Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 134 (2014)

Parameterized Complexity of CTL: A Generalization of Courcelle's Theorem

Martin Lück, Arne Meier, and Irina Schindler*

Institut für Theoretische Informatik
Leibniz Universität Hannover
{lueck, meier, schindler}@thi.uni-hannover.de

Abstract. We present an almost complete classification of the parameterized complexity of all operator fragments of the satisfiability problem in computation tree logic CTL. The investigated parameterization is temporal depth and pathwidth. The classification shows a dichotomy between W[1]-hard and fixed-parameter tractable fragments. The only real operator fragments which is in FPT is the fragment containing solely AX. Also we prove a generalization of Courcelle's theorem to infinite signatures which will be used to proof the FPT-membership cases.

1 Introduction

Temporal logic is the most important concept in computer science in the area of program verification and is a widely used concept to express specifications. Introduced in the late 1950s by Prior [1] a large area of research has been evolved up to today. Here the most seminal contributions have been made by Kripke [2], Pnueli [3], Emerson, Clarke, and Halpern [4,5] to name only a few. The maybe most important temporal logic so far is the computation tree logic CTL due to its polynomial time solvable model checking problem which influenced the area of program verification significantly. However the satisfiability problem, i.e., the question whether a given specification is consistent, is beyond tractability, i.e., complete for deterministic exponential time. One way to attack this intrinsic hardness is to consider restrictions of the problem by means of operator fragments leading to a trichotomy of computational complexity shown bei Meier [6]. This landscape of intractability depicted completeness results for nondeterministic polynomial time, polynomial space, and (of course) deterministic exponential time showing how combinations of operators imply jumps in computational complexity of the corresponding satisfiability fragment.

For more than a decade now there exists a theory which allows us to better understand the structure of intractability: 1999 Downey and Fellows developed the area of parameterized complexity [7] and up to today this field has grown vastly. Informally the main idea is to detect a specific part of the problem, the *parameter*, such that the intractability of the problems complexity vanishes if the parameter is assumed to be constant. Through this approach the notion of *fixed*

* Supported in part by DFG ME 4279/1-1.

ISSN 1433-8092

parameter tractability has been founded. A problem is said to be fixed parameter tractable (or short, FPT) if there exists a deterministic algorithm running in time $f(k) \cdot poly(n)$ for all input lengths n, corresponding parameter values k, and a recursive function f. As an example, the usual propositional logic satisfiability problem SAT (well-known to be NP-complete) becomes fixed parameter tractable under the parameter number of variables.

In this work we almost completely classify the parameterized complexity of all operator fragments of the satisfiability problem for the computation tree logic CTL under the parameterization of formula pathwidth and temporal depth. Only the case for AF resisted a full classification. We will explain the reasons in the conclusion. For all other fragments we show a dichotomy consisting of two fragments being fixed parameter tractable and the remainder being hard for the complexity class **W**[1] under fpt-reductions. **W**[1] can be seen as an analogue of intractability in the decision case in the parameterized world. To obtain this classification we prove a generalization of Courcelle's theorem [8] for *infinite* signatures which may be of independent interest.

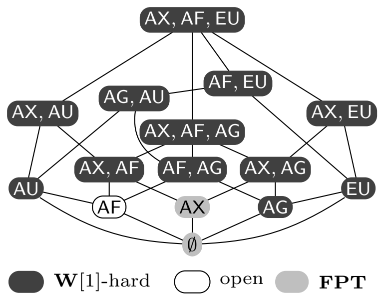

Fig. 1. Parameterized complexity of CTL-SAT($\mathcal{T}$) parameterized by formula pathwidth and temporal depth (see Theorem 1).

Related work. Similar research for modal logic has been done by Praveen and influenced the present work in some parts [9]. Other applications of Courcelle's theorem have been investigated by Meier et al. [10] and Gottlob et al. [11]. In 2010 Elberfeld et al. proved that Courcelle's theorem can be extended to give results in XL as well [12] wherefore the results of Corollary 4 can be extended to this class, too.

2 Preliminaries

We assume familiarity with standard notions of complexity theory as Turing machines, reductions, the classes **P** and **NP**. For an introduction into this field we confer the reader to the very good textbook of Pippenger [13].

2.1 Complexity Theory

Let Σ be an alphabet. A pair $\Pi = (Q, \kappa)$ is a *parameterized 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 a *parameterization of Π* or the *parameter of Π*. We say a parameterized problem Π is *fixed-parameter tractable* (or in the class **FPT**) if there exists a deterministic algorithm deciding Π in time

$f(\kappa(x)) \cdot |x|^{O(1)}$ for every $x \in \Sigma^*$ and a recursive function f. Note that the notion of fixed-parameter tractability is easily extended beyond decision problems.

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

1. For all $x \in \Sigma^*$ it holds $x \in Q$ 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 $\kappa'(r(x)) \leq g(\kappa(x))$.

The class $\mathbf{W}[1]$ is a parameterized complexity class which plays a similar role as **NP** in the sense of intractability in the parameterized world. The class $\mathbf{W}[1]$ is a superset of **FPT** and a hierarchy of other **W**-classes are build above of 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 text book of Flum and Grohe [14].

2.2 Tree- and Pathwidth

Given a structure $\mathcal{A}$ we define a *tree decomposition of* $\mathcal{A}$ (with universe A) to be a pair (T, X) where $X = \{B_1, \ldots, B_r\}$ is a family of subsets of A (the set of *bags*), and T is a tree whose nodes are the bags B_i satisfying the following conditions:

1. Every element of the universe appears in at least one bag: $\bigcup X = A$.
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 $B \in X$ such that $\{a_1, \ldots, a_k\} \in B$.
3. For every element a the set of bags containing a is connected: for all $a \in A$ the set $\{B \mid a \in B\}$ forms a connected subtree in T.

The *width* of a decomposition (T, X) is $\mathrm{width}(T, X) := \max\{|B| \mid B \in X\} - 1$ 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 the tree-likeliness of it. The closer the value is to 1 the more the structure is a tree.

A *path decomposition* of a structure $\mathcal{A}$ is similarly defined to tree decompositions however T has to be a path. Here $\mathrm{pw}(\mathcal{A})$ denotes the *pathwidth* of $\mathcal{A}$. Likewise the size of the pathwidth describes the similarity of a structure to a path. Observe that pathwidth bounds treewidth from above.

2.3 Logic

Let Φ be a finite set of propositional letters. A *propositional formula* ($\mathcal{PL}$ formula) is inductively defined as follows. The constants $\top, \bot$, (true, false) and any *propositional letter* (or *proposition*) $p \in \Phi$ are $\mathcal{PL}$ formulas. If ϕ, ψ are $\mathcal{PL}$ formulas then so are $\phi \wedge \psi, \neg\phi, \phi \vee \psi$ with their usual semantics (we further use the shortcuts $\rightarrow, \leftrightarrow$). Temporal logic extends propositional logic by introducing four *temporal operators*, i.e., *next* X, *future* F, *globally* G, and *until* U. Together with the two *path quantifiers*, *exists* E and *all* A, they fix the set of *computation tree logic formulas* ($\mathcal{CTL}$ formulas) as follows. If $\phi \in \mathcal{PL}$ then $\mathsf{PT}\phi, \mathsf{P}[\phi \mathsf{U} \psi] \in \mathcal{CTL}$ and if $\phi, \psi \in \mathcal{CTL}$ then $\mathsf{PT}\phi, \mathsf{P}[\phi \mathsf{U} \psi], \phi \vee \psi, \neg\psi, \phi \wedge \psi \in \mathcal{CTL}$ hold, where $\mathsf{P} \in \{\mathsf{A}, \mathsf{E}\}$ is a path quantifier and $\mathsf{T} \in \{\mathsf{X}, \mathsf{F}, \mathsf{G}\}$ is a temporal operator. The pair of a single path quantifier and a single temporal operator is referred to as a CTL-operator. If T is a set of CTL-operators then $\mathcal{CTL}(T)$ is the restriction of $\mathcal{CTL}$ to formulas that are allowed to use only CTL-operators from T.

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 *total successor relation* $R \colon W \to W$ (i.e., for every $w \in W$ there exists a $w' \in W$ with wRw'), and an *evaluation function* $V \colon W \to 2^{\Phi}$ labeling 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 $\mathfrak{P}(w)$ the set of all paths starting at w. For $\mathcal{CTL}$ formulas we define the semantics of $\mathcal{CTL}$ formulas ϕ, ψ for a given Kripke structure $K = (W, R, V)$, a world $w \in W$, and a path π as

$$
\begin{aligned}
K, w \models \mathsf{AT}\phi \quad &\Leftrightarrow \quad \text{for all } \pi \in \mathfrak{P}(w) \text{ it holds } K, \pi \models \mathsf{T}\phi,\\
K, w \models \mathsf{ET}\phi \quad &\Leftrightarrow \quad \text{there exists a } \pi \in \mathfrak{P}(w) \text{ it holds } K, \pi \models \mathsf{T}\phi,\\
K, \pi \models \mathsf{X}\phi \quad &\Leftrightarrow \quad K, \pi(1) \models \phi,\\
K, \pi \models \mathsf{F}\phi \quad &\Leftrightarrow \quad \text{there exists an } i \geq 0 \text{ such that } K, \pi(i) \models \phi,\\
K, \pi \models \mathsf{G}\phi \quad &\Leftrightarrow \quad \text{for all } i \geq 0\ K, \pi(i) \models \phi,\\
K, \pi \models \phi \mathsf{U} \psi \quad &\Leftrightarrow \quad \exists i \geq 0 \forall j < i\ K, \pi(j) \models \phi \text{ and } K, \pi(i) \models \psi.
\end{aligned}
$$

For a formula $\phi \in \mathcal{CTL}$ we define the satisfiability problem CTL-SAT asking if there exists a Kripke structure $K = (W, R, V)$ and $w \in W$ such that $K, w \models \phi$. Then we also say that M is a *model (of* ϕ. Similar to before CTL-SAT(T) is the restriction of CTL-SAT to formulas in $\mathcal{CTL}(T)$ for a set of CTL-operators T. A formula $\phi \in \mathcal{CTL}$ is said to be in *negation normal form (NNF)* if its negation symbols $\neg$ occur only in front of propositions; we will use the symbol $\mathcal{CTL}_{\mathrm{NNF}}$ to denote the set of CTL-formulas which are in NNF only.

Given $\phi \in \mathcal{CTL}$ we define $\mathrm{SF}(\phi)$ as the *set of all subformulas of* ϕ (containing ϕ itself). The *temporal depth of* ϕ, in symbols $\mathrm{td}(\phi)$, is defined inductively as follows. If Φ is a finite set of propositional symbols and $\phi, \psi \in \mathcal{CTL}$ then

$$
\begin{array}{lll}
\mathrm{td}(p) := 0, & \mathrm{td}(\phi \circ \psi) & := \max\{\mathrm{td}(\phi), \mathrm{td}(\psi)\},\\
\mathrm{td}(\top) := 0, & \mathrm{td}(\neg\phi) & := \mathrm{td}(\phi),\\
\mathrm{td}(\bot) := 0, & \mathrm{td}(\mathsf{PT}\phi) & := \mathrm{td}(\phi) + 1,\\
 & \mathrm{td}(\mathsf{P}[\phi \mathsf{U} \psi]) & := \max\{\mathrm{td}(\phi), \mathrm{td}(\psi)\} + 1,
\end{array}
$$

where $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, $\mathsf{P} \in \{\mathsf{A}, \mathsf{E}\}$, and $\mathsf{T} \in \{\mathsf{X}, \mathsf{F}, \mathsf{G}\}$. If $\psi \in \mathrm{SF}(\phi)$ then the *temporal depth of* ψ *in* ϕ is $\mathrm{td}_{\phi}(\psi) := \mathrm{td}(\phi) - \mathrm{td}(\psi)$.

Vocabularies are *finite* sets of *relation symbols* (or *predicates*) of finite arity $k \geq 1$ (if $k = 1$ then we say the predicate is *unary*) which are usually denoted with the symbol τ. Later we will also refer to similar objects of infinite size wherefore we prefer to denote them with the term *signature* which usually is an countable infinite sized set of symbols. A *structure* $\mathcal{A}$ over a vocabulary (or signature) τ consists of a *universe* A which is a non-empty set, and a relation $P^{\mathcal{A}} \subseteq A^k$ for each predicate P of arity k. Monadic second order logic (MSO) is the restriction of second order logic (SO) in which only quantification over unary relations is allowed (elements of the universe can still be quantified existentially or universally). If P is a unary predicate then $P(x)$ is true if and only if $x \in P$ holds (otherwise it is false).

3 Parameterized Complexity of CTL-SAT(T)

In this section we investigate all operator fragments of CTL-SAT parameterized by temporal depth and formula pathwidth with respect to its parameterized complexity. This means, we the given formulas from $\mathcal{CTL}$ as input are represented by relational structures as follows.

Let $\varphi \in \mathcal{CTL}$ be a $\mathcal{CTL}$ formula. The vocabulary of our interest is τ being defined as $\tau := \{\mathrm{const}^1_f \mid f \in \{\top, \bot\}\} \cup \{\mathrm{conn}^2_{f,i} \mid f \in \{\wedge, \vee, \neg\}, 1 \leq i \leq \mathrm{ar}(f)\} \cup \{\mathrm{var}^1, \mathrm{repr}^1, \mathrm{repr}^1_{\mathsf{PL}}\} \cup \{\mathrm{repr}^1_{\mathsf{C}}, \mathrm{body}^2_{\mathsf{C}} \mid \mathsf{C} \text{ is a CTL-operator}\}$. We then associate the vocabulary τ with the structure $\mathcal{A}_\varphi$ where its universe consists of elements representing subformulas of φ. The predicates are defined as follows

- $\mathrm{var}^1(x)$ holds iff x represents a variable,
- $\mathrm{repr}^1(x)$ holds iff x represents the formula φ,
- $\mathrm{repr}^1_{\mathsf{PL}}(x)$ holds iff x represents a propositional formula,
- $\mathrm{repr}^1_{\mathsf{C}}(x)$ holds iff x represents a formula $\mathsf{C}\psi$ where C is a CTL-operator,
- $\mathrm{body}^2_{\mathsf{C}}(y, x)$ holds iff x represents a formula $\mathsf{C}\psi$ and ψ is represented by y where C is a CTL-operator,
- $\mathrm{const}^1_f(x)$ holds iff x represents the constant of f,
- $\mathrm{conn}^2_{f,i}(x, y)$ holds iff x represents the ith argument of the function f at the root of the formula tree represented by y.

Now we see the problem CTL-SAT parameterized by the pathwidth of its instance structures $\mathcal{A}_\varphi$ (for the instances φ) as well as the temporal depth of the formula. Hence the parameterization function κ maps, given an instance formula $\varphi \in \mathcal{CTL}$ to the pathwidth of the structures $\mathcal{A}_\varphi$ plus the temporal depth of φ, i.e., $\kappa(\varphi) = \mathrm{pw}(\mathcal{A}_\varphi) + \mathrm{td}(\varphi)$.

The following theorem summarizes the collection of results we have proven in the upcoming lemmas. The subsection on page 6 contains the **FPT** result together with the generalization of Courcelle's theorem to infinite signatures.

Theorem 1. CTL-SAT(T) *parameterized by formula pathwidth and temporal depth is*

1. *in* **FPT** *if* $T = \{\mathsf{AX}\}$ *or* $T = \emptyset$, *and*

2. $\mathbf{W}[1]$-*hard if* $\mathsf{AG} \in T$*, or* $\mathsf{AU} \in T$*, or* $\{\mathsf{AX}, \mathsf{AF}\} \subseteq T$.

Proof. (1.) is witnessed by Corollary 4. The proof of (2.) is split into Lemmas 5 to 7. □

One way to prove the containment of a problem parameterized in that way in the class **FPT** is to use the prominent result of Courcelle [8, Thm. 6.3 (1)]. Informally, satisfiability of CTL-formulas therefore has to be formalized in monadic second order logic. The other ingredient of this approach is expressing formulas by relational structures as described before. Now the crux is that our case requires a family of MSO formulas which depend on the instance. This however seems to be a serious issue at first sight as this prohibits the application of Courcelle's theorem. Fortunately we are able to generalize Courcelle's theorem to circumvent this problem. Moreover we generalized it to work with infinite sized signatures under a specific restrictions which allows us to state the desired **FPT** result described as follows.

A Generalized Version of Courcelle's Theorem

Assume we are able to express a problem Q in MSO. If instances $x \in Q$ can be modeled via some relational structure $\mathcal{A}_x$ over some finite vocabulary τ and we see Q as a parameterized problem (Q, κ) where κ is the treewidth of $\mathcal{A}_x$ then by Courcelle's theorem we immediately obtain that (Q, κ) is in **FPT** [8]. If we do not have a fixed MSO formula (which is independent of the instance) then we are not able to use the mentioned result. However the following theorem shows how it is possible even with infinite signatures to apply the result of Courcelle. For this, we assume that the problem can be expressed by an infinite family $(\phi_n)_{n \in \mathbb{N}}$ of MSO-formulas along with the restriction that $(\phi_n)_{n \in \mathbb{N}}$ is uniform, i.e., there is a recursive function $f \colon 1^n \to \phi_n$.

Theorem 2. *Let* (Q, κ) *be a parameterized problem such that instances* $x \in \Sigma^*$ *can be expressed via relational structures* $\mathcal{A}_x$ *over an* infinite *signature* τ*. For* $i \in \mathbb{N}$ *let* $(Q|_i, \kappa_2)$ *be a restriction of* (Q, κ) *such that for all* $x \in \Sigma^*$ *we have* $\kappa(x) \geq \kappa_1(x), \kappa_2(x)$ *and* $\kappa_2(x) \geq tw(\mathcal{A}'_x)$ *for a relational structure* $\mathcal{A}'_x$ *expressing* x *over a* finite *vocabulary* $\tau|_i$*, and*

1. $\bigcup_{i \in \mathbb{N}} Q|_i = Q$,
2. $\bigcup_{i \in \mathbb{N}} \tau|_i = \tau$, *and*
3. *for all* $x \in Q|_i$ *it holds* $i \leq \kappa_1(x)$.

If there exists a uniform family $(\phi_n)_{n \in \mathbb{N}}$ *of MSO-formulas over* τ *and a function* $g \colon \Sigma^* \to \{1\}^*$ *computable in* **FPT** *w.r.t.* κ *such that for all* $x \in \Sigma^*$ *it holds* $x \in Q|_n \Leftrightarrow \mathcal{A}'_x \models \phi_n \Leftrightarrow |g(x)| = n$ *then* $(Q, \kappa) \in \mathbf{FPT}$.

Proof. Let (Q, κ), $(\phi_n)_{n \in \mathbb{N}}$, κ_1, and κ_2 be given as in the conditions of the theorem. For every $x \in \Sigma^*$ there exists an $i \in \mathbb{N}$ such that there is a structure $\mathcal{A}'_x$ over the finite vocabulary $\tau|_i$ and it holds $x \in Q|_i$ if and only if $\mathcal{A}'_x \models \phi_i$. We compute $i := |g(x)|$ in **FPT**. Since $(\phi_n)_{n\in\mathbb{N}}$ is uniform we can construct ϕ_i in time $h(i)$ for an h that is w.l.o.g. recursive and non-decreasing. Because it holds that $h(i) \leq h(\kappa_1(x)) \leq h(\kappa(x))$, the construction is **FPT** w.r.t. κ.

Now we are able to solve the model checking problem instance $(\mathcal{A}'_x, \phi')$ in time $f'\big(\text{tw}(\mathcal{A}'_x), |\phi'|\big) \cdot |\mathcal{A}'_x|$ for a recursive f' due to Courcelle's theorem. W.l.o.g. assume that the structure $\mathcal{A}'_x$ is encoded in polynomial size and f' is non-decreasing. Then

$$f'\big(\text{tw}(\mathcal{A}'_x), |\phi'|\big) \cdot |\mathcal{A}'_x| \leq f'\big(\kappa_2(x), h(\kappa(x))\big) \cdot |x|^{O(1)} \leq f'\big(\kappa(x), h(\kappa(x))\big) \cdot |x|^{O(1)}.$$

As $x \in Q|_n \Leftrightarrow \mathcal{A}'_x \models \phi_n \Leftrightarrow g(x) = n$,this algorithm decides Q in **FPT**. □

Praveen [9] shows the fixed-parameter tractability of ML-SAT (parameterized by pathwidth and modal depth) by applying Courcelle's theorem, using for each modal formula an MSO-formula whose length is linear in the modal depth. This can be seen as a special case of Theorem 2 using a **P**-uniform MSO family that partitions the instance set according to the modal depth.

Again we want to stress that formula pathwidth of φ refers to the pathwidth of the corresponding structures $\mathcal{A}_\varphi$ as defined above.

Lemma 3. *Let $\varphi \in \mathcal{CTL}_{\text{NNF}}(\{\mathsf{AX}, \mathsf{EX}\}, B)$ given by the structure $\mathcal{A}_\varphi$ over τ. Then there exists an MSO formula θ such that $\varphi \in$ CTL-SAT($\{\mathsf{AX}\}$) iff $\mathcal{A}_\varphi \models \theta$.*

Proof. The first step is to show that a formula $\varphi \in \mathcal{CTL}_{\text{NNF}}(\{\mathsf{AX}, \mathsf{EX}\})$ is satisfiable if and only if it is satisfied by a Kripke structure of depth $\text{td}(\varphi)$, where the depth of a structure (M, w_0) is the maximal distance in M from w_0 to another state from M. This can be similar proven as the *tree model property* of modal logic [15, p. 269, Lemma 35].

Let φ be the given formula in $\mathcal{CTL}_{\text{NNF}}(\{\mathsf{AX}, \mathsf{EX}\})$. The following formula θ_{struc} describes the properties of the structure $\mathcal{A}_\varphi$. At first it takes care of the uniqueness of the formula representative. If an element x does not represent a formula then it has to be a subformula. Additionally if x it is not a variable it has to be either a constant, or a Boolean function $f \in B$ with the corresponding arity $\text{ar}(f)$, or an AX-, or an EX-formula respectively. Furthermore the distinctness of the representatives has to be ensured which together with the previous constraints implies acyclicity.

In the following $f_1(u, v, w, x)$ corresponds to the operator of the function which is true if exactly one of its arguments is true.

$$
\begin{aligned}
\theta_{\text{struc}} :=& \forall x \forall y (\text{repr}(x) \land \text{repr}(y) \to x = y) \land \\
& \forall x \Big(\neg\text{repr}(x) \to \exists y \big(\neg\text{var}(y) \land \bigvee_{\substack{f \in \{\land, \lor, \neg\}, \\ 1 \le i \le \text{ar}(f)}} \text{conn}_{f,i}(x, y) \big) \Big) \land \\
& \forall x\, f_1 \Big(\text{var}(x), \bigvee_{f \in \{\top, \bot\}} \text{const}_f(x), \\
& \qquad \bigvee_{\substack{f \in B, \\ \text{ar}(f) \ge 1}} \bigwedge_{1 \le i \le \text{ar}(f)} \exists y \big(\text{conn}_{f,i}(y, x) \land \forall z \big(\text{conn}_{f,i}(z, x) \to z = y \big) \big) \\
& \qquad \exists y \big(\text{body}_{\mathsf{AX}}(y, x) \land \forall z \big(\text{body}_{\mathsf{AX}}(z, x) \to z = y \big) \big) \\
& \qquad \exists y \big(\text{body}_{\mathsf{EX}}(y, x) \land \forall z \big(\text{body}_{\mathsf{EX}}(z, x) \to z = y \big) \big) \Big) \land \\
& \forall x \forall y \big((\text{body}_{\mathsf{AX}}(y, x) \to \text{repr}_{\mathsf{AX}}(x)) \land (\text{body}_{\mathsf{EX}}(y, x) \to \text{repr}_{\mathsf{EX}}(x)) \big).
\end{aligned}
$$

The previous formula is a modification of the formula used in the proof of Lemma 1 in [10].

The next formulas will quantify sets M_i which represent sets of satisfied subformulas at worlds in the Kripke structure at depth i. Here the formulas with propositional connectives, resp., all constants, have a valid assignment obeying their function value in the model M_i. The AX- and EX-formulas are processed as expected: the EX-formulas branch to different worlds and the AX-formulas have to hold in all possible next worlds. Now we are ready to define θ^i_{assign} in an inductive way. At depth 0 we want to consider only propositional formulas. Here it ensures that all Boolean functions obey the model.

$$
\begin{aligned}
\theta^0_{\text{assign}}(M_0) :=& \forall x, y_1, \ldots, y_n \in M_0 : \text{repr}_{\mathsf{PL}}(x) \land \\
& \bigwedge_{f \in B} \Big(\bigwedge_{\text{ar}(f)=0} \text{const}_f(x) \to f \land \bigwedge_{1 \le i \le \text{ar}(f)} \text{conn}_{f,i}(y_i, x) \to f(M_0(y_1), \ldots, M_0(y_{\text{ar}(f)})) \Big)
\end{aligned}
$$

In the general definition of θ^i_{assign} we utilize for convenience two subformulas, $\theta^i_{\text{branchEX}}$ and θ^i_{stepAX}. The first is defined for an element x representing an EX-formula, a set of elements M_i representing to be satisfied formulas, and a set of elements M_{AX} representing the AX-formulas which are satisfied in the current world. The formula enforces that the formula $\mathsf{EX}\psi$ represented by x has to hold in the next world together with all bodies of the AX-formulas.

$$
\begin{aligned}
\theta^i_{\text{branchEX}}(M_i, M_{\mathsf{AX}}, x) := \exists y \Big(& \text{body}_{\mathsf{EX}}(y, x) \land \\
& \exists M_{i-1} \big(M_{i-1}(y) \land \forall z \in M_{\mathsf{AX}} (\exists w\, \text{body}_{\mathsf{AX}}(w, z) \land M_{i-1}(w)) \land \\
& \qquad \theta^{i-1}_{\text{assign}}(M_{i-1}) \big) \Big),
\end{aligned}
$$

The second formula is crucial when there are no EX-formulas represented in M_i. Then the AX-formulas still have to be satisfied eventually wherefore we proceed with a single next world (without any branching required).

$$\theta^i_{\mathrm{stepAX}}(M_{\mathsf{AX}}) := \exists M_{i-1} \forall z \in M_{\mathsf{AX}}(\exists w \, \mathrm{body}_{\mathsf{AX}}(w,z) \wedge M_{i-1}(w)) \wedge \theta^{i-1}_{\mathrm{assign}}(M_{i-1})$$

Now we turn towards the complete inductive definition step where we need to differentiate between the two possible cases for representatives: either a propositional or a temporal formula is represented. The first part is similar to the induction start and the latter follows the observation that for every EX-preceded formula we want to branch. In each such branch all not yet satisfied AX-preceded formulas have to hold. The set M_{AX} contains all AX-formulas which are satisfied in the current world. If we do not have any EX-formulas then we enforce a single next world for the remaining AX-formulas.

$$\begin{aligned}
\theta^i_{\mathrm{assign}}(M_i) :=& \forall x, y_1, \ldots, y_n \in M_i \\
& \bigwedge_{f \in B} \Bigg(\bigwedge_{\mathrm{ar}(f)=0} \mathrm{const}_f(x) \to (M_i(x) \leftrightarrow f) \wedge \\
& \qquad \bigwedge_{1 \le i \le \mathrm{ar}(f)} \mathrm{conn}_{f,i}(y_i, x) \to \big(M_i(x) \leftrightarrow f(M_i(y_1), \ldots, M_i(y_{\mathrm{ar}(f)}))\big) \Bigg) \wedge \\
& \exists M_{\mathsf{AX}} \subseteq M_i \Big(\forall x \Big(M_{\mathsf{AX}}(x) \leftrightarrow \big(\mathrm{repr}_{\mathsf{AX}}(x) \wedge M_i(x)\big)\Big) \wedge \\
& \qquad \forall x \in M_i \big(\mathrm{repr}_{\mathsf{EX}}(x) \to \theta^i_{\mathrm{branchEX}}(M_i, M_{\mathsf{AX}}, x)\big) \wedge \\
& \qquad (\forall x \in M_i(\neg \mathrm{repr}_{\mathsf{EX}}(x))) \to \theta^i_{\mathrm{stepAX}}(M_{\mathsf{AF}}) \Big)
\end{aligned}$$

Through the construction we get that φ is satisfiable iff $\mathcal{A}_\varphi \models \theta_{\mathrm{struc}} \wedge \exists M(\theta^{\mathrm{td}(\varphi)}_{\mathrm{assign}}(M))$. □

Corollary 4. CTL-SAT($\{\mathsf{AX}\}$) *parameterized by formula pathwidth and temporal depth is fixed-parameter tractable.*

Proof. For every fixed $i \in \mathbb{N}$ transform the formula into NNF and apply Lemma 3. As pathwidth is an upper bound for treewidth, we apply Theorem 2 in the following way. The parameter $\kappa_1 = i$ is the temporal depth and κ_2 is the formula pathwidth. It holds that $\kappa = \kappa_1 + \kappa_2$. The set $Q|_i$ restricts CTL-SAT($\{\mathsf{AX}\}$) to formulas φ for which $\mathrm{td}_i(\varphi) \wedge \neg \mathrm{td}_{i+1}(\varphi)$ holds. Using the computable family of MSO-formulas from Lemma 3 we already get the desired result (as the vocabulary itself is finite). □

Intractable fragments of CTL-SAT

Lemma 5. CTL-SAT(T) *parameterized by formula pathwidth and temporal depth is* $\mathbf{W}[1]$*-hard if* $\{\mathsf{AX}, \mathsf{AF}\} \subseteq T$.

Proof. We will modify the construction in the proof of Praveen [9, Lemma A.3] and thereby state an fpt-reduction from the parameterized problem p-PW-SAT whose input is $(\mathcal{F},\ part : \Phi \to [k],\ tg : [k] \to \mathbb{N})$, where $\mathcal{F}$ is a propositional CNF formula, *part* is a function that partitions the set of propositional variables of $\mathcal{F}$ into k parts, and tg is a function which maps to each part a natural number. The task is to find a satisfying assignment of $\mathcal{F}$ such that in each part $p \in [k]$ exactly $tg(p)$ variables are set to true. A generalization of this problem to arbitrary formulas $\mathcal{F}$ (i.e., the CNF constraint is dropped) is $\mathbf{W}[1]$-hard when parameterized by k and the pathwidth of the structural representation $\mathcal{A}_{\mathcal{F}}$ of $\mathcal{F}$ which is similar proven as in [9, Lemma 7.1].

The further idea is to construct a $\mathcal{CTL}$-formula $\phi_{\mathcal{F}}$ in which we are able to verify the required targets. The formula enforces a Kripke structure $K = (W, R, V)$ where in each world $w \in W$ the value of $V(w)$ coincides with a satisfying assignment f of $\mathcal{F}$ together with the required targets. Each such K contains as a substructure a chain $w_0 R w_1 R \cdots R w_n$ of worlds and all variables q_i in $\mathcal{F}$ are labeled to each w_j if $f(q_i)$ holds.

Let $q_1, \dots, q_n$ be all the propositional variables in $\mathcal{F}$. Then $t_{\uparrow 1}, \dots, t_{\uparrow k}$, respectively, $f_{\uparrow 1}, \dots, f_{\uparrow k}$ are propositions to distinguish the parts, $tr_p^0, \dots, tr_p^{n[p]}$, respectively, $fl_p^0, \dots, fl_p^{n[p]}$ for $p \in [k]$, are counter propositions for the number of variables set to true and false, $d_0, \dots, d_{n+1}$ are depth propositions, and $\Phi(p)$ denotes the set of variables in part $p \in [k]$.

The formula $\phi_{\mathcal{F}}$ that is the conjunction of subformulas (Figure 2) similar to [9, Lemma A.3] states the reduction from p-PW-SAT to CTL-SAT($\{\mathsf{AX}, \mathsf{AG}\}$) parameterized by temporal depth and pathwidth. With respect to Praveens approach we explain how to obtain a formula consisting of only one single AG operator leading to a formula $\phi_{\mathcal{F}} = \psi \land \mathsf{AG}\chi$, where ψ is purely propositional and $\chi \in \mathcal{CTL}(\{\mathsf{AX}\})$. Then AG can be replaced by EG and the proof stays valid since there is only one instance of an existential temporal operator and it occurs at temporal depth zero. As $\mathsf{AG}(\alpha) \land \mathsf{AG}(\beta) \equiv \mathsf{AG}(\alpha \land \beta)$ we can modify the formula $\phi_{\mathcal{F}}$ which is a conjunction of the formulas from above to the desired form containing only a single AG. This is then replaced by EG and the argumentation follows below.

In the following we assume the chain of worlds as explained before to be the relevant part of the model. The world where the conjunction $\phi_{\mathcal{F}}$ holds is assumed to be w_0. The formula *determined* forces the variables q_i not to change their value in successor levels by passing the value of each q_i to all next levels. Hence we get $\mathcal{M}, w_0 \models \mathcal{F} \land \mathit{determined}$. *depth* ensures that in the world w_i holds $d_i \land \neg d_{i+1}$, $\mathcal{M}, w_0 \models \mathit{determined} \land (d_0 \land \neg d_1)$ by *countInt*. In the next formula *setCounter* the variable $t_{\uparrow part(i)}$ holds if q_i is set to true at the world w_{i-1}, respectively, the variable $f_{\uparrow part(i)}$ if q_i does not hold at w_{i-1}.

Now we use the variables $t_{\uparrow part(i)}$ to increment the counter propositions $tr_p^0, \dots, tr_p^{n[p]}$ for all variables set to true in the formula *incCounter* as follows. If the j variables $\Phi(p) \cap \{q_1, \dots q_j\}$ at $part(p)$ are set to true so is $t_{\uparrow p}$ set at the world w_{i-1} to true and all successors of w_{i-1} force increment of the value ℓ in tr_p^{ℓ}, respectively, fl_p^{ℓ}. This is ensured step wisely depending on the temporal depth n.

$$
\begin{aligned}
\mathit{determined} :=& \mathsf{AG} \bigwedge_{i=1}^{n} \Big((q_i \Rightarrow \mathsf{AX} q_i) \wedge (\neg q_i \Rightarrow \mathsf{AX} \neg q_i) \Big) \\
\mathit{depth} :=& \bigwedge_{i=1}^{n-2} \Big((d_i \wedge \neg d_{i+1}) \Rightarrow \mathsf{AX}(d_{i+1} \wedge \neg d_{i+2}) \Big) \\
\mathit{setCounter} :=& (q_1 \Rightarrow t_{\uparrow part(1)}) \wedge (\neg q_1 \Rightarrow f_{\uparrow part(1)}) \wedge \\
& \mathsf{AG} \bigwedge_{i=2}^{n} \Big((d_{i-1} \wedge \neg d_i) \Rightarrow \big[(q_i \Rightarrow t_{\uparrow part(i)}) \wedge (\neg q_i \Rightarrow f_{\uparrow part(i)}) \big] \Big) \\
\mathit{incCounter} :=& \Big((t_{\uparrow part(1)} \Rightarrow \mathsf{AX} tr^1_{part(1)}) \wedge (f_{\uparrow part(1)} \Rightarrow \mathsf{AX} fl^1_{part(1)}) \Big) \wedge \\
& \mathsf{AG} \bigwedge_{p=1}^{k} \bigwedge_{j=0}^{n[p]-1} \Big[\Big(t_{\uparrow p} \Rightarrow \Big(tr^j_p \Rightarrow tr^{j+1}_p \wedge \mathsf{AX} tr^{j+1}_p \Big) \Big) \wedge \Big(f_{\uparrow p} \Rightarrow \Big(fl^j_p \Rightarrow fl^{j+1}_p \wedge \mathsf{AX} fl^{j+1}_p \Big) \Big) \Big] \\
\mathit{targetMet} :=& \mathsf{AG} \bigwedge_{p=1}^{k} (d_n \Rightarrow tr^{tg(p)}_p \wedge \neg tr^{tg(p)+1}_p \wedge fl^{n[p]-tg(p)}_p \wedge \neg tr^{n[p]-tg(p)+1}_p) \\
\mathit{determined'} :=& \mathsf{AG} \bigwedge_{p=1}^{k} \Big((tr^0_p \Rightarrow tr^0_p) \wedge (fl^0_p \Rightarrow fl^0_p) \Big) \\
\mathit{countInt} :=& d_0 \wedge \neg d_1 \wedge \bigwedge_{p=1}^{k} (\neg tr^1_p \wedge \neg fl^1_p \wedge tr^0_p \wedge fl^0_p) \\
\mathit{depth'} :=& \mathsf{AG} \bigwedge_{p=1}^{k} \bigwedge_{j=0}^{n[p]} \Big[\Big(tr^j_p \Rightarrow tr^j_p \Big) \wedge \Big(fl^j_p \Rightarrow (fl^j_p) \Big) \Big] \\
\mathit{countMonotone} :=& \mathsf{AG} \left(\bigwedge_{i=1}^{n} \Big((d_i \Rightarrow d_{i-1}) \Big) \wedge \bigwedge_{p=1}^{k} \bigwedge_{l=2}^{n[p]} \Big[(tr^j_p \Rightarrow tr^{j-1}_p) \wedge (fl^j_p \Rightarrow fl^{j-1}_p) \Big] \right)
\end{aligned}
$$

Fig. 2. Reduction from p-PW-SAT to CTL-SAT($\{\mathsf{AX}, \mathsf{AG}\}$)

The counters for the target function tr^i_p and fl^i_p are initialized by *countInt*, i.e., tr^0_p and fl^0_p are set to true in all w_is and tr^1_p and fl^1_p are set to false in w_0 by *determined'* and *countInt*. Additionally the formula *depth'* defines the scope of the counters tr^i_p and fl^i_p which depends on the temporal depth and the property of being monotonically nondecreasing is defined by *countMonotone*. The given target function $tg : [k] \to \mathbb{N}$ is then checked with the formula *targetMet* such that $\mathcal{M}, w_0 \models \mathit{targetMet}$, i.e., in the world at depth k the target proposition $tr^{tg(p)}_p$ must hold (and must stop, i.e., $tr^{tg(p)+1}_p$ is false) for each part $p \in [k]$. The correctness of the reduction is similarly proven as in [9, Lemma A.3]. □

Lemma 6. CTL-SAT(T) *parameterized by formula pathwidth and temporal depth is* $\mathbf{W}[1]$*-hard if* $\mathsf{AG} \in T$.

Proof. Now we consider the case were $T = \{\mathsf{AG}\}$. As $\mathsf{AG}\varphi$ is equivalent to $\neg\mathsf{EF}\neg\varphi$ we can simple substitute in the constructed formula $\phi_{\mathcal{F}}$ from [9, Lemma A.3] the occurrence of EX with EF. By this the possible "steps" invoked by the EX-operator

$$
\begin{aligned}
\mathit{determined} :=& \bigwedge_{i=1}^{n} \Big(q_i \Rightarrow \mathsf{A}[q_i \mathsf{U} d_{n+1}]\Big) \wedge \bigwedge_{i=1}^{n} \Big(\neg q_i \Rightarrow \mathsf{A}[\neg q_i \mathsf{U} d_{n+1}]\Big) \\
\mathit{depth} :=& \bigwedge_{i=1}^{n} \Big(\mathsf{A}\,[d_i \wedge \neg d_{i+1} \mathsf{U} d_{i+1} \wedge \neg d_{i+2}]\Big) \\
\mathit{setCounter} :=& (q_1 \Rightarrow t_{\uparrow part(1)}) \wedge (\neg q_1 \Rightarrow f_{\uparrow part(1)}) \wedge \\
& \bigwedge_{i=2}^{n} \mathsf{A}[(d_{i-1} \wedge \neg d_i) \Rightarrow \big((q_i \Rightarrow t_{\uparrow part(i)}) \wedge (\neg q_i \Rightarrow f_{\uparrow part(i)})\big)] \mathsf{U} d_{n+1}] \\
\mathit{incCounter} :=& (t_{\uparrow part(1)} \Rightarrow \mathsf{A}[tr^1_{part(1)} \mathsf{U} d_{n+1}]) \wedge (f_{\uparrow part(1)} \Rightarrow \mathsf{A}[fl^1_{part(1)} \mathsf{U} d_{n+1}]) \wedge \\
& \bigwedge_{p=1}^{k} \bigwedge_{j=0}^{n[p]-1} \Big[\Big(\mathsf{A}[t_{\uparrow p} \mathsf{U} d_{n+1}] \Rightarrow \Big(tr^j_p \Rightarrow \mathsf{A}[tr^{j+1}_p \mathsf{U} d_{n+1}]\Big)\Big) \wedge \\
& \qquad \Big(\mathsf{A}[f_{\uparrow p} \mathsf{U} d_{n+1}] \Rightarrow \Big(fl^j_p \Rightarrow \mathsf{A}[fl^{j+1}_p \mathsf{U} d_{n+1}]\Big)\Big)\Big] \\
\mathit{targetMet} :=& \bigwedge_{p=1}^{k} \mathsf{A}\Big[\big(d_n \Rightarrow (tr^{tg(p)}_p \wedge \neg tr^{tg(p)+1}_p \wedge fl^{n[p]-tg(p)}_p \wedge \neg fl^{n[p]-tg(p)+1}_p) \mathsf{U} d_{n+1}\Big] \\
\mathit{determined'} :=& \bigwedge_{p=1}^{k} \Big(\big((tr^0_p \Rightarrow \mathsf{A}[tr^0_p \mathsf{U} d_{n+1}])\big) \wedge (fl^0_p \Rightarrow \mathsf{A}[fl^0_p \mathsf{U} d_{n+1}])\Big) \\
\mathit{countInt} :=& d_0 \wedge \neg d_1 \wedge \bigwedge_{p=1}^{k} (\neg tr^1_p \wedge \neg fl^1_p \wedge tr^0_p \wedge fl^0_p) \\
\mathit{depth'} :=& \bigwedge_{p=1}^{k} \bigwedge_{j=0}^{n[p]} \Big(\mathsf{A}[(tr^j_p \Rightarrow tr^j_p) \wedge (fl^j_p \Rightarrow fl^j_p) \mathsf{U} d_{n+1}]\Big) \\
\mathit{countMonotone} :=& \bigwedge_{i=1}^{n} \mathsf{A}\left[(d_i \Rightarrow d_{i-1}) \wedge \bigwedge_{p=1}^{k} \bigwedge_{l=2}^{n[p]} (tr^j_p \Rightarrow tr^{j-1}_p) \wedge (fl^j_p \Rightarrow fl^{j-1}_p) \mathsf{U} d_{n+1}\right]
\end{aligned}
$$

Fig. 3. Reduction from p-PW-SAT to CTL-SAT($\{\mathsf{AU}\}$)

become "jumps" through EF. The proof in Lemma 5 and [9, Lemma A.3] allows this modification without any change in the argumentation. ☐

Lemma 7. CTL-SAT(T) *parameterized by formula pathwidth and temporal depth is* $\mathbf{W}[1]$*-hard if* $\mathsf{AU} \in T$.

Proof. We modify the reduction proven in [9, Lemma A.3] to simulate the AG-formulas with the help of AU-formulas as shown in Figure 3. The idea is to introduce another depth proposition after d_n, namely d_{n+1}. This is used to express $\mathsf{AG}\phi$ by $\mathsf{A}[\phi \mathsf{U} d_{n+1}]$. ☐

4 Conclusion

In this work we presented an almost complete classification with respect to parameterized complexity of all possible CTL-operator fragments of the satisfiability problem in computation tree logic CTL parameterized by formula pathwidth and temporal depth. Only the case for the fragment containing solely AF remains open. Currently we are working on a classification which aims for an **FPT** result and uses the "full version" of Theorem 2; the main goal is to prove for unsatisfiable formulas a contradiction is found within $f(\mathrm{td})$ steps, where td is the temporal depth and f is some computable function. Then we construct a family of MSO formulas for each such temporal depth and use Theorem 2. The classified results form a dichotomy with two fragments in **FPT** and the remainder being **W**[1]-hard.

Comparing our results to the situation in usual computational complexity for the decision case they do not behave as expected. Surprisingly the fragment $\{\mathsf{AX}\}$ is **FPT** whereas on the decision side this fragment is **PSPACE**-complete. For the other classified fragments the rule of thumb is the following: The **NP**-complete fragments are **FPT** whereas the **PSPACE**- and **EXPTIME**-complete fragments are **W**[1]-hard. For the shown **W**[1]-hardness results an exact classification with matching upper bounds is open for further research. Similarly a complete classification with respect to all possible Boolean fragments in the sense of Post's lattice is one of our next steps.

Furthermore we constructed a generalization of Courcelle's theorem to infinite signatures for parameterization compositions containing at least a parameter which bounds the treewidth of relational structures expressing the inputs under the existence of a computable family of MSO-formulas (cf. Theorem 2). Previously such a general result for infinite signatures was not known to the best of the authors knowledge and is of independent interest.

Another consequent step will be the classification of other temporal logics fragments, e.g., of linear temporal logic LTL and the full branching time logic CTL* with respect to their parameterized complexity. Also the investigation of other parameterizations beyond the usual considered measures of pathwidth or treewidth and temporal depth may lead to a better understanding of intractability in the parameterized sense.

References

1. Prior, A.N.: Time and Modality. Clarendon Press, Oxford (1957)
2. Kripke, S.: Semantical considerations on modal logic. In: Acta Philosophica Fennica. Volume 16. (1963) 84–94
3. Pnueli, A.: The temporal logic of programs. In: Proc. 18th Symposium on Foundations of Computer Science, IEEE Computer Society Press (1977) 46–57
4. Emerson, E.A., Halpern, J.Y.: Decision procedures and expressiveness in the temporal logic of branching time. Journal of Computer and System Sciences **30**(1) (1985) 1–24
5. Clarke, E.M., Emerson, E.A.: Desing and synthesis of synchronisation skeletons using branching time temporal logic. In: Logic of Programs. Volume 131 of Lecture Notes in Computer Science., Springer Verlag (1981) 52–71
6. Meier, A.: On the Complexity of Modal Logic Variants and their Fragments. PhD thesis, Gottfried Wilhelm Leibniz Universität Hannover (2011)
7. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer-Verlag (1999) 530 pp.

8. Courcelle, B., Engelfriet, J.: Graph structure and monadic second-order logic, a language theoretic approach. Cambridge University Press (2012)
9. Praveen, M.: Does treewidth help in modal satisfiability? ACM Transactions on Computational Logic **14**(3) (2013) 18:1–18:32
10. Meier, A., Schmidt, J., Thomas, M., Vollmer, H.: On the Parameterized Complexity of Default Logic and Autoepistemic Logic. In: 6th International Conference on Language and Automata Theory and Applications. Volume 7183 of LNCS. (2012) 389–400
11. Gottlob, G., Pichler, R., Wei, F.: Bounded treewidth as a key to tractability of knowledge representation and reasoning. Artificial Intelligence **174**(1) (2010) 105–132
12. Elberfeld, M., Jakoby, A., Tantau, T.: Logspace versions of the theorems of Bodlaender and Courcelle. In: Proc. 51th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society (2010)
13. Pippenger, N.: Theories of Computability. Cambridge University Press, Cambridge (1997)
14. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer Verlag (2006)
15. Blackburn, P., de Rijke, M., Venema, Y.: Modal logic. Cambridge University Press, New York, NY, USA (2001)

ECCC ISSN 1433-8092
http://eccc.hpi-web.de