A}}(H,\overline{H},H_{\mathsf{A}},\overline{H}_{\mathsf{A}}) \wedge \theta_{\mathsf{E}}(H,\overline{H},H_{\mathsf{E}},\overline{H}_{\mathsf{E}})\end{aligned}$$

The final formula recursively expresses the existence of the tree-like quasi-model by pushing the arguments of the X operator to the respectively quantified successor worlds (represented by six new set variables that form the corresponding quasi-labels).

$n = 0$: $\theta^n_{\text{tree}}(H,\overline{H},H_{\mathsf{A}},\overline{H}_{\mathsf{A}},H_{\mathsf{E}},\overline{H}_{\mathsf{E}}) :=\ \theta_{\text{world}}(H,\overline{H},H_{\mathsf{A}},\overline{H}_{\mathsf{A}},H_{\mathsf{E}},\overline{H}_{\mathsf{E}})$

$n > 0$: $\theta^n_{\text{tree}}(H,\overline{H},H_{\mathsf{A}},\overline{H}_{\mathsf{A}},H_{\mathsf{E}},\overline{H}_{\mathsf{E}}) :=\ \theta_{\text{world}}(H,\overline{H},H_{\mathsf{A}},\overline{H}_{\mathsf{A}},H_{\mathsf{E}},\overline{H}_{\mathsf{E}}) \wedge$

$$\Big(\exists H'\,\exists\overline{H}'\,\exists H'_{\mathsf{A}}\,\exists\overline{H}'_{\mathsf{A}}\,\exists H'_{\mathsf{E}}\,\exists\overline{H}'_{\mathsf{E}}$$

$$\theta^{n-1}_{\text{tree}}(H',\overline{H}',H'_{\mathsf{A}},\overline{H}'_{\mathsf{A}},H'_{\mathsf{E}},\overline{H}'_{\mathsf{E}}) \wedge \theta_{\text{inherit X}}(H_{\mathsf{A}},\overline{H}_{\mathsf{A}},H'_{\mathsf{A}},\overline{H}'_{\mathsf{A}})\Big) \wedge$$

$$\bigwedge_{K\in\{H_{\mathsf{E}},\overline{H}_{\mathsf{E}}\}} \forall x\forall y\Big(K(x)\wedge \mathrm{body}_{\mathsf{X}}(x,y)\to \exists H'\,\exists\overline{H}'\,\exists H'_{\mathsf{A}}\,\exists\overline{H}'_{\mathsf{A}}\,\exists H'_{\mathsf{E}}\,\exists\overline{H}'_{\mathsf{E}}$$
$$\theta^{n-1}_{\text{tree}}(H',\overline{H}',H'_{\mathsf{A}},\overline{H}'_{\mathsf{A}},H'_{\mathsf{E}},\overline{H}'_{\mathsf{E}})\wedge\theta_{\text{inherit}\,\mathsf{X}}(H_{\mathsf{A}},\overline{H}_{\mathsf{A}},H'_{\mathsf{A}},\overline{H}'_{\mathsf{A}})\wedge K'(y)\Big)$$

Assuming that formulas are restricted to temporal depth $\leq n$, it holds that $\theta^n_{\text{tree}}(H,\overline{H},H_{\mathsf{E}},\overline{H}_{\mathsf{E}},H_{\mathsf{A}},\overline{H}_{\mathsf{A}})$ is true when there is a serial tree-like quasi-model of depth n with root w such that $L(w)=H\cup\{\,\neg\alpha\mid\alpha\in\overline{H}\,\}$, $L_{\mathsf{A}}(w)=H_{\mathsf{A}}\cup\{\,\neg\alpha\mid\alpha\in\overline{H}_{\mathsf{A}}\,\}$ and $L_{\mathsf{E}}(w)=H_{\mathsf{E}}\cup\{\,\neg\alpha\mid\alpha\in\overline{H}_{\mathsf{E}}\,\}$, where all quasi-label conditions are satisfied. We show this by induction over n.

For $n=0$, θ^n_{tree} just expresses that there is a single world w with a valid quasi-label but arbitrary successors, so it is true iff ϕ is satisfiable in the semantics of propositional logic.

Let $n>0$. θ^n_{tree} then additionally imposes that w has some successor w' (this is required for seriality). Also, for each formula $\mathsf{X}\psi\in L_{\mathsf{E}}(w)$ there is a successor w' that has $\psi\in L_{\mathsf{E}}(w')$. Furthermore, all successors w' respect the label $L_{\mathsf{A}}(w)$, i.e., $\mathsf{X}\psi\in L_{\mathsf{A}}(w)$ implies $\psi\in L_{\mathsf{A}}(w')$. The world w hence satisfies all quasi-label conditions (1)–(6), and as for every successor world w' the formula $\theta^{n-1}_{\text{tree}}$ is imposed, they all have tree-like quasi-models. To construct the quasi-model for w we apply the induction hypothesis and insert the respective trees with w' as root together with an edge from w to every w'.

As any quasi-model can be brought in this normal form, i.e., each existentially quantified X formula is satisfied by a new successor, we can from such a model conclude the truth of $\theta^n_{\text{tree}}(H,\overline{H},H_{\mathsf{E}},\overline{H}_{\mathsf{E}},H_{\mathsf{A}},\overline{H}_{\mathsf{A}})$ where the arguments are chosen according to the quasi-labels $L,L_{\mathsf{A}},L_{\mathsf{E}}$ of the root of the model. This shows the other direction.

Finally consider the $\mathcal{MSO}$ formula

$$\theta_n:=\exists H\exists\overline{H}\exists H_{\mathsf{A}}\exists\overline{H}_{\mathsf{A}}\exists H_{\mathsf{E}}\exists\overline{H}_{\mathsf{E}}\;\exists x\,\mathrm{repr}(x)\;\wedge H(x)\;\wedge\;\theta^n_{\text{tree}}(H,\overline{H},H_{\mathsf{E}},\overline{H}_{\mathsf{E}},H_{\mathsf{A}},\overline{H}_{\mathsf{A}}).$$

We use this formula to apply Courcelle's theorem. It additionally ensures the condition (7), i.e., $\phi\in L(w)$ for the root w. Hence for all formulas $\phi\in\mathcal{CTL}^\star(B,\{\mathsf{A},\mathsf{E},\mathsf{X}\})$ with $\mathrm{td}(\phi)\leq n$ it holds that ϕ has a quasi-model of depth n iff $\mathcal{S}_\phi\models\theta_n$ iff $\mathcal{C}_\phi\models\theta_n$, and, due to the previous lemma, iff ϕ is satisfiable. □

THEOREM 4.6. *Let B be a finite set of Boolean functions, $T\subseteq\{\mathsf{A},\mathsf{E},\mathsf{X}\}$. Then the problem* CTL*-SAT(B,T) *parametrised by td $+\,\kappa$ is fixed-parameter tractable if $\kappa\in\{tw_{\mathcal{C}},tw_{\mathcal{S}},pw_{\mathcal{C}},pw_{\mathcal{S}}\}$.*

PROOF. As the pathwidth of a formula is bounded from below by the treewidth of the formula it suffices to consider the treewidth parameters. We apply Theorem 4.1 in the following way: For every given formula ϕ we compute $n:=\mathrm{td}(\phi)$. This allows us to compute the MSO formula θ_n from Lemma 4.5 in time depending only on the parameter td. Further it holds that ϕ is satisfiable if and only if $\mathcal{S}_\phi\models\theta_n$ if and only if $\mathcal{C}_\phi\models\theta_n$. Also $\mathcal{S}_\phi$ and $\mathcal{C}_\phi$ can be computed in polynomial time, thus the theorem follows. □

CTL-SAT and LTL-SAT straightforwardly reduce to CTL*-SAT:

COROLLARY 4.7. *The problems* CTL-SAT(B,T), LTL-SAT(B,T') *parametrised by td $+\,\kappa$ are fixed-parameter tractable if $\kappa\in\{tw_{\mathcal{C}},tw_{\mathcal{S}},pw_{\mathcal{C}},pw_{\mathcal{S}}\}$ and $T\subseteq\{\mathsf{AX}\}$, $T'\subseteq\{\mathsf{X}\}$.*

THEOREM 4.8. *Let B be a finite set of Boolean functions. For $T\subseteq\{\mathsf{X}\}$, the problem* LTL-SAT(B,T) *is in* **FPT** *when parametrised by $tw_{\mathcal{S}}$ or $pw_{\mathcal{S}}$.*

PROOF. It holds due to the path semantics of LTL that X distributes over arbitrary Boolean functions, i.e., $\mathsf{X}f(\phi_1,\dots,\phi_n) \equiv f(\mathsf{X}\phi_1,\dots,\mathsf{X}\phi_n)$ for $f \in B, \phi_1,\dots,\phi_n \in \mathcal{LTL}$. Hence every LTL formula with only X-operators can efficiently be converted to an equivalent Boolean combination β of X-preceded variables:

$$\phi \equiv \beta(\mathsf{X}^{n_1}q_1,\dots,\mathsf{X}^{n_m}q_m), \text{ where } \mathsf{X}^{n_i} := \underbrace{\mathsf{X}\dots\mathsf{X}}_{n_i \text{ times}},$$

the q_i are propositional variables and $n_i \geq 0$. Inconsistent literals can only occur inside the same world and therefore at the same nesting depth of X. Hence the above formula ϕ is satisfiable if and only if it is satisfiable as a purely propositional formula where each expression $\mathsf{X}^{n_i}q_i$ is interpreted as an atomic proposition.

For example,

$$\begin{aligned}\phi &= \mathsf{X}(a_1 \wedge a_2) \vee \mathsf{X}(a_3 \wedge \mathsf{X}(a_1 \wedge a_2))\\ &\equiv (\underbrace{\mathsf{X}a_1}_{p_1} \wedge \underbrace{\mathsf{X}a_2}_{p_2}) \vee (\underbrace{\mathsf{X}a_3}_{p_3} \wedge \underbrace{\mathsf{XX}a_1}_{p_4} \wedge \underbrace{\mathsf{XX}a_2}_{p_5})\\ &\equiv (p_1 \wedge p_2) \vee (p_3 \wedge p_4 \wedge p_5).\end{aligned}$$

Formally we have $(\text{LTL-SAT}(B,\mathsf{X}),\kappa) \leq^{\mathit{fpt}} (\text{SAT}(B),\kappa)$ for $\kappa \in \{\, \text{tw}_{\mathcal{S}}, \text{pw}_{\mathcal{S}} \,\}$, or in other words, LTL-SAT is as easy as propositional satisfiability for the temporal operator X. In particular it is $\mathcal{MSO}$-expressible. It remains to show that the pathwidth and treewidth are not increased too much when the X operator is distributed over the Boolean functions. Informally, in the syntax tree the substructure with nodes $\{\, \mathsf{X}, f, \phi_1,\dots,\phi_n \,\}$ and edges $\{\, (\mathsf{X},f),(f,\phi_1),\dots,(f,\phi_n) \,\}$ is locally replaced by a substructure with nodes $\{\, f, \mathsf{X}_1,\dots,\mathsf{X}_n,\phi_1,\dots,\phi_n \,\}$ and edges $\{\, (f,\mathsf{X}_1),(\mathsf{X}_1,\phi_1),\dots,(f,\mathsf{X}_n),(\mathsf{X}_n,\phi_n) \,\}$.

To likewise adapt the tree-, resp., path-decomposition, add to every bag $\mathcal{B}$ containing X or f the nodes $\mathsf{X}, f, \mathsf{X}_1,\dots,\mathsf{X}_n$. Here n is a constant and depends only on the particular $f \in B$. This ensures that the edges to former parents of X that are now parents of f are covered, and that the edges to former children of f that are now children of $\mathsf{X}_1,\dots,\mathsf{X}_n$ are covered. The edges inside the new substructure are covered as well. If the path-, resp., treewidth previously was k, then every bag $\mathcal{B}$ contained at most $k+1$ nodes representing some f or X. Therefore the new width of the bag is at most $(k+1)\cdot(c+2)$, where c is the maximum arity of any $f \in B$ and thus constant.

Now $(\text{SAT}(B),\text{tw}_{\mathcal{S}})$ and $(\text{SAT}(B),\text{pw}_{\mathcal{S}})$ are a special case of CTL* with $\text{td} = 0$, which is itself in **FPT** according to Theorem 4.6. Hence the above reduction yields $(\text{LTL-SAT}(B,\{\,\mathsf{X}\,\}),\kappa) \in \textbf{FPT}$ for $\kappa \in \{\, \text{pw}_{\mathcal{S}}, \text{tw}_{\mathcal{S}} \,\}$. □

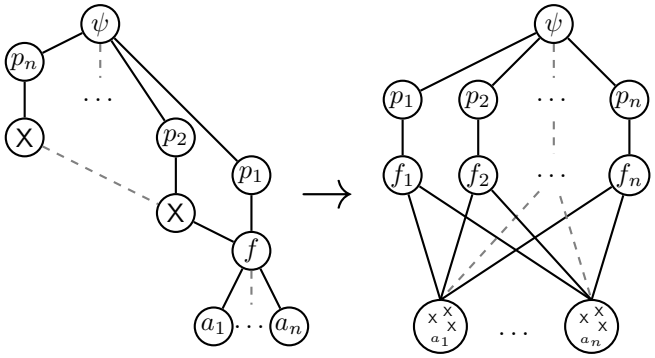


Fig. 3: From constant pathwidth to unbounded treewidth after distributing X

In contrast to other operator fragments and parametrisations, the choice of the structural representation of formulas is crucial in the last result. The proof does not work in the circuit representation as the resulting treewidth and pathwidth cannot be bounded during the transformation which pushes the X operator inside. We elucidate the example from Figure 3: Let f be a node in the circuit with parents $p_1, \ldots, p_n$ such that between p_i and f there is a stack of $i-1$ X-operators. In the circuit, these Xes are represented by only n nodes $\mathsf{X}_1, \ldots, \mathsf{X}_n$ with edges between them, as subformulas are reused. Assume that $p_1, \ldots, p_n$ form the whole formula together with some root node ψ. Let f further have propositional arguments $a_1, \ldots, a_n$. The pathwidth and therefore the treewidth is constant: Use a sequence of bags $B_1, \ldots, B_n$ with $B_i := \{ \psi, p_i, p_{i-1}, \mathsf{X}_i, \mathsf{X}_{i-1}, f, a_i \}$.

But if X is distributed over f, then the circuit must contain n copies f_i of f such that each f_i has again the arguments $a_1, \ldots, a_n$ but with a stack of i Xes between f_i and the arguments. Also f_i has a single parent p_i, which itself has the parent ψ. This circuit is illustrated in Figure 3 as well. We observe that for each pair (i, j) with $1 \leq i, j \leq n$ the node f_i has a disjoint path to each a_j, always going through i X nodes. Obtain a graph minor from the circuit by merging each node a_j with the stack of n Xes above it to a node A_j. Then for each (i, j) as above we have an edge (f_i, A_j), therefore the circuit contains the $K_{n,n}$ biclique as a minor and therefore has treewidth and hence pathwidth at least n.

We close this gap for syntax circuits in the next section when we present $\mathbf{W}[1]$-hardness of this case. The used reduction forms a syntax circuit with stacked Xes which is similar to our counter-example.

5. FIXED-PARAMETER INTRACTABLE FRAGMENTS

In the following section we discuss fragments of CTL, LTL, and CTL* that have more operators than *next* (X). They have in common that the depth of satisfying models cannot be bounded by the temporal depth of the formula anymore. Therefore, the framework used for the X operator cannot be applied. Instead, we consistently prove $\mathbf{W}[1]$-hardness for all other operators, expanding results for the case of unrestricted temporal operators [Praveen 2013].

5.1. Temporal Depth and Treewidth as Parameter

The problem of *partitioned weighted satisfiability* further generalises the problem of weighted satisfiability and was shown to be $\mathbf{W}[1]$-hard by Praveen [2013] as a step to the intractability of transitive modal logic.

An instance is a tuple $I = \big(\phi, k, (Q_i)_{i\in[k]}, (C_i)_{i\in[k]}\big)$ where ϕ is a propositional formula in CNF over variables $q_1, \ldots, q_n$, and the variables are partitioned into disjoint sets $Q_1, \ldots, Q_k$. Here $[k]$ is a shorthand for the set $\{ 1, \ldots, k \}$. Each partition Q_i has an assigned *capacity* $C_i \in \mathbb{N}$. An assignment θ is called *saturated* for an instance I if in every partition Q_i there are exactly C_i variables set to true by θ. We define a parametrised problem of finding a saturated assignment:

Problem:	p-PW-SAT
Input:	Propositional CNF ϕ, $k \in \mathbb{N}$, $(Q_i)_{i\in[k]}$, $(C_i)_{i\in[k]}$
Question:	Has ϕ a satisfying saturated assignment?
Parameter:	$\kappa := \mathrm{pw}^*(\phi) + k$

Here, the parameter is the sum of the primal pathwidth of ϕ and the number of partitions. Praveen proved the $\mathbf{W}[1]$-hardness of modal satisfiability in transitive frames by formulating the problem of partitioned weighted satisfiability in modal logic. The rough idea is as follows: Consider an instance $I = \big(\phi,\, k,\, (Q_i)_{i\in[k]},\, (C_i)_{i\in[k]}\big)$ of

p-PW-SAT with ϕ containing variables $q_1, \ldots, q_n$. Then a modal formula is constructed which enforces the existence of a connected sequence $w_0 \to w_1 \to w_2 \to \ldots \to w_n$ of worlds, which we call a *chain*. For $i \geq 1$ the world w_i then has the purpose to assign to q_i either $\top$ or $\bot$. In the last world w_n another subformula eventually checks the number of variables set to $\top$ in each partition; the number has to equal the respective capacity.

Crafting such a type of formula is not a new technique, but doing it in the context of a parametrised reduction requires a careful construction of subformulas in a way that keeps the pathwidth low. Significant parts of the technical work in [Praveen 2013] are devoted to prove that the structural pathwidth of the produced formula is bounded by the parameter. We will cover the different CTL fragments in two steps: First the reduction from saturated satisfiability is presented in detail for general CTL. The transition from modal logic to temporal logic that is required in this step is not hard. In the second step the result is transferred to the fragments of CTL.

LEMMA 5.1. $\mathrm{CTL\text{-}SAT}(T)$ *is* $\mathbf{W}[1]$*-hard for* $\{ \mathsf{AX}, \mathsf{AG} \} \subseteq T$ *when parametrised by* $\kappa := td + f$*, where* $f \in \{ tw_{\mathcal{C}}, tw_{\mathcal{S}}, pw_{\mathcal{C}}, pw_{\mathcal{S}} \}$.

PROOF. We give an fpt-reduction from p-PW-SAT that transforms an instance I into a CTL formula $\psi(I)$. The formula $\psi(I)$ is a conjunction of several subformulas that will be presented next. In the construction we assume reasonable properties of the instance, e.g., that the capacity of a partition is positive and less than its size (otherwise such a partition could be removed entirely), and that ϕ is in CNF.

The original formula ϕ should be true in the initial world w_0 to ensure its satisfiability in propositional semantics.

$$\psi[\text{formula}] \quad := \phi$$

Enforce a model that contains the chain of worlds mentioned above. For this we use "depth" variables $d_0, d_1, d_2, \ldots$ that have to be labeled in the desired order.

$$\psi[\text{depth}] \qquad := \mathsf{AG} \bigwedge_{i=0}^{n} [(d_i \wedge \neg d_{i+1}) \to \mathsf{AX}(d_{i+1} \wedge \neg d_{i+2})]$$

Let Q be the subset of the variables $\{ q_1, \ldots, q_n \}$ that is labeled in w_0. $\psi[\text{formula}]$ ensures that Q represents a satisfying assignment of ϕ. To check the saturation of Q w.r.t. the given capacities the set Q should be repeatedly labeled in each consecutive world.

$$\psi[\text{fixed-}Q] \qquad := \mathsf{AG} \bigwedge_{i=1}^{n} [q_i \leftrightarrow \mathsf{AX} q_i]$$

Next let $p(i) \in [k]$ denote the partition number of q_i. We introduce new propositional variables $\top^{\uparrow}_{p(i)}$ which signal that the count of labeled variables from partition $p(i)$ has increased.

$$\psi[\text{signal}] \qquad := \mathsf{AG} \bigwedge_{i=1}^{n} \left[(d_i \wedge \neg d_{i+1}) \to \left(\left(q_i \leftrightarrow \top^{\uparrow}_{p(i)} \right) \wedge \bigwedge_{p' \neq p(i)} \neg \top^{\uparrow}_{p'} \right) \right]$$

To count the total number of labeled variables per partition we need several variables named $\top^j_p$ here. Whenever an increment signal for partition p is encountered, increment the counter from j to the next integer $j+1$, otherwise the counter stays the same.

$$\psi[\text{count}] \qquad := \mathsf{AG} \bigwedge_{p=1}^{k} \bigwedge_{j=0}^{|Q_p|} \left[\top^{\uparrow}_p \to \left((\top^j_p \to \mathsf{AX}\, \top^{j+1}_p) \wedge (\neg \top^j_p \to \mathsf{AX} \neg \top^{j+1}_p) \right) \right.$$

$$\wedge \neg \top_p^{\uparrow} \rightarrow \left((\top_p^j \rightarrow \mathsf{AX}\, \top_p^j) \wedge (\neg \top_p^j \rightarrow \mathsf{AX} \neg \top_p^j) \right) \Big]$$

Make sure that all the used variables carry out a consistent counting.

$$\psi[\text{monotone}] := \mathsf{AG} \left[\bigwedge_{i=1}^{n} (d_i \rightarrow d_{i-1}) \wedge \bigwedge_{p=1}^{k} \bigwedge_{j=1}^{|Q_p|+1} \left(\top_p^j \rightarrow \top_p^{j-1} \right) \right]$$

Begin the counting correctly at the initial world.

$$\psi[\text{init}] \qquad := d_0 \wedge \neg d_1 \wedge \bigwedge_{p=1}^{k} \left[\neg \top_p^{\uparrow} \wedge \neg \top_p^1 \right] \wedge \mathsf{AG} \bigwedge_{p=1}^{k} \top_p^0$$

Require that the counted number of positive variables per partition equals the capacity.

$$\psi[\text{target}] \qquad := \mathsf{AG} \bigwedge_{p=1}^{k} \left[d_{n+1} \rightarrow \left(\top_p^{C_p} \wedge \neg \top_p^{C_p+1} \right) \right]$$

CLAIM 5.2. *The reduction is correct, i.e.,* $I \in$ p-PW-SAT $\Leftrightarrow \psi(I) \in$ CTL-SAT.

PROOF OF CLAIM. "$\Rightarrow$": Assume $I = \left(\phi,\, k,\, (Q_i)_{i\in[k]},\, (C_i)_{i\in[k]}\right) \in$ p-PW-SAT. We construct a model for $\psi(I)$ as follows. Start with the world w_0. ϕ is satisfied by setting a saturated subset Q of variables to one, i.e., the number of ones in a partition equals its capacity. Label all propositions of Q in the world w_0 such that ϕ is satisfied there. Construct successor worlds $w_1, \ldots, w_{n+1}$, add a self-loop to w_{n+1} and label the minimal amount of propositional variables in $w_0, w_1, \ldots$ to satisfy ψ[init], ψ[depth] and ψ[fixed-Q]. Note that this step is always possible. Now label the variables $\top_{p(i)}^{\uparrow}$ where necessary to fulfil ψ[signal]. This leads to exactly $C_{p(i)}$ occurrences of $\top_{p(i)}^{\uparrow}$ since Q is chosen saturated. For this reason, the formula ψ[count] allows for every partition p that its counter is incremented exactly C_p times. This construction does not violate the ψ[monotone] condition and allows to satisfy ψ[target] in the world w_{n+1}.

"$\Leftarrow$": Let $\mathcal{M}$ be a model of $\psi(I)$. We can assume that $\mathcal{M}$ is a tree of depth $n+1$. We show that $\mathcal{M}$ contains a path $w_0, \ldots, w_{n+1}$ from its root w_0 such that this path is *jump-free* in the following sense: If a counter value $\top_p^j$ is labeled in w_i, then in the predecessor w_{i-1} there was already $\top_p^{j-1}$ labeled. Therefore if $\top_p^j$ is set in w_i, but $\top_p^{j+1}$ is not, then in w_{i+1} the proposition $\top_p^{j+1}$ can be set or not set, but $\top_p^{j+2}$ cannot be set.

On such a path the $\top_p^{\uparrow}$ signal must be labeled exactly C_p times for every partition p, as $\top_p^{C_p} \wedge \neg \top_p^{C_p+1}$ holds in w_{n+1} due to ψ[target]. If it was set more often, then $\top_p^{C_p+1}$ would be true in w_{n+1}, and if it is set less than C_p times, but $\top_p^{C_p}$ is still set in w_{n+1}, then the path cannot be jump-free. As ψ[signal] allows $\top_p^{\uparrow}$ only if the corresponding variable q_i is set to one in the world w_i, any jump-free path in $\mathcal{M}$ proves the existence of a saturating assignment θ for ϕ. θ also is satisfying for ϕ since w_0 was labeled consistently.

To see that every Kripke model $\mathcal{M}$ has in fact only jump-free paths, assume for the sake of contradiction that there is a world w in which $\top_p^j$ and $\neg \top_p^{j+1}$ hold, but in a successor $\top_p^{j+2}$ holds. Regardless of whether $\top_p^{\uparrow}$ is set in w, $\mathsf{EX}\, \top_p^{j+2}$ is true in w, and therefore from one of the clauses of ψ[count] it follows that either $\top_p^{j+1}$ or even $\top_p^{j+2}$ is already labeled in w. The latter case implies that $\top_p^{j+1}$ is labeled as well, as ψ[monotone] holds. This is a contradiction to the assumption that $\neg \top_p^{j+1}$ is true in w. So $\mathcal{M}$ has at least one jump-free path due to seriality. $\diamond$

CLAIM 5.3. *$td(\psi(I)) + \kappa(\psi(I))$ is bounded by $pw^*(\phi) + k$.*

PROOF OF CLAIM. Observe that the temporal depth is constant in the reduction. As the treewidth is at most the pathwidth, it remains to show that $\mathrm{pw}_{\mathcal{S}}$ and $\mathrm{pw}_{\mathcal{C}}$ are bounded. Write $\mathcal{S}$ for $\mathcal{S}_{\psi(I)}$, i.e., the syntax tree of $\psi(I)$. Let $\mathcal{P}$ denote an optimal path decomposition of the primal graph of ϕ. We demonstrate how $\mathcal{P}$ can be extended to a path-decomposition $\mathcal{P}'$ of $\mathcal{S}$.

For this we first assume a special structure of $\mathcal{P}$, the *one-step addition property* that was already used by Praveen [2013]. It says that the bag $\mathcal{B}_i$ in $\mathcal{P}$ introduces exactly one variable q, i.e., q and only q is present in $\mathcal{B}_i$ but was not present in $\mathcal{B}_{i-1}$. A bag that introduces no new variable can be deleted, and a bag introducing multiple variables can be split into multiple bags. Therefore assume the one-step addition property. Also use a renaming of bags $\mathcal{B}_i$ and variables q_i s.t. the bags are ordered along their number and bag $\mathcal{B}_i$ introduces variable q_i.

The process of *augmenting a bag $\mathcal{B}$ with x* means inserting a copy $\mathcal{B}'$ of $\mathcal{B}$ between $\mathcal{B}$ and its successor bag and placing the additional element x there. It holds that $|\mathcal{B}'| = |\mathcal{B}| + 1$. Augmenting a bag does preserve the one-step addition property in the sense that there always is a "leftmost" bag introducing a variable q_i. Now use the following procedure to construct $\mathcal{P}'$:

(1) For $1 \leq p \leq k$ add the variable $\top_p^{\uparrow}$ to every bag. This increases every bag size by the number k of partitions.
(2) ψ[formula]: ϕ is in CNF. For every clause the primal graph of ϕ has to contain a clique of its variables, therefore $\mathcal{P}$ already must contain a bag $\mathcal{B}$ covering all these variables. Assume that this clause has size m and is represented as a subformula $((((L_1 \vee L_2) \vee L_3) \cdots) \vee L_m)$ where every literal L_i is a variable q or its negation $\neg q$. Augment $\mathcal{B}$ with the $\vee$-nodes in the following way: Create m copies of $\mathcal{B}$. Add the j-th $\vee$-node to the j-th and the $(j+1)$-th copy of $\mathcal{B}$. This results in the bags containing an $\vee$-node inducing a connected component. Refer to the outmost $\vee$-operators as the *primary $\vee$-nodes*. Proceed similar for the $\wedge$-nodes: Select two primary $\vee$-clauses that are "neighbours" in the path decomposition and add a $\wedge$-node to all bags that connect them.
An optimal path decomposition can be computed in **FPT** [Flum and Grohe 2006, Corollary 11.28]. Hence the structure $\mathcal{S}$ can be constructed in a way that allows the argumentation above *a priori*, and the placement of parentheses in $\psi(I)$ can always be chosen to associate literals in ascending order of variables in $\mathcal{P}$, and to associate clauses in ascending order of primary $\vee$-nodes. Then in the structure the edge linking these primary $\vee$-nodes and their common conjunction is covered and every bag receives at most two additional $\vee$-nodes and at most two additional $\wedge$-nodes. Figure 4 illustrates the procedure.
(3) For $1 \leq i \leq n$ add the variables $d_{i-1}, d_i, d_{i+1}, d_{n+2}$ and the nodes representing their negations as well the nodes representing $(d_i \wedge \neg d_{i+1})$, $(d_{i+1} \wedge \neg d_{i+2})$ and $(d_i \to d_{i-1})$ to the bag $\mathcal{B}$ that introduces q_i. Because of the one-step addition property regarding the q_i's every inserted node induces a connected component. This adds a constant number of items to every bag. The same holds when adding the necessary AX-nodes and $\to$-nodes to every bag to cover each conjunct of ψ[depth]. Process its $\bigwedge$-node like before, adding at most two items to every bag, and also add its AG-node to every bag to completely cover this formula.
(4) ψ[fixed-Q]: Add the AX- and $\leftrightarrow$-nodes for the i-th conjunct exactly to the respective bags that introduce q_i, increasing the size of each bag by at most two. Process $\bigwedge$ and AG as before.

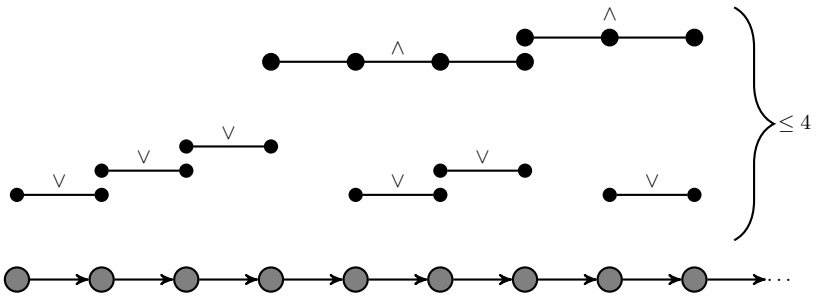


Fig. 4: Bag augmentation for many small $\wedge$,$\vee$-nodes

(5) ψ[signal]: Again augment the bag introducing q_i by the nodes $\leftrightarrow$, $(d_i \wedge \neg d_{i+1})$ and $\rightarrow$. The signal variable $\top^{\uparrow}_{p(i)}$ is already added in every bag, then add $\bigwedge$ and AG as before.
(6) For the remaining formulas we extend the decomposition after the last bag $\mathcal{B}$. W.l.o.g. assume that $\mathcal{B}$ contains the variable d_{n+1}. Let C be the maximum of the capacities. For every $1 \leq j \leq C$ now do the following: Append one bag that is a copy of $\mathcal{B}$, but the j-th appended bag additionally contains $\top^{j}_{p}$, $\top^{j+1}_{p}$ and $\top^{j-1}_{p}$ for every partition p. This increases every bag size by $3k$. Since d_{n+1} also is in these bags, the nodes representing subformulas of ψ[count], ψ[monotone], ψ[init], and ψ[target] containing $\top^{j}_{p}$'s can be added (each a constant number of items). Note that ψ[target] actually requires these nodes to be added after the last bag so the bags containing d_{n+1} are connected. Note that the subformulas containing d_is are already covered in the decomposition by item (3).
(7) The remaining subformulas of $\psi(I)$ are conjunctions of size C over $\top^{j}_{p}$'s (which can be covered by augmenting the bags introduced in the previous step by constantly many items), conjunctions of size k (which again can be added to all bags), and lastly the constantly many remaining connectives that link the previously considered subformulas together.

The above construction results in a path decomposition of the structure $\mathcal{S}_\phi$ whose width is a recursive function of κ. The use of signal variables $\top^{\uparrow}_{p}$ is crucial for the construction: Them being the only "link" between variables q_i and partition weight counters $\top^{j}_{p}$ is necessary for keeping the pathwidth low.

Observe that the pathwidth resulting from this construction will in general be higher than the similar approach for modal CNF by Praveen [2013] (which is only $4\mathrm{pw}(G_\phi) + 2k + 9$). The reason lies in the chosen structural representation of modal formulas, which is similar to primal graphs: For every clause, only one node is added to the structure, whereas the syntax circuits and trees only allow connectives with fixed arity. Therefore the number of $\wedge, \vee$-connectives in $\psi(I)$ itself is not bounded by κ, but by a careful choice of the association order of subformulas the items can be added with bag size increasing only by a constant number.

If $\mathcal{P}'$ should also be a path-decomposition of $\mathcal{C}_{\psi(I)}$, then identical subformulas of $\psi(I)$ actually have to induce connected subpaths in $\mathcal{P}'$, not only propositions, according to the definition of a path-decomposition. Obviously, for every bag $\mathcal{B}$ that contains a formula ξ, the formulas $\neg\xi$ and AXξ can be added to $\mathcal{B}$. The only other subformulas that occur multiple times in $\psi(I)$ are the $(d_i \wedge \neg d_{i+1})$ for $i \in [n]$, but this node can be added to any bag that contains d_i or d_{i+1}. Then the occurrences of $(d_i \wedge \neg d_{i+1})$ are connected in $\mathcal{P}'$, as

the occurrences of d_i as well as d_{i+1} are each connected and they are overlapping. Thus the pathwidth of the circuit representation is bounded by κ, too. $\diamond$

As the reduction is correct and the value of the parameter is bounded we achieve the desired **FPT**-reduction. $\square$

LEMMA 5.4. CTL-SAT(T) *parametrised by td* $+ \kappa$ *is* $\mathbf{W}[1]$*-hard if* $\{ \mathsf{AX}, \mathsf{AF} \} \subseteq T$ *and* $\kappa \in \{ tw_{\mathcal{S}}, pw_{\mathcal{S}}, tw_{\mathcal{C}}, pw_{\mathcal{C}} \}$.

PROOF. The formulas in the proof of Lemma 5.1 are deliberately chosen to have AG operators only at temporal depth zero and in conjunctions. As $\mathsf{AG}(\alpha) \wedge \mathsf{AG}(\beta) \equiv \mathsf{AG}(\alpha \wedge \beta)$, we can modify the formula $\phi_{\mathcal{F}}$ that is a conjunction of the formulas from above to the form containing only a single AG, namely $\phi_{\mathcal{F}} = \psi \wedge \mathsf{AG}\chi$, where ψ is purely propositional and $\chi \in \mathcal{CTL}(\{ \mathsf{AX} \})$. In this formula, AG can be replaced by EG. If a model $\mathcal{M}$ satisfies $\psi \wedge \mathsf{EG}\chi$, then the path π witnessing the path formula $\mathsf{G}\chi$ is again jump-free. Hence the correctness proof works as before, the submodel of $\mathcal{M}$ induced by π still provides a satisfying, saturated assignment. $\square$

LEMMA 5.5. CTL-SAT(T) *parametrised by td* $+ \kappa$ *is* $\mathbf{W}[1]$*-hard if* $\mathsf{AG} \in T$ *or* $\mathsf{AR} \in T$ *and* $\kappa \in \{ tw_{\mathcal{S}}, pw_{\mathcal{S}}, tw_{\mathcal{C}}, pw_{\mathcal{C}} \}$.

PROOF. It is sufficient to consider $T = \{ \mathsf{AG} \}$ as $\mathsf{AG}\phi \equiv \mathsf{A}[\bot \mathsf{R} \phi]$. Both the operators G and F are *stutter-invariant*, i.e., they cannot distinguish a path π and a path π' which is obtained from π by duplicating arbitrary worlds on the path. Hence a more sophisticated construction is required to maintain correctness of the reduction. The first obstacle is that even if we can enforce the worlds $w_1, \ldots, w_n$ to appear, we cannot avoid redundant intermediate worlds. Therefore we cannot check the counter for its exact value as it was the case for the X operator. As we can duplicate any world in the model due to stutter-invariance, we get an arbitrary number of occurrences of variables. In this case we would count one variable several times and get a satisfiable $\mathcal{CTL}$ formula even if ϕ has no saturated assignment. Instead, we impose upper bounds for both the number of ones and the number of zeros in every partition. The second obstacle is that without an accessible X operator we cannot say that the counter has to increase in the next world but *not* in the current, as G and F both are reflexive in the sense that the present world is a part of the future. To circumvent this, we supplement the depth propositions with their "parities". These are represented by two additional variables and thus does not increase the pathwidth much. The following formulas from Lemma 5.1 have to be changed:

In ψ[depth] we replace AX with the branching operator $\neg\mathsf{AG}\neg \equiv \mathsf{EF}$. It is then not longer the case that all paths reachable from w_0 form the desired chain of worlds, however, the branch of the model that satisfies one of the EF-formulas has again to branch correctly at least once for the next depth level because of the nesting inside an AG operator. The depth indicator has to increase in some reachable world due to the semantics of EF. Hence, at least one path starting at w_0 eventually visits all depth propositions in the correct order. To deal with the problem of irreflexivity, we enforce an alternation in terms of variables. Label a new variable m_0, resp., m_1 in worlds of parity 0, resp., 1.

$$\psi[\text{depth}]' := \mathsf{AG} \bigwedge_{i=0}^{n} [(d_i \wedge \neg d_{i+1}) \rightarrow \mathsf{EF}(d_{i+1} \wedge \neg d_{i+2})]$$

$$\psi[\text{alternation}] := \mathsf{AG} \bigwedge_{i=0}^{n-1} \left[(d_i \wedge \neg d_{i+1}) \rightarrow (m_{i \bmod 2} \wedge \neg m_{1-(i \bmod 2)})\right]$$

Fixing the chosen subset of variables q_i is easily done using only AG.

$$\psi[\text{fixed-}Q]' \quad := \bigwedge_{i=1}^{n} [(q_i \to \mathsf{AG} q_i) \wedge (\neg q_i \to \mathsf{AG} \neg q_i)]$$

The signal counting formula has to be adapted to the fact that now there can exist multiple consecutive worlds having the same depth proposition labeled. Also, the counting procedure has to be implemented differently for the case that no increment signal is set. Without AX, we cannot express that the labeled counter propositions may *not* change in the next world. To maintain correctness of the reduction, we have to introduce a second type of counters for variables set to zero, $\bot_p^{\uparrow}$. The following formulas ensure the correctness of the new counter.

$$\psi[\text{signal}]_2 \quad := \mathsf{AG} \bigwedge_{i=1}^{n} \left[(d_i \wedge \neg d_{i+1}) \to \left(\neg q_i \leftrightarrow \bot_{p(i)}^{\uparrow} \right) \right]$$

$$\psi[\text{init}]_2 \quad := \bigwedge_{p=1}^{k} \left[\neg \bot_p^{\uparrow} \wedge \neg \bot_p^{1} \right] \wedge \mathsf{AG} \bigwedge_{p=1}^{k} \bot_p^{0}$$

$$\psi[\text{monotone}]_2 \; := \mathsf{AG} \bigwedge_{p=1}^{k} \bigwedge_{j=1}^{|Q_p|+1} \left(\bot_p^{j} \to \bot_p^{j-1} \right)$$

Check if for partition p at most C_p variables have been set to one and at most $|Q_p| - C_p$ variables have been set to zero.

$$\psi[\text{target}] \quad := \mathsf{AG} \bigwedge_{p=1}^{k} \left[\neg \top_p^{C_p+1} \wedge \neg \bot_p^{|Q_p|-C_p+1} \right]$$

At last, the existing counting procedure has to be replaced and split up into counting of ones and zeros.

$$\psi[\text{count}]_1 \quad := \mathsf{AG} \bigwedge_{p=1}^{k} \bigwedge_{j=0}^{|Q_p|} \bigwedge_{i=0}^{1} \Big[\left(\top_p^{\uparrow} \wedge \top_p^{j} \wedge m_i \right) \to \mathsf{AG} \left(m_{1-i} \to \mathsf{AG}\, \top_p^{j+1} \right) \Big]$$

$$\psi[\text{count}]_2 \quad := \mathsf{AG} \bigwedge_{p=1}^{k} \bigwedge_{j=0}^{|Q_p|} \bigwedge_{i=0}^{1} \Big[\left(\bot_p^{\uparrow} \wedge \bot_p^{j} \wedge m_i \right) \to \mathsf{AG} \left(m_{1-i} \to \mathsf{AG}\, \bot_p^{j+1} \right) \Big]$$

As explained above, there is at least one path of worlds where every depth proposition is reached at least once. If a depth proposition d_i is reached with a signal variable $\top_p^{\uparrow}$ or $\bot_p^{\uparrow}$ labeled, then the corresponding counter value increases during the next parity change of i. Hence, if a partition p has weight k, then on this path there are at least k parity changes with the proposition $\top_p^{\uparrow}$ labeled, and at least $|Q_p| - k$ parity changes with the proposition $\bot_p^{\uparrow}$ labeled. This leads to the counter $\top_p^{j}$ having a value $j \geq k$ and the counter $\bot_p^{j}$ having a value $j \geq |Q_p| - k$ in world w_{n+1}. This contradicts $\psi[\text{target}]$ unless j is exactly k, resp., $|Q_p| - k$ and the partition is saturated.

The pathwidths $\mathrm{pw}_{\mathcal{S}}$ and $\mathrm{pw}_{\mathcal{C}}$ increase only by a constant when considering the changes of the two formulas $\psi[\text{depth}]'$ and $\psi[\text{fixed-}Q]'$. The formula $\psi[\text{alternation}]$ can

be handled by augmenting the bags which introduce d_i. To add the new counting and target formulas the same procedure as in Lemma 5.4 can be used: Treat every variable of the type $\bot_p^{\uparrow}$, $\bot_p^{j}$ like its $\top_p^{\uparrow}$ or $\top_p^{j}$ counterpart to preserve the boundedness of the parameter. □

LEMMA 5.6. CTL-SAT(T) *parametrised by td* + κ *is* $\mathbf{W}[1]$*-hard if* $\mathsf{AU} \in T$ *and* $\kappa \in \{ tw_{\mathcal{S}}, pw_{\mathcal{S}}, tw_{\mathcal{C}}, pw_{\mathcal{C}} \}$.

PROOF. Only minor changes compared to Lemma 5.5 are required to show the $\mathbf{W}[1]$-hardness of the fragment $\{ \mathsf{AU} \}$. Change the formulas as follows: Introduce an additional depth proposition d_{n+2} that has to hold after d_{n+1}. The subformula $\psi[\mathsf{depth}]$ is adapted as follows.

$$\psi[\mathsf{depth}]' := \bigwedge_{i=0}^{n} \mathsf{A}\Big[(d_i \wedge \neg d_{i+1}) \rightarrow \big(m_{i \bmod 2} \wedge \neg m_{1-(i \bmod 2)} \wedge \mathsf{A}\left[\neg d_{n+2} \mathsf{U} (d_{i+1} \wedge \neg d_{i+2})\right]\big) \mathsf{U} d_{n+2}\Big] \quad \square$$

Every other occurrence of AG is positive, thus every subformula $\mathsf{AG}\gamma$ can be replaced by $\mathsf{A}\left[\gamma \mathsf{U} d_{n+2}\right]$ and the reduction stays correct.

The next result differs from the previous ones as it preserves low circuit pathwidths but not low syntax tree pathwidths. This can be explained with the lack of the AG and AU operator. This restriction severely weakens the expressive power of CTL as already observed in the classical case [Meier et al. 2009].

LEMMA 5.7. CTL-SAT(T) *parametrised by td* + κ *is* $\mathbf{W}[1]$*-hard if* $\mathsf{AF} \in T$ *and* $\kappa \in \{ tw_{\mathcal{C}}, pw_{\mathcal{C}} \}$.

PROOF. As in the previous lemmas, we reduce from the problem p-PW-SAT. The saturated, satisfying assignment is again verified by implementing a "counter" for every partition. A crucial difference is that we cannot easily count worlds on a path, but only "frontiers of reachability" about which the operators AF and EG can speak.

Let K be a quasi-model. We use some notions introduced by Allen Emerson [1990]: For a formula $\mathsf{AF}\beta$ labeled in a world w, write DAG$[w, \beta]$ for the finite dag (Directed Acyclic Graph) that starts at w and contains all worlds reachable from w up to the first occurrence of the formula β in a quasi-label. Such a finite dag always exists due to the semantics of AF. Furthermore the dag is not only contained in K, but *embedded* in K, which means that every path through K that leads out of the dag has to go through its leaves. The leaves of DAG$[w, \beta]$ are also called *frontier worlds* and its non-leaves (including w if w is not already a leaf) are called *interior worlds*.

We proceed to the formula which witnesses the reduction. The given formula $\psi(I)$ is satisfiable if and only if $I \in$ p-PW-SAT. It enforces the existence of dags in the above sense which are "nested" in each other; the idea is that every dag increments a counter value depending on the counter values of the dags reachable from its frontier, hence the total number of such dags is propagated to the root of the model.

The formula α will be labeled in the frontier nodes of every such dag. For all partitions p it says the following: "If I am enclosed by a frontier with $\neg\top_p^{\uparrow}$ which is itself enclosed by a frontier with $\top_p^j$, then I assume the counter value $\top_p^{j+1}$", thus it increments the counter for the ones, and similar for the zeros. The condition with $\neg\top_p^{\uparrow}$ is necessary as future is reflexive in CTL and the counter should not jump to the maximal value at the first occurrence of $\top_p^{\uparrow}$. The last two clauses initialise their respective counter to a value of one. By precomputation we can again exclude any instance of p-PW-SAT where for some p it is $C_p = 0$ or $C_p = |Q_p|$, so both one and zero have to occur at least once in every partition.

$$\begin{aligned}\alpha \quad := \bigwedge_{j=0}^{n}\bigwedge_{p=1}^{k}(\top_p^{\uparrow} \wedge \mathsf{AF}(\neg\top_p^{\uparrow} \wedge \mathsf{AF}\top_p^j) \rightarrow \top_p^{j+1}) \\ \wedge (\bot_p^{\uparrow} \wedge \mathsf{AF}(\neg\bot_p^{\uparrow} \wedge \mathsf{AF}\bot_p^j) \rightarrow \bot_p^{j+1}) \\ \wedge (\top_p^{\uparrow} \rightarrow \top_p^1) \wedge (\bot_p^{\uparrow} \rightarrow \bot_p^1)\end{aligned}$$

The formulas β_i^d enforce the existence of nested dags, their frontier worlds having sligthly different labels depending on whether q_i or $\neg q_i$ was chosen for the saturated, satisfying assignment. The formulas β_i^e enforce more dags between the β_i^d-dags which serve as "gaps". These gaps are required for α to work in an irreflexive way, as only alternation of variables can be distinguished by the stutter-invariant operators AF and EG.

$$\begin{aligned}\beta_i^d \quad &:= \left[q_i \rightarrow \mathsf{AF}\left(\top_{p(i)}^{\uparrow} \wedge d_i \wedge \mathsf{EG}(\neg e_{i-1}) \wedge \alpha\right)\right] \\ &\wedge \left[\neg q_i \rightarrow \mathsf{AF}\left(\bot_{p(i)}^{\uparrow} \wedge d_i \wedge \mathsf{EG}(\neg e_{i-1}) \wedge \alpha\right)\right] \\ \beta_i^e \quad &:= \mathsf{AF}\left(e_i \wedge \mathsf{EG}(\neg d_i) \wedge \bigwedge_{p=1}^{k} \neg\top_p^{\uparrow} \wedge \neg\bot_p^{\uparrow}\right)\end{aligned}$$

Finally the formula $\psi(I)$ enforces ϕ to be satisfiable, the dags mentioned above to exist, and that every partition has labeled the correct number of ones and zeros, i.e., the assignment is saturated with respect to I.

$$\psi(I) := \phi \wedge \bigwedge_{i=1}^{n}\left(\beta_i^d \wedge \beta_i^e\right) \wedge \mathsf{EG}\bigwedge_{p=1}^{k} \neg\top_p^{C(p)+1} \wedge \neg\bot_p^{|Q_p|-C(p)+1}$$

Let (K, w_0) be a quasi-model of $\psi(I)$. In the next claims use the shortcut DAG$[i]$ for DAG$[w_0, \beta_i']$ (where β_i' is the AF-preceded formula implied by β_i^d depending on q_i) and DAG$'[i]$ for DAG$[w_0, \beta_i'']$ where $\beta_i^e = \mathsf{AF}\beta_i''$.

CLAIM 5.8. *DAG$[i]$ is contained in the interior worlds of DAG$'[i]$, and DAG$'[i]$ is contained in the interior worlds of DAG$[i+1]$.*

PROOF OF CLAIM. It suffices to show that DAG$[i]$ is contained in DAG$'[i]$. The two dags cannot have common frontier worlds as those worlds would have both d_i and $\neg d_i$ labeled. The same holds for DAG$'[i]$ and DAG$[i+1]$ and the proposition e_i.

Let w be a frontier world of DAG$'[i]$. Then $\beta_i'' \in L(w)$ which implies $\mathsf{EG}\neg d_i \in L(w)$. Let $\pi \in \Pi(w)$ be the path that satisfies $\mathsf{G}\neg d_i$. Every path $\pi' \in \Pi(w_0)$ which runs through w has to visit a "shallower" world w' with $\beta_i' \in L(w')$ before w: Otherwise the path

$$(w_0 = \pi'[0], \pi'[1], \ldots, w = \pi[0], \pi[1], \ldots)$$

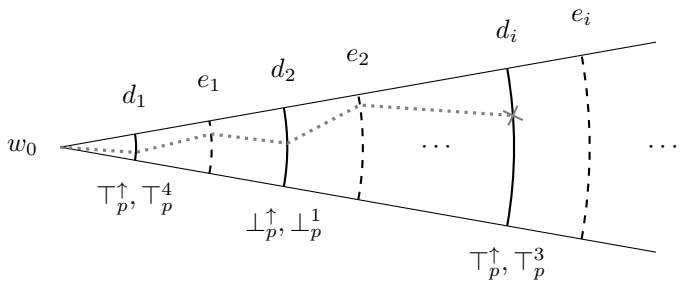


Fig. 5: Example: $\mathsf{EG}\neg\top_p^3$ is false in w_0, Q_p has weight > 2.

would be a path starting in w_0 but not fulfilling $\mathsf{F}\beta_i'$. Then (K, w_0) would not be a quasi-model. This implies that on every path to a frontier node of DAG$'[i]$ there already occurs a frontier node of DAG$[i]$. The same can be shown for DAG$[i+1]$ and DAG$'[i]$. $\diamond$

CLAIM 5.9. *If $\psi(I)$ has a quasi-model (K, w_0), then $I \in$ p-PW-SAT.*

PROOF OF CLAIM. By the previous claim we can assume K to contain $2n$ nested dags such that their frontier worlds have labeled the corresponding subformulas of $\beta_1^d, \beta_1^e, \beta_2^d, \beta_2^e, \ldots, \beta_n^d, \beta_n^e$ in this order. As mentioned, the formula α expresses that the frontier of each β_i^d should, under the condition that it has $\top_p^{\uparrow}$ labeled, do the following: if $\top_p^j$ is labeled in some *reachable*, but *different* frontier, i.e., the counter for partition p was at j, then label $\top_p^{j+1}$, i.e., set the counter to $j+1$. Similar for $\bot_p^j$. Also, initialise the counting with $\top_p^1$ and $\bot_p^1$ if $\top_p^{\uparrow}$, resp., $\bot_p^{\uparrow}$ holds. Choosing a non-saturated assignment for ϕ now contradicts the last part of $\psi(I)$. Therefore we can extract a satisfying, saturated assignment for ϕ from $L(w_0)$. Figure 5 illustrates this argumentation.

The other direction, i.e., the construction of a model of $\psi(I)$ from a satisfying, saturated assignment of ϕ is straightforward: a model in the form of a chain of length $2n+1$ suffices. $\diamond$

Clearly $\psi(I)$ has a constant temporal depth, it remains to show the bounded pathwidth. The subformula α can easily be split into a path-decomposition of width $O(k)$ and length n. The path-decomposition of the remaining formulas can simply be appended to this one, as α has only $O(k)$ subformulas in common with the other formulas (including α itself). It remains to show that the remaining subformulas β_i^d, β_i^e and $\psi(I)$ also have a path-decomposition of low width. Starting from a decomposition of the primal graph of ϕ, the other formulas can be placed by bag augmentation as in Lemma 5.4, again leading to a total syntax circuit pathwidth of $O(k + \mathrm{pw}^*(\phi))$. Hence the lemma holds. $\square$

Note that the formula α of the previous lemma leads to an unbounded pathwidth in the sense of syntax trees. Also, it is open how the reduction could be changed to have low syntax tree pathwidth (or, for that matter, even treewidth).

LEMMA 5.10. *If $T \not\subseteq \{\mathsf{X}\}$, then* LTL-SAT$(T)$ *is* $\mathbf{W}[1]$*-hard when parametrised by $td + \kappa$ where $\kappa \in \{ tw_{\mathcal{S}}, pw_{\mathcal{S}}, tw_{\mathcal{C}}, pw_{\mathcal{C}} \}$.*

PROOF. The result is proven by a parametrised reduction from the problem p-PW-SAT similar to the CTL-SAT(AG) case (cf. Theorem 5.5): Let I be an instance of p-PW-SAT, i.e., $I = \big(\phi, k, (Q_i)_{i\in[k]}, (C_i)_{i\in[k]}\big)$. We consider an equivalent LTL formula $\psi(I)$ that has a low structural pathwidth.

The formula $\psi(I) \in \mathcal{LTL}(\mathsf{G})$ is a conjunction of the following subformulas.

$$\psi[\textbf{formula}] := \phi$$

$$\psi[\textbf{depth}] := \mathsf{G} \bigwedge_{i=0}^{n-1} \big[(d_i \wedge \neg d_{i+1}) \rightarrow (m_{i \bmod 2} \wedge \neg m_{1-(i \bmod 2)} \wedge \neg\mathsf{G}\neg(d_{i+1} \wedge \neg d_{i+2}))\big]$$

$$\psi[\textbf{fixed-}Q] := \bigwedge_{i=1}^{n} [(q_i \rightarrow \mathsf{G}q_i) \wedge (\neg q_i \rightarrow \mathsf{G}\neg q_i)]$$

$$\psi[\textbf{signal}] := \mathsf{G} \bigwedge_{i=1}^{n} \Big[(d_i \wedge \neg d_{i+1}) \rightarrow \Big(\Big(q_i \leftrightarrow \top_{p(i)}^{\uparrow}\Big) \wedge \Big(\neg q_i \leftrightarrow \bot_{p(i)}^{\uparrow}\Big) \Big)\Big]$$

$$\psi[\textbf{count}] := \mathsf{G} \bigwedge_{p=1}^{k} \bigwedge_{j=0}^{|Q_p|} \bigwedge_{i=0}^{1} \Big[\big(\top_p^{\uparrow} \wedge \top_p^{j} \wedge m_i\big) \rightarrow \mathsf{G}\big(m_{1-i} \rightarrow \mathsf{G}\top_p^{j+1}\big) \Big] \wedge \Big[\big(\bot_p^{\uparrow} \wedge \bot_p^{j} \wedge m_i\big) \rightarrow \mathsf{G}\big(m_{1-i} \rightarrow \mathsf{G}\bot_p^{j+1}\big) \Big]$$

$$\psi[\textbf{monotone}] := \mathsf{G}\Big[\bigwedge_{i=1}^{n} (d_i \rightarrow d_{i-1}) \wedge \bigwedge_{p=1}^{k} \bigwedge_{j=1}^{|Q_p|+1} \big[\big(\top_p^{j} \rightarrow \top_p^{j-1}\big) \wedge \big(\bot_p^{j} \rightarrow \bot_p^{j-1}\big)\big] \Big]$$

$$\psi[\textbf{target}] := \mathsf{G} \bigwedge_{p=1}^{k} \Big[d_{n+1} \rightarrow \Big(\top^{C_p} \wedge \neg\top^{C_p+1} \wedge \bot_p^{|Q_p|-C_p} \wedge \neg\bot_p^{|Q_p|-C_p+1} \Big)\Big]$$

$$\psi[\textbf{init}] := d_0 \wedge \neg d_1 \wedge \mathsf{G} \bigwedge_{p=1}^{k} \big[\top_p^0 \wedge \bot_p^0\big]$$

The boundedness of the pathwidth of $\psi(I)$ is proven similar to Lemma 5.1. The pathwidth increases only marginally when replacing G by F, U, or R in the given formulas for the other cases. The correctness follows the argumentation of Lemma 5.5: In the CTL case the formula enforces at least one path which does the correct counting of variables for all partitions. The given LTL formula, which is a path formula, ensures the same behaviour on a single path. □

5.2. Only Temporal Depth or Treewidth as Parameter

Next, we show that the chosen parameters, i.e., temporal depth and treewidth, are both necessary to achieve fixed-parameter tractability of CTL-SAT($\{\mathsf{AX}\}$) and also of LTL-SAT($\{\mathsf{X}\}$) (in the case of syntax treewidth). It is clear that a parametrisation only by temporal depth is unlikely to work, as propositional satisfiability and therefore the case $\mathrm{td} = 0$ of temporal satisfiability is already NP-complete.

THEOREM 5.11. *When parametrised by temporal depth,* $\mathrm{CTL\text{-}SAT}(T)$ *is para-***NP***-complete if* $T = \emptyset$*, para-***PSPACE***-complete if* $T = \{ \mathsf{AG} \}$*, or* $\{ \mathsf{AF} \} \subseteq T \subseteq \{ \mathsf{AF}, \mathsf{AX} \}$*, and para-***EXPTIME***-complete if* T *contains* AR*,* AU*,* $\{ \mathsf{AG}, \mathsf{AF} \}$*, or* $\{ \mathsf{AG}, \mathsf{AX} \}$*.*

PROOF. All classical hardness results except for the $\{ \mathsf{AX} \}$ fragment already hold for temporal depth of two [Lück 2015]. Therefore application of Theorem 2.1 immediately yields the result. □

THEOREM 5.12. *When parametrised by temporal depth,* $\mathrm{LTL\text{-}SAT}(T)$ *is para-***NP***-complete if* $T \subseteq \{ \mathsf{X} \}$ *or* $T \subseteq \{ \mathsf{F} \}$ *and para-***PSPACE***-complete otherwise.*

PROOF. Application of Theorem 2.1 on the classical results of Demri and Schnoebelen [2002]. □

The second case, parametrisation by pathwidth or treewidth only, requires another reduction from p-PW-SAT. As the temporal depth no longer has to be bounded, nested AX operators can be used to formulate the same idea as in Lemma 5.1. Note that $\mathrm{LTL\text{-}SAT}(\{\mathsf{X}\})$ is the only fragment where the different choice of structural representation leads to a difference with respect to fixed-parameter tractability: the problem is in **FPT** when the parameter is the syntax treewidth or pathwidth (see Theorem 4.8), but $\mathbf{W}[1]$-hard when the parameter is the circuit treewidth or pathwidth, as shown in the next lemma.

LEMMA 5.13. $\mathrm{CTL\text{-}SAT}(\{ \mathsf{AX} \})$ *and* $\mathrm{LTL\text{-}SAT}(\{ \mathsf{X} \})$ *are* $\mathbf{W}[1]$*-hard when parametrised by* $tw_{\mathcal{C}}$ *or* $pw_{\mathcal{C}}$*.*

PROOF. The reduction from p-PW-SAT to $\mathrm{CTL\text{-}SAT}(\{ \mathsf{AX} \})$ is witnessed by the same formulas as in the $\{ \mathsf{AG}, \mathsf{AX} \}$ case, Lemma 5.1. Just every formula $\mathsf{AG}\alpha$ is replaced by $\bigwedge_{j=0}^{n} \mathsf{AX}^j \alpha$, where AX^j is the j-fold nesting of AX operators. In the syntax circuit this corresponds to deletion of the AG node and introduction of $2n$ new vertices $v_1, \ldots, v_n, u_0, \ldots, u_{n-1}$ where each v_i represents an AX and each u_i represents a binary $\wedge$. u_0 has as parents every parent of $\mathsf{AG}\alpha$, and each u_i with $1 \leq i < n-1$ has as a child v_i and u_{i+1}. u_0 has α and u_1 as its children, while u_{n-1} has v_{n-1} and v_n. Furthermore v_0 has α as only child, and each v_i with $i > 0$ has v_{i-1} as child. Figure 6 illustrates this.

To keep the pathwidth of the resulting circuit structure low, proceed as follows: As seen in Figure 6, the $2n$ vertices have a path-decomposition of width four and length n. Append this decomposition to $\mathcal{P}$ to obtain $\mathcal{P}'$, where $\mathcal{P}$ is an optimal path-decomposition of the formula before AG is replaced. To avoid to violate the connectedness condition of path-decompositions in $\mathcal{P}'$, add u_0 and α to each bag of $\mathcal{P}'$. The resulting pathwidth is still low as only constantly many AG operators have to be replaced in the proof of Lemma 5.1.

For $\mathrm{LTL\text{-}SAT}(\{ \mathsf{X} \})$, proceed exactly like for the CTL case but replace AX by X. As the branching semantics of CTL is not required in the reduction, and in fact AX occurs only positively, the reduction stays correct. □

For the next lemma, we require a result about complexity of modal logic. Define $\mathrm{KD\text{-}ML_1\text{-}SAT}$ as the set of modal formulas ϕ that have at most one propositional variable and are satisfied by a serial Kripke structure.

LEMMA 5.14 ([HALPERN 1995]). $\mathrm{KD\text{-}ML_1\text{-}SAT}$ *is* **PSPACE***-complete.*

The idea is to reduce the number of propositional variables by replacing them by so-called *primitive-proposition-like (pp-like)* formulas. Intuitively, pp-like formulas are "independent enough" from each other so they can be used as an satisfiability-preserving replacement for propositional variables occurring in a modal formula. One can easily

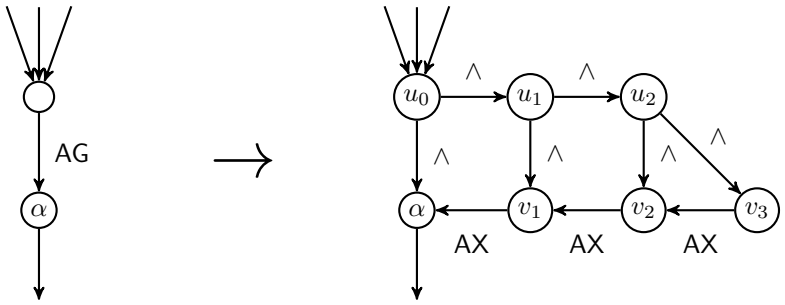


Fig. 6: Circuit transformation from AG to nested AX for $n = 3$.

modify the PSPACE-hardness proof for serial Kripke frames such that the family $\phi_1, \phi_2, \ldots$ of pp-like formulas used by Halpern stays pp-like in such frames.

LEMMA 5.15. $\mathrm{CTL\text{-}SAT}(\{\,\mathsf{AX}\,\})$ *when parametrised by* tw_S *is complete for* para-**PSPACE**.

PROOF. A syntax tree with only one proposition has treewidth at most two, hence the problem is para-**PSPACE**-hard by a simple reduction from $\mathrm{KD\text{-}ML_1\text{-}SAT}$. As $\mathrm{KD\text{-}ML\text{-}SAT}$ or equivalently $\mathrm{CTL\text{-}SAT}(\{\,\mathsf{AX}\,\})$ is in **PSPACE**, the completeness follows. □

We saw that for the fragment $\mathcal{CTL}(\{\,\mathsf{AX}\,\})$ (or equivalently modal logic on serial frames) to be in **FPT**, and the same for $\mathcal{LTL}(\{\,\mathsf{X}\,\})$ with syntax circuit representation, we need the temporal depth as an additional parameter. In the CTL case this is equivalent to the results in modal logic [Praveen 2013]. As satisfiable LTL formulas are already satisfied on paths and LTL is less expressive than modal logic, this extra parameter is not required for the $\mathcal{LTL}(\{\mathsf{X}\})$ fragment in the syntax tree representation.

6. PARAMETRISED COMPLEXITY OF SATISFIABILITY IN POST'S LATTICE

In this section we consider subclones of BF from Post's lattice. We determine for which clones the induced temporal satisfiability problem becomes easier, depending on the investigated parametrisation.

We show that the satisfiability problem stays $\mathbf{W}[1]$-hard for a clone $[B]$ as long as $[B]$ is a superclone of S_1. The same behaviour is shown by propositional logic, where the satisfiability problem stays **NP**-hard for superclones of S_1, but is in **P** for the other clones [Lewis 1979]. Analogously we show that the problem is fixed-parameter tractable in the other cases, except for the open cases $\mathsf{L} = [\{\oplus, \top\}]$ and $\mathsf{L}_0 = [\{\oplus\}]$. Here, the algorithm solving the propositional case cannot be transferred easily.

LEMMA 6.1. *Let* $\kappa \in \{\, tw_{\mathcal{S}}, pw_{\mathcal{S}}, tw_{\mathcal{C}}, pw_{\mathcal{C}} \,\}$*,* T *be a set of CTL-operators,* T' *be a set of LTL-operators, and* B *be a finite set of Boolean functions. If* $\mathsf{S}_1 \subseteq [B]$*, then*

$$\mathrm{CTL\text{-}SAT}(\{\,\wedge, \vee, \neg\,\}, T, \kappa + td) \leq^{fpt} \mathrm{CTL\text{-}SAT}(B, T, \kappa + td),$$
$$\mathrm{LTL\text{-}SAT}(\{\,\wedge, \vee, \neg\,\}, T', \kappa + td) \leq^{fpt} \mathrm{LTL\text{-}SAT}(B, T', \kappa + td).$$

PROOF. For the reduction we use a similar idea as the one used by Lück [2015], and utilise the fact that $\mathsf{BF} = [\{\,\wedge, \vee, \neg\,\}] = [\mathsf{S}_1 \cup \{\,\top\,\}] = [B \cup \{\,\top\,\}]$ for sets B as in the claim of the lemma. The given formula ϕ is first brought into *negation normal form (NNF)*,

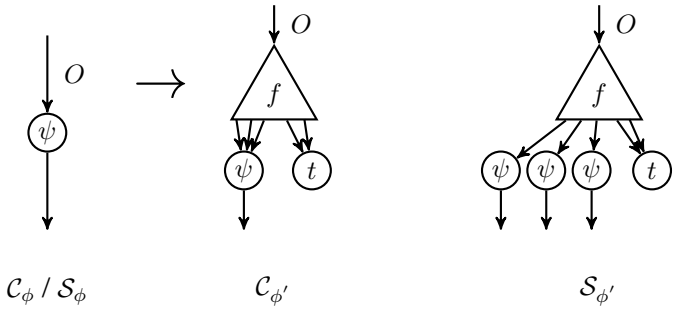


Fig. 7: Transformation of the syntax tree and syntax circuit for $c = 3$.

i.e., such that its negation symbols $\neg$ occur only in front of propositions. Furthermore, we can rewrite ϕ to a logically equivalent formula over the base $B \cup \{\top\}$ in polynomial time.

As $\mathsf{E}_0 \subseteq \mathsf{S}_1$, we deduce $\wedge \in [B]$. Then, to reduce the formula to the base B, we simulate the constant $\top$ by a new variable t, similarly as in propositional logic case [Lewis 1979]. To simulate $\top$ with t, we add "$\wedge t$" to certain subformulas $\alpha \in \mathrm{SF}(\phi)$ of ϕ. Then $\top$ can be replaced by t, as t can be assumed true in every world of a model of ϕ due to ϕ being in NNF. The relevant subformulas to add "$\wedge t$" to are exactly those which are directly under a temporal operator; we transform each $O\alpha \in \mathrm{SF}(\phi)$ to $O(\alpha \wedge t)$ for unary $O \in T$ (resp., $O \in T'$), and $\alpha O \beta \in \mathrm{SF}(\phi)$ to $(\alpha \wedge t) o (\beta \wedge t)$ for binary $O \in T$ (resp., $O \in T'$). Let ϕ^* be the formula obtained from ϕ by performing these substitutions, and appending $\wedge t$ to ϕ itself. Using the base B, we represent $\alpha \wedge t$ as some fixed function

$$f(\underbrace{\alpha, \ldots, \alpha}_{c \text{ times}}, \underbrace{t, \ldots, t}_{c' \text{ times}}), \tag{1}$$

where f is a function composed of symbols of B and $c, c' \in \mathbb{N}$ are constants depending on the base B. In general $c, c' > 1$ as there does not necessarily exist a short representation of $\wedge$ in B (in which every argument would occur only once). The blowup factor of the formula ϕ^* is $c^{\mathrm{td}+1}$ and therefore exponential in general, but only FPT with parameter td, i.e., $|\phi^*| = |\phi| \cdot c^{\mathrm{td}+1}$, which is also a bound for the runtime of the reduction to within a polynomial factor. That ϕ is satisfiable if and only if ϕ^* is satisfiable is shown as in [Lück 2015]. An example for a transformed structure is shown in Figure 7.

In the following, we will show that the substitution does not increase the parameter $\kappa + \mathrm{td}(\phi)$ too much. Clearly the temporal depth is unchanged. It remains to show the boundedness of the pathwidth and treewidth of the transformed formula ϕ^*. We assume the case where we represent ϕ^* as a syntax tree $\mathcal{S}_{\phi^*}$, the case of a syntax circuit is proven analogously. In place of each $\alpha \wedge t$, $\alpha \in \mathrm{SF}(\phi)$, we obtain a subformula of the form of equation (1). Let F be the set of nodes of the local substructure which represents f. To adapt some optimal tree-decomposition or path-decomposition of the syntax tree $\mathcal{S}_\phi$, proceed as follows: For each bag $\mathcal{B}$ and every node $u \in \mathcal{B}$ representing $\wedge$, we require only constantly more variables. Formally, define the new bag as $\mathcal{B}^* := \mathcal{B} \cup \{\, t \,\} \cup \{\, u \mid u \in F$ for some F simulating some $\wedge$ in $\mathcal{B} \,\}$. As the size of F is a constant d and depends only on B, we have $|\mathcal{B}^*| \leq d \cdot |\mathcal{B}| + 1$ and therefore the reduction maintains a bounded parameter. □

THEOREM 6.2. *Let B be a finite set of Boolean functions s.t. $[B] \notin \{ \mathsf{L}, \mathsf{L}_0 \}$. Then for $\kappa \in \{ tw_{\mathcal{S}}, pw_{\mathcal{S}}, tw_{\mathcal{C}}, pw_{\mathcal{C}} \}$ the problems* CTL-SAT(B), LTL-SAT(B) *and* CTL*-SAT(B) *parametrised by $\kappa + td$ are*

(*1*) $\mathbf{W}[1]$*-hard if* $\mathsf{S}_1 \subseteq [\mathcal{B}]$,
(*2*) *in* **FPT** *otherwise.*

PROOF. Follows from Meier et al. [2009] for the fixed-parameter tractable cases as already the non-parametrised cases are tractable, and from Lemma 5.1 and 5.10 for the intractable cases. □

7. CONCLUSION

Table I: Overview of the parametrised complexity of CTL, LTL, and CTL*. The empty operator set is not mentioned as its complexity is equivalent to the one of SAT. Numbers in the exponent refer to the corresponding lemma, theorem, or corollary in the paper. The notion tw, resp., pw indicates both syntax circuits and trees unless otherwise stated.

Problem Q	Parameter κ		
CTL-SAT$(\cdot)$	td	tw / pw	td + tw / td + pw
AX	para-**NP**-h.$^{5.11}$	$(\ldots)^b$	**FPT**$^{4.7}$
AF	para-**PSPACE**-c.$^{5.11}$	$\mathbf{W}[1]$-h.$^{5.7}$	$\mathbf{W}[1]$-h.$^{5.7,a}$
AF, AX	para-**PSPACE**-c.$^{5.11}$	$\mathbf{W}[1]$-h.$^{5.4,c}$	$\mathbf{W}[1]$-h.$^{5.4}$
AG	para-**PSPACE**-c.$^{5.11}$	$\mathbf{W}[1]$-h.$^{5.5}$	$\mathbf{W}[1]$-h.$^{5.5}$
other	para-**EXPTIME**-c.$^{5.11}$	$\mathbf{W}[1]$-h.$^{5.1,5.5,5.6}$	$\mathbf{W}[1]$-h.$^{5.1,5.5,5.6}$
LTL-SAT$(\cdot)$	td	tw / pw	td + tw / td + pw
X	para-**NP**-c.$^{5.12}$	$(\ldots)^d$	**FPT**$^{4.7}$
F	para-**NP**-c.$^{5.12}$	$\mathbf{W}[1]$-h.$^{5.10}$	$\mathbf{W}[1]$-h.$^{5.10}$
other	para-**PSPACE**-c.$^{5.12}$	$\mathbf{W}[1]$-h.$^{5.10}$	$\mathbf{W}[1]$-h.$^{5.10}$
CTL*-SAT$(\cdot)$	td	tw / pw	td + tw / td + pw
A, X	para-**NP**-h.$^{5.12}$	$(\ldots)^b$	**FPT**$^{4.6}$
other	para-**EXPTIME**-h.$^{5.11}$	$\mathbf{W}[1]$-h.$^{5.10}$	$\mathbf{W}[1]$-h.$^{5.10}$

[a] Only for $\mathcal{C}$, open for $\mathcal{S}$.
[b] para-**PSPACE**-c. for $\text{tw}_{\mathcal{S}}$ $^{5.15}$, $\mathbf{W}[1]$-h. for $\text{tw}_{\mathcal{C}}$ and $\text{pw}_{\mathcal{C}}$ $^{5.13}$, open for $\text{pw}_{\mathcal{S}}$.
[c] para-**PSPACE**-c. for $\text{tw}_{\mathcal{S}}$ [Lück 2015].
[d] **FPT** for $\text{tw}_{\mathcal{S}}$ and $\text{pw}_{\mathcal{S}}$ $^{4.8}$, $\mathbf{W}[1]$-h. for $\text{tw}_{\mathcal{C}}$ and $\text{pw}_{\mathcal{C}}$ $^{5.13}$

In this work, we presented an almost complete classification with respect to parametrised complexity of all possible operator fragments of temporal satisfiability problems. The considered temporal logics are linear temporal logic LTL, computation tree logic CTL and the full branching time logic CTL*. The problems are parametrised by temporal depth and different notions of pathwidth and treewidth. We have also given an almost complete classification with respect to the Boolean fragments in Post's lattice.

We have shown that the known results for modal logic, i.e., the possibility of applying Courcelle's theorem [Praveen 2013], carry over to all three considered temporal logics when restricted to X operators. Therefore, these fragments are fixed-parameter tractable for the given parameters. This even holds for CTL* which is strictly more powerful than the modal logic KD as in CTL* path formulas and state formulas can be combined freely.

Furthermore, we have shown that every other temporal operator leads to an increase in expressive power such that their satisfiability problem is $\mathbf{W}[1]$-hard and presumably cannot be expressed anymore by $\mathcal{MSO}$ formulas of bounded length. The results seem to imply a direct correlation between fixed-parameter tractability and the property of having shallow tree models.

To loosen the restriction of formulas to CNF, two new representations of formulas as relational structures have been investigated, namely syntax trees and syntax circuits. Large parts of the results do not distinguish between the representations in terms of parametrised complexity. On syntax trees, only the $\{ \mathsf{AF} \}$ case is still open, but the authors conjecture another $\mathbf{W}[1]$-hard fragment here.

Without the temporal depth as parameter, the situation quickly becomes more complicated. It is open if the $\{ \mathsf{AX} \}$, resp., $\{ \mathsf{A}, \mathsf{X} \}$ fragment is $\mathbf{FPT}$ with only the syntax tree pathwidth parameter. With syntax tree treewidth, or when represented as syntax circuits, it is $\mathbf{W}[1]$-hard, but neither result implies the same for syntax tree pathwidth. It would not be the only fragment where the exact representation makes a difference: Very interestingly, LTL with only X is in $\mathbf{FPT}$ on syntax trees but $\mathbf{W}[1]$-hard on syntax circuits, which is a hint that circuits could have smaller width than trees in general.

Another consequent step will be the investigation of other parametrisations beyond the usual considered measures of pathwidth or treewidth and temporal depth. Further finding matching upper bounds to state completeness results may lead to a better understanding of (different levels of) intractability not only in the parametrised sense. The role of relational structures of formulas and their possible parametrisations seem to be important for complexity theoretic aspects in general and should be investigated. Moreover the exploration of other parametrised decision problems like model checking in temporal logic is one of our next research steps.

ACKNOWLEDGMENT

The authors are thankful to Anselm Haak (Hannover) and Maurice Chandoo (Hannover), as well to the anonymous referees, for their valuable corrections and hints.

REFERENCES

E. Allen Emerson. 1990. Temporal and Modal Logic. In *Handbook of Theoretical Computer Science (Vol. B)*, Jan van Leeuwen (Ed.). MIT Press, Cambridge, MA, USA, 995–1072.

E. Allen Emerson and E. M. Clarke. 1981. Design and synthesis of synchronisation skeletons using branching time temporal logic. In *Logic of Programs (Lecture Notes in Computer Science)*, Vol. 131. Springer Verlag, 52–71.

E. Allen Emerson and J. Y. Halpern. 1985. Decision procedures and expressiveness in the temporal logic of branching time. *J. Comput. System Sci.* 30, 1 (1985), 1–24.

M. Bauland, E. Böhler, N. Creignou, S. Reith, H. Schnoor, and H. Vollmer. 2010. The Complexity of Problems for Quantified Constraints. *Theory Computing Systems* 47 (2010), 454–490.

O. Beyersdorff, A. Meier, M. Mundhenk, T. Schneider, M. Thomas, and H. Vollmer. 2011. Model Checking CTL is almost always inherently sequential. *Logical Methods in Computer Science* 7, 2 (2011).

O. Beyersdorff, A. Meier, M. Thomas, and H. Vollmer. 2010. The Complexity of Reasoning for Fragments of Default Logic. *Journal of Logic and Computation* (2010).

P. Blackburn, M. de Rijke, and Y. Venema. 2001. *Modal logic*. Cambridge University Press, New York, NY, USA.

E. Böhler, N. Creignou, M. Galota, S. Reith, H. Schnoor, and H. Vollmer. 2012. Complexity classifications for different equivalence and audit problems for Boolean circuits. *Logical Methods in Computer Science* 8, 3:27 (2012), 1–25.

C. Chekuri and A. Rajaraman. 1997. Conjunctive query containment revisited. In *Database Theory — ICDT '97*. Vol. 1186. Springer Berlin Heidelberg, Berlin, Heidelberg, 56–70.

B. Courcelle and J. Engelfriet. 2012. *Graph structure and monadic second-order logic, a language theoretic approach*. Cambridge University Press.

N. Creignou, A. Meier, H. Vollmer, and M. Thomas. 2012. The Complexity of Reasoning for Fragments of Autoepistemic Logic. *ACM Transactions on Computational Logic* 13, 2 (April 2012), 1–22.

S. Demri and P. Schnoebelen. 2002. The Complexity of Propositional Linear Temporal Logics in Simple Cases. *Information and Computation* 174, 1 (April 2002), 84–103.

R. G. Downey and M. R. Fellows. 1999. *Parameterized Complexity*. Springer-Verlag. 530 pp.

M. Elberfeld, A. Jakoby, and T. Tantau. 2010. Logspace Versions of the Theorems of Bodlaender and Courcelle. In *Proc. 51th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society.

M. J. Fischer and R. E. Ladner. 1979. Propositional dynamic logic of regular programs. *J. Comput. System Sci.* 18, 2 (1979), 194 – 211.

J. Flum and M. Grohe. 2006. *Parameterized Complexity Theory*. Springer Verlag.

Haim Gaifman. 1982. On Local and Non-local Properties. In *Herbrand Symposium, Logic Colloquium'81*. North-Holland, 105–135.

G. Gottlob, R. Pichler, and F. Wei. 2010. Bounded treewidth as a key to tractability of knowledge representation and reasoning. *Artificial Intelligence* 174, 1 (2010), 105–132.

J.Y. Halpern. 1995. The Effect Of Bounding The Number Of Primitive Propositions And The Depth Of Nesting On The Complexity Of Modal Logic. *Artificial Intelligence* 75 (1995), 361–372.

E. Hemaspaandra, H. Schnorr, and I. Schnoor. 2010. Generalized modal satisfiability. *J. Comput. System Sci.* 76 (2010), 561–578.

P.G. Kolaitis and M.Y. Vardi. 2000. Conjunctive-Query Containment and Constraint Satisfaction. 61, 2 (2000), 302–332.

S. Kripke. 1963. Semantical Considerations on Modal Logic. In *Acta Philosophica Fennica*, Vol. 16. 84–94.

H. Lewis. 1979. Satisfiability problems for propositional calculi. *Mathematical Systems Theory* 13 (1979), 45–53.

M. Lück. 2015. Quirky Quantifiers: Optimal Models and Complexity of Computation Tree Logic. *CoRR* abs/1510.08786 (2015).

M. Lück, A. Meier, and I. Schindler. 2015. Parameterized Complexity of CTL. In *Language and Automata Theory and Applications*. Lecture Notes in Computer Science, Vol. 8977. Springer International Publishing, 549–560.

A. Meier, M. Mundhenk, T. Schneider, M. Thomas, V. Weber, and F. Weiss. 2010. The Complexity of Satisfiability for Fragments of Hybrid Logic – Part I. *Journal of Applied Logic* 8, 4 (2010), 409–421.

A. Meier, M. Mundhenk, M. Thomas, and H. Vollmer. 2009. The Complexity of Satisfiability for Fragments of CTL and CTL*. *International Journal of Foundations of Computer Science* 20, 05 (2009), 901–918. Erratum see [Meier et al. 2015].

A. Meier, M. Mundhenk, M. Thomas, and H. Vollmer. 2015. Erratum: The Complexity of Satisfiability for Fragments of CTL and CTL*. *International Journal of Foundations of Computer Science* 26, 08 (2015), 1189–1190. DOI:http://dx.doi.org/10.1142/S012905411592001X

A. Meier, J. Schmidt, M. Thomas, and H. Vollmer. 2012. On the Parameterized Complexity of Default Logic and Autoepistemic Logic. In *Proc. 6th International Conference on Language and Automata Theory and Applications (LATA) (LNCS)*, Vol. 7183. 389–400.

A. Meier and T. Schneider. 2013. Generalized satisfiability for the description logic ALC. *Theoretical Computer Science* 505, 0 (2013), 55 – 73. Theory and Applications of Models of Computation 2011.

N. Pippenger. 1997. *Theories of Computability*. Cambridge University Press.

A. Pnueli. 1977. The Temporal Logic of Programs. In *Proc. 18th Symposium on Foundations of Computer Science*. IEEE Computer Society Press, 46–57.

E. Post. 1941. The two-valued iterative systems of mathematical logic. *Annals of Mathematical Studies* 5 (1941), 1–122.

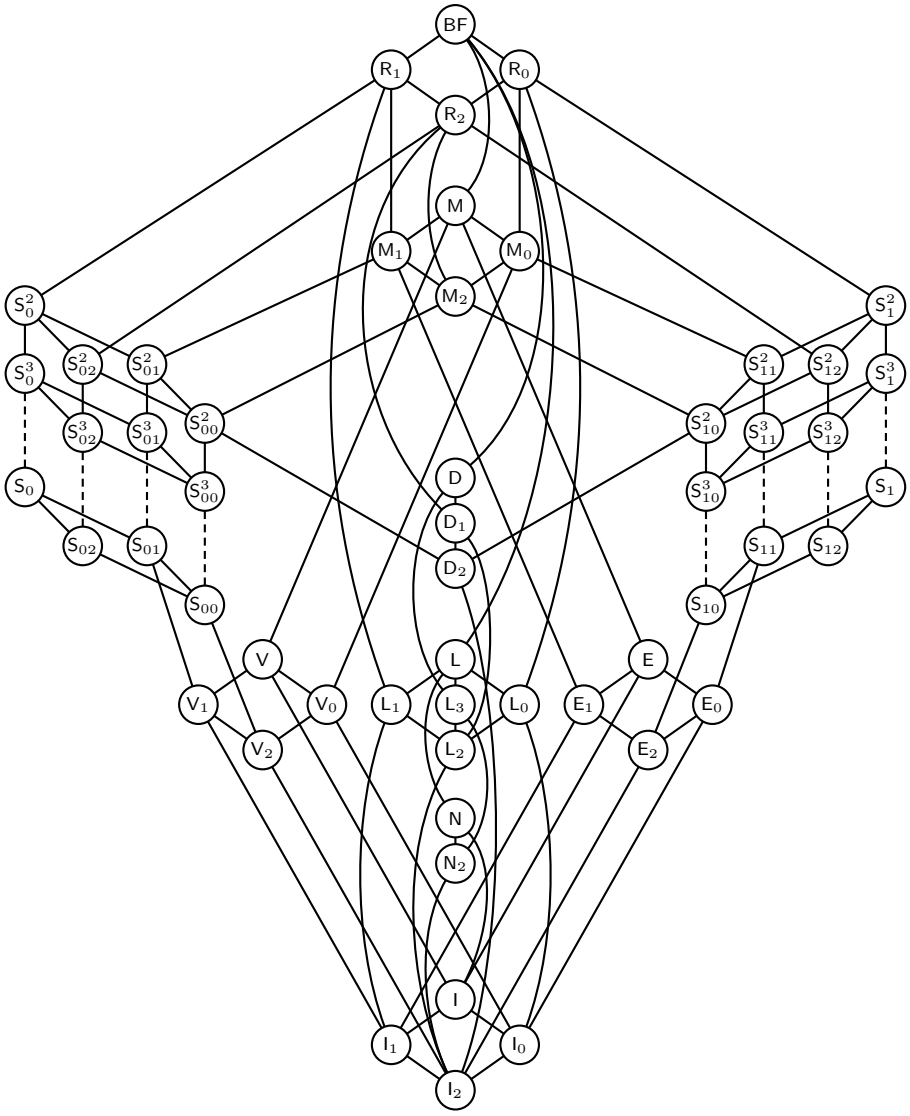
M. Praveen. 2013. Does Treewidth Help in Modal Satisfiability? *ACM Transactions on Computational Logic* 14, 3 (2013), 18:1–18:32. DOI:http://dx.doi.org/10.1145/2499937.2499939

A. N. Prior. 1957. *Time and Modality*. Clarendon Press, Oxford.

M. Samer and S. Szeider. 2006. A Fixed-Parameter Algorithm for #SAT with Parameter Incidence Treewidth. *CoRR* abs/cs/0610174 (2006).

M. Samer and S. Szeider. 2010. Constraint satisfaction with bounded treewidth revisited. *J. Comput. System Sci.* 76, 2 (2010), 103 – 114.

A. POST'S LATTICE AND LIST OF CLONES



Class	Definition	Base
BF	All Boolean functions	$\{ x \wedge y, \neg x \}$
R_0	$\{ f \mid f$ is $\bot$-reproducing $\}$	$\{ x \wedge y, x \oplus y \}$
R_1	$\{ f \mid f$ is $\top$-reproducing $\}$	$\{ x \vee y, x \leftrightarrow y \}$
R_2	$\mathsf{R}_0 \cap \mathsf{R}_1$	$\{ \vee, x \wedge (y \leftrightarrow z) \}$
M	$\{ f \mid f$ is monotone $\}$	$\{ x \vee y, x \wedge y, \bot, \top \}$
M_0	$\mathsf{M} \cap \mathsf{R}_0$	$\{ x \vee y, x \wedge y, \bot \}$
M_1	$\mathsf{M} \cap \mathsf{R}_1$	$\{ x \vee y, x \wedge y, \top \}$
M_2	$\mathsf{M} \cap \mathsf{R}_2$	$\{ x \vee y, x \wedge y \}$
S_0	$\{ f \mid f$ is $\bot$-separating $\}$	$\{ x \rightarrow y \}$
S_1	$\{ f \mid f$ is $\top$-separating $\}$	$\{ x \nrightarrow y \}$
S_0^n	$\{ f \mid f$ is $\bot$-separating of degree $n \}$	$\{ x \rightarrow y, \mathrm{dual}(T_n^{n+1}) \}$
S_1^n	$\{ f \mid f$ is $\top$-separating of degree $n \}$	$\{ x \nrightarrow y, T_n^{n+1} \}$
S_{00}	$\mathsf{S}_0 \cap \mathsf{R}_2 \cap \mathsf{M}$	$\{ x \vee (y \wedge z) \}$
S_{00}^n	$\mathsf{S}_0^n \cap \mathsf{R}_2 \cap \mathsf{M}$	$\{ x \vee (y \wedge z), \mathrm{dual}(T_n^{n+1}) \}$
S_{01}	$\mathsf{S}_0 \cap \mathsf{M}$	$\{ x \vee (y \wedge z), \top \}$
S_{01}^n	$\mathsf{S}_0^n \cap \mathsf{M}$	$\{ \mathrm{dual}(T_n^{n+1}), \top \}$
S_{02}	$\mathsf{S}_0 \cap \mathsf{R}_2$	$\{ x \vee (y \nrightarrow z) \}$
S_{02}^n	$\mathsf{S}_0^n \cap \mathsf{R}_2$	$\{ x \vee (y \nrightarrow z), \mathrm{dual}(T_n^{n+1}) \}$
S_{10}	$\mathsf{S}_1 \cap \mathsf{R}_2 \cap \mathsf{M}$	$\{ x \wedge (y \vee z) \}$
S_{10}^n	$\mathsf{S}_1^n \cap \mathsf{R}_2 \cap \mathsf{M}$	$\{ x \wedge (y \vee z), T_n^{n+1} \}$
S_{11}	$\mathsf{S}_1 \cap \mathsf{M}$	$\{ x \wedge (y \vee z), \bot \}$
S_{11}^n	$\mathsf{S}_1^n \cap \mathsf{M}$	$\{ T_n^{n+1}, \bot \}$
S_{12}	$\mathsf{S}_1 \cap \mathsf{R}_2$	$\{ x \wedge (y \rightarrow z) \}$
S_{12}^n	$\mathsf{S}_1^n \cap \mathsf{R}_2$	$\{ x \wedge (y \rightarrow z), T_n^{n+1} \}$
D	$\{ f \mid f$ is self-dual $\}$	$\{ (x \nrightarrow y) \vee (x \nrightarrow z) \vee (\overline{y} \nrightarrow z) \}$
D_1	$\mathsf{D} \cap \mathsf{R}_2$	$\{ (x \nrightarrow \overline{y}) \vee (x \nrightarrow z) \vee (y \nrightarrow z) \}$
D_2	$\mathsf{D} \cap \mathsf{M}$	$\{ (x \nrightarrow \overline{y}) \vee (x \nrightarrow \overline{z}) \vee (y \nrightarrow \overline{z}) \}$
L	$\{ f \mid f$ is linear $\}$	$\{ x \oplus y, \top \}$
L_0	$\mathsf{L} \cap \mathsf{R}_0$	$\{ x \oplus y \}$
L_1	$\mathsf{L} \cap \mathsf{R}_1$	$\{ x \leftrightarrow y \}$
L_2	$\mathsf{L} \cap \mathsf{R}_2$	$\{ x \oplus y \oplus z \}$
L_3	$\mathsf{L} \cap \mathsf{D}$	$\{ x \oplus y \oplus z \oplus \top \}$
V	$\{ f \mid f$ is a disjunction or constant $\}$	$\{ x \vee y, \bot, \top \}$
V_0	$\mathsf{M}_0 \cap \mathsf{V}$	$\{ x \vee y, \bot \}$
V_1	$\mathsf{M}_1 \cap \mathsf{V}$	$\{ x \vee y, \top \}$
V_2	$\mathsf{M}_2 \cap \mathsf{V}$	$\{ x \vee y \}$
E	$\{ f \mid f$ is a conjunction or constant $\}$	$\{ x \wedge y, \bot, \top \}$
E_0	$\mathsf{M}_0 \cap \mathsf{E}$	$\{ x \wedge y, \bot \}$
E_1	$\mathsf{M}_1 \cap \mathsf{E}$	$\{ x \wedge y, \top \}$
E_2	$\mathsf{M}_2 \cap \mathsf{E}$	$\{ x \wedge y \}$
N	$\{ f \mid f$ depends on at most one variable $\}$	$\{ \neg x, \bot, \top \}$
N_2	$\mathsf{L}_3 \cap \mathsf{N}$	$\{ \neg x \}$
I	$\{ f \mid f$ is a projection or a constant $\}$	$\{ \mathrm{id}, \bot, \top \}$
I_0	$\mathsf{R}_0 \cap \mathsf{I}$	$\{ \mathrm{id}, \bot \}$
I_1	$\mathsf{R}_1 \cap \mathsf{I}$	$\{ \mathrm{id}, \top \}$
I_2	$\mathsf{R}_2 \cap \mathsf{I}$	$\{ \mathrm{id} \}$

Table II: A list of all Boolean clones with definitions and bases.