Technical Report AC-TR-15-009

December 2015

Machine Characterizations for Parameterized Complexity Classes Beyond Para-NP

Ronald de Haan and Stefan Szeider

This is the authors' copy of a paper that appeared in the proceedings of SOFSEM 2015, pp. 217–229, LNCS 8939, Springer, 2015.

www.ac.tuwien.ac.at/tr

Machine Characterizations for Parameterized Complexity Classes beyond para-NP

Ronald de Haan⋆ and Stefan Szeider⋆

Institute of Information Systems, Vienna University of Technology, Vienna, Austria

Abstract. Due to the remarkable power of modern SAT solvers, one can efficiently solve NP-complete problems in many practical settings by encoding them into SAT. However, many important problems in various areas of computer science lie beyond NP, and thus we cannot hope for polynomial-time encodings into SAT. Recent research proposed the use of fixed-parameter tractable (fpt) reductions to provide efficient SAT encodings for these harder problems. The parameterized complexity classes $\exists^k\forall^*$ and $\forall^k\exists^*$ provide strong theoretical evidence that certain parameterized problems are not fpt-reducible to SAT. Originally, these complexity classes were defined via weighted satisfiability problems for quantified Boolean formulas, extending the general idea for the canonical problems for the Weft Hierarchy.

In this paper, we provide alternative characterizations of $\exists^k\forall^*$ and $\forall^k\exists^*$ in terms of first-order logic model checking problems and problems involving alternating Turing machines with appropriate time bounds and bounds on the number of alternations. We also identify parameterized Halting Problems for alternating Turing machines that are complete for these classes.

The alternative characterizations provide evidence for the robustness of the new complexity classes and extend the toolbox for establishing membership results. As an illustration, we consider various parameterizations of the 3-coloring extension problem.

1 Introduction

The recent success of modern SAT solvers in many practical settings has placed them at the heart of an important approach to solving NP-complete problems, where problem instances are encoded to SAT and subsequently solved using a SAT solver [3,12,17,22]. However, many important computational problems lie above the first level of the Polynomial Hierarchy (PH), and thus this approach does not work to solve these problems, as polynomial-time reductions to SAT are not possible for these problems, unless the PH collapses.

Problem instances occurring in practical settings are not random, and often contain some kind of structure, which can be exploited by parameterized algorithms. Recently, the structure in problems instances was used to break the complexity barriers between the first and second level of the PH, by means of fpt-reductions [9,21]. Such fpt-reducibility results adopt a new perspective on what amounts to positive results in

⋆ Supported by the European Research Council (ERC), project 239962 (COMPLEX REASON), and the Austrian Science Fund (FWF), project P26200 (Parameterized Compilation).

parameterized complexity. This new perspective (i.e., aiming at fpt-reducibility to SAT rather than fpt-solvability) greatly extends the power of positive results, as parameters can be less restrictive, and problems can be solved efficiently on larger classes of instances.

In order to provide suitable negative results, a new parameterized complexity classes $\exists^k\forall^*$ has been introduced [13,14], which lies at the basis of a hardness theory that provides such negative evidence. The class $\exists^k\forall^*$ is located above para-co-NP and below para-Σ_2^{P} (see Figure 1), and is based on weighted variants of quantified Boolean satisfiability problems. Several problems from various domains have already been shown hard or complete for the class $\exists^k\forall^*$ or its dual $\forall^k\exists^*$, including problems in Knowledge Representation [14], Boolean Optimization [13], and Computational Social Choice [8].

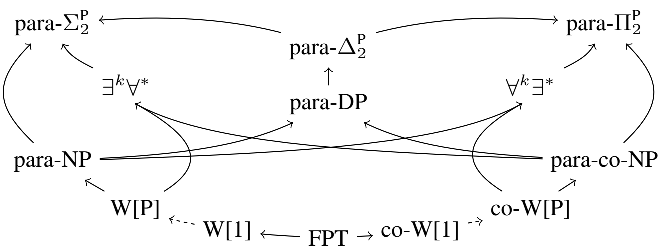

Fig. 1: Parameterized complexity classes up to the second level of the polynomial hierarchy. Arrows indicate inclusion relations. (For a definition of the classes para-DP and para-Δ_2^{P}, we refer to other resources [2,10,20].)

The role that $\exists^k\forall^*$ and $\forall^k\exists^*$ play in the analysis of parameterizations of problems complete for the second level of the PH, is analogous to the role that the Weft-hierarchy plays in the analysis of parameterizations of NP-complete problems. The parameterized complexity classes para-NP and para-co-NP, on the one hand, and the classes $\exists^k\forall^*$ and $\forall^k\exists^*$, on the other hand, constitute a borderline between problems that are fpt-reducible to SAT (or UNSAT) and problems that are not, similarly to the way in which the classes W[1] and FPT provide a borderline between problems that are fixed-parameter tractable and problems that are not. Neither W[1] nor $\exists^k\forall^*$ and $\forall^k\exists^*$ have a direct counterpart in classical complexity theory, and these classes thus provide a tighter complexity analysis than parameterized complexity classes that are derived from classical classes [10].

1.1 New Contributions

We provide new characterizations of the parameterized complexity class $\exists^k\forall^*$ in terms of first-order model checking problems and in terms of alternating Turing machines, with appropriate time bounds. Consequently, dual characterizations hold for the parameterized complexity class $\forall^k\exists^*$. More specifically, we show the following results.

1. $\exists^k\forall^*$ is precisely the class of all parameterized problems that can be expressed in terms of checking whether a given first-order formula $\exists x_1,\dots,x_k.\forall y_1,\dots,y_n.\psi$ is true in a given relational structure, taking the number of existential variables as the parameter. (Theorem 1)

2. $\exists^k\forall^*$ is precisely the class of all parameterized problems that can be decided by a 2-alternating Turing machine that runs in fixed-parameter tractable time, starts in an existential state, and uses a number of nondeterministic existential steps that is bounded by a function of the parameter. (Theorem 2)
3. The Halting Problem for 2-alternating Turing machines that start in an existential state, parameterized by the number of nondeterministic existential steps, is complete for $\exists^k\forall^*$. (Theorem 3)

Theorem 1 provides an easy and convenient way for establishing membership results; we use it also in the proofs of Theorems 2 and 3 and give an example application to a combinatorial problem in Section 5.

Theorem 2 establishes the robustness of the class $\exists^k\forall^*$, in analogy to the characterization of the first two classes of the Weft-hierarchy in terms of Turing machines [4,5].

Theorem 3 provides an analogue to the Cook-Levin Theorem for the complexity class $\exists^k\forall^*$, which supports our assumption that $\exists^k\forall^* \neq$ para-co-NP, in analogy to the argumentation that the W[1]-completeness of the Halting Problem for nondeterministic Turing machines, parameterized by the number of steps, supports the assumption W[1] $\neq$ FPT (see [6] and cf. the discussion in [5]). Interestingly, our version of the Halting Problem remains $\exists^k\forall^*$-complete, independently of whether the Turing machine uses a single tape, or an arbitrary number of tapes, in contrast to versions of the Halting Problem that characterize classes of the Weft-hierarchy, where a single tape captures W[1], and an arbitrary number of tapes captures W[2] [4,5].

We would like to remark that the membership in W[1] or W[2] for some parameterized problems remained open for a long time, and was finally established by means of machine characterizations [4]. We expect that our machine characterizations for $\exists^k\forall^*$ can be of similar use.

In Section 5 we exemplify our new complexity toolbox by applying it to parameterizations of a graph coloring problem, shown to be Π_2^{P}-complete by Ajtai, Fagin, and Stockmeyer [1].

We provide proof sketches for the results presented in this paper. For full detailed proofs we refer to a technical report [15].

2 Preliminaries

Propositional and First-Order Logic A *literal* is a propositional variable x or a negated variable $\neg x$. We use the standard notion of *(truth) assignments* $\alpha : \mathrm{Var}(\varphi) \rightarrow \{0,1\}$ for Boolean formulas and *truth* of a formula under such an assignment.

A *(relational) vocabulary* τ is a finite set of relation symbols. Each relation symbol R has an *arity* $\mathrm{arity}(R) \geq 1$. A *structure* $\mathcal{A}$ of vocabulary τ, or τ*-structure* (or simply *structure*), consists of a set A called the *domain* and an interpretation $R^{\mathcal{A}} \subseteq A^{\mathrm{arity}(R)}$ for each relation symbol $R \in \tau$. We use the usual definition of truth of a first-order logic sentence φ over the vocubulary τ in a τ-structure $\mathcal{A}$. We let $\mathcal{A} \models \varphi$ denote that the sentence φ is true in structure $\mathcal{A}$. If φ is a first-order formula with free variables $\mathrm{Free}(\varphi)$, and $\mu : \mathrm{Free}(\varphi) \rightarrow A$ is an assignment, we use the notation $\mathcal{A}, \mu \models \varphi$ to denote that φ is true in structure $\mathcal{A}$ under the assignment μ.

The Polynomial Hierarchy There are many natural decision problems that are not contained in the classical complexity classes P or NP. The *Polynomial Hierarchy* [18,20,23,24] contains a hierarchy of increasing complexity classes Σ^{P}_i, for all $i \geq 0$. We give a characterization of these classes based on the satisfiability problem of various classes of quantified Boolean formulas. A *quantified Boolean formula* is a formula of the form $Q_1 X_1 Q_2 X_2 \dots Q_m X_m \psi$, where each Q_i is either $\forall$ or $\exists$, the X_i are disjoint sets of propositional variables, and ψ is a Boolean formula over the variables in $\bigcup_{i=1}^{m} X_i$. The quantifier-free part of such formulas is called the *matrix* of the formula. Truth of such formulas is defined in the usual way. Let $\gamma = \{x_1 \mapsto d_1, \dots, x_n \mapsto d_n\}$ be a function that maps some variables of a formula φ to truth values. We let $\varphi[\gamma]$ denote the application of such a substitution γ to the formula φ. For each $i \geq 1$, we let QSAT_i be the problem to decide whether a given quantified Boolean formula $\varphi = \exists X_1 \forall X_2 \exists X_3 \dots Q_i X_i \psi$ is true, where Q_i is a universal quantifier if i is even and an existential quantifier if i is odd.

Input formulas to the problem QSAT_i are called Σ^{P}_i-formulas. For each nonnegative integer $i \leq 0$, the complexity class Σ^{P}_i can be characterized as the closure of the problem QSAT_i under polynomial-time reductions [23,24]. The Σ^{P}_i-hardness of QSAT_i holds already when the matrix of the input formula is restricted to 3CNF for odd i, and restricted to 3DNF for even i. The class Σ^{P}_0 coincides with P, and the class Σ^{P}_1 coincides with NP. For each $i \geq 1$, the class Π^{P}_i is defined as co-Σ^{P}_i.

Parameterized Complexity We briefly introduce some core notions from parameterized complexity theory. For an in-depth treatment we refer to other sources [6,7,11,19]. A *parameterized problem* L is a subset of $\Sigma^* \times \mathbb{N}$ for some finite alphabet Σ. For an instance $(I, k) \in \Sigma^* \times \mathbb{N}$, we call I the *main part* and k the *parameter*. A parameterized problem L is *fixed-parameter tractable* if there exists a computable function f and a constant c such that there exists an algorithm that decides whether $(I, k) \in L$ in time $O(f(k)\|I\|^c)$, where $\|I\|$ denotes the size of I. Let $L \subseteq \Sigma^* \times \mathbb{N}$ and $L' \subseteq (\Sigma')^* \times \mathbb{N}$ be two parameterized problems. An *fpt-reduction* from L to L' is a mapping $R : \Sigma^* \times \mathbb{N} \to (\Sigma')^* \times \mathbb{N}$ from instances of L to instances of L' such that there exist some computable function $g : \mathbb{N} \to \mathbb{N}$ such that for all $(I, k) \in \Sigma^* \times \mathbb{N}$: (i) (I, k) is a yes-instance of L if and only if $(I', k') = R(I, k)$ is a yes-instance of L', (ii) $k' \leq g(k)$, and (iii) R is computable in fpt-time. We write $L \leq_{\mathrm{fpt}} L'$ if there is an fpt-reduction from L to L'. Similarly, we call reductions that satisfy properties (i) and (ii) but that are computable in time $O(\|I\|^{f(k)})$, for some fixed computable function f, *xp-reductions*.

Parameterized complexity theory also offers complexity classes for problems that lie higher in the polynomial hierarchy. Let C be a classical complexity class, e.g., NP. The parameterized complexity class para-C is then defined as the class of all parameterized problems $L \subseteq \Sigma^* \times \mathbb{N}$, for some finite alphabet Σ, for which there exist an alphabet Π, a computable function $f : \mathbb{N} \to \Pi^*$, and a problem $P \subseteq \Sigma^* \times \Pi^*$ such that $P \in C$ and for all instances $(x, k) \in \Sigma^* \times \mathbb{N}$ of L we have that $(x, k) \in L$ if and only if $(x, f(k)) \in P$. Intuitively, the class para-C consists of all problems that are in C after a precomputation that only involves the parameter [10]. In particular, the class para-NP consists of all parameterized problems that can be fpt-reduced to the trivial parameterized variant of the propositional satisfiability problem, i.e., the problem SAT where the parameter value is a fixed constant for all instances.

The basic complexity classes $\exists^k\forall^*$ and $\forall^k\exists^*$ are defined in terms of the following weighted variant of QSAT$_2$ [13,14].

$\exists^k\forall^*$-WSAT *Instance:* A quantified Boolean formula $\varphi = \exists X.\forall Y.\psi$ and an integer k. *Parameter:* k. *Question:* Does there exist a truth assignment α to X with weight k such that $\forall Y.\psi[\alpha]$ is true?

The class $\exists^k\forall^*$ is defined to be the closure of $\exists^k\forall^*$-WSAT under fpt-reductions. Moreover, its dual class $\forall^k\exists^*$ is defined by $\forall^k\exists^* = $ co-$\exists^k\forall^*$.

We will also consider the variant $\exists^{\leq k}\forall^*$-WSAT of $\exists^k\forall^*$-WSAT, where the problem is to decide whether there exists a truth assignment α to X with weight at most k such that $\forall Y.\psi[\alpha]$ is true. This problem is also $\exists^k\forall^*$-complete. A proof of this can be found in the technical report [15].

Alternating Turing machines We use the same notation as Flum and Grohe [11, Appendix A.1]. Let $m \geq 1$ be a positive integer. An *alternating Turing machine (ATM)* with m tapes is a 6-tuple $\mathbb{M} = (S_\exists, S_\forall, \Sigma, \Delta, s_0, F)$, where: $S_\exists$ and $S_\forall$ are disjoint sets; $S = S_\exists \cup S_\forall$ is the finite set of *states*; Σ is the alphabet; $s_0 \in S$ is the *initial state*; $F \subseteq S$ is the set of *accepting states*; and $\Delta \subseteq S \times (\Sigma \cup \{\$, \square\})^m \times S \times (\Sigma \cup \{\$\})^m \times \{\mathbf{L}, \mathbf{R}, \mathbf{S}\}^m$ is the *transition relation*. The elements of Δ are the *transitions*). The symbols $\$, \square \notin \Sigma$ are special symbols. "\$" marks the left end of any tape. It cannot be overwritten and only allows $\mathbf{R}$-transitions.[1] "$\square$" is the *blank symbol*. Intuitively, the tapes of our machine are bounded to the left and unbounded to the right. The leftmost cell, the 0-th cell, of each tape carries a "\$", and initially, all other tape cells carry the blank symbol. The input is written on the first tape, starting with the first cell, the cell immediately to the right of the "\$". A *configuration* is a tuple $C = (s, x_1, p_1, \dots, x_m, p_m)$, where $s \in S$, $x_i \in \Sigma^*$, and $0 \leq p_i \leq |x_i| + 1$ for each $1 \leq i \leq k$. Intuitively, $\$x_i\square\square\dots$ is the sequence of symbols in the cells of tape i, and the head of tape i scans the p_i-th cell. The *initial configuration* for an input $x \in \Sigma^*$ is $C_0(x) = (s_0, x, 1, \epsilon, 1, \dots, \epsilon, 1)$, where ϵ denotes the empty word. A *computation step* of $\mathbb{M}$ is a pair (C, C') of configurations such that the transformation from C to C' obeys the transition relation. We omit the formal details. We write $C \to C'$ to denote that (C, C') is a computation step of $\mathbb{M}$. If $C \to C'$, we call C' a *successor configuration* of C. A *halting configuration* is a configuration that has no successor configuration. A halting configuration is *accepting* if its state is in F. A step $C \to C'$ is *nondeterministic* if there is a configuration $C'' \neq C'$ such that $C \to C''$, and is *existential* if C is an existential configuration. A state $s \in S$ is called *deterministic* if for any $a_1, \dots, a_m \in \Sigma \cup \{\$, \square\}$, there is at most one $(s, (a_1, \dots, a_m), s', (a'_1, \dots, a'_m), (d_1, \dots, d_m)) \in \Delta$. Similarly, we call a non-halting configuration *deterministic* if its state is deterministic, and *nondeterministic* otherwise. A configuration is called *existential* if it is not a halting configuration and its state is in $S_\exists$, and *universal* if it is not a halting configuration and its state is in $S_\forall$. Intuitively, in an existential configuration, there must be at least one possible run that leads to acceptance, whereas in a universal

[1] To formally achieve that "\$" marks the left end of the tapes, whenever $(s, (a_1, \dots, a_m), s', (a'_1, \dots, a'_m), (d_1, \dots, d_m)) \in \Delta$, then for all $1 \leq i \leq m$ we have that $a_i = \$$ if and only if $a'_i = \$$ and that $a_i = \$$ implies $d_i = \mathbf{R}$.

configuration, all possible runs must lead to acceptance. Formally, a *run* of an ATM $\mathbb{M}$ is a directed tree where each node is labeled with a configuration of $\mathbb{M}$ such that: (1) The root is labeled with an initial configuration. (2) If a vertex is labeled with an existential configuration C, then the vertex has precisely one child that is labeled with a successor configuration of C. (3) If a vertex is labeled with a universal configuration C, then for every successor configuration C' of C the vertex has a child that is labeled with C'. We often identify nodes of the tree with the configurations with which they are labeled. The run is *finite* if the tree is finite, and *infinite* otherwise. The *length* of the run is the height of the tree. The run is *accepting* if it is finite and every leaf is labeled with an accepting configuration. If the root of a run ρ is labeled with $C_0(x)$, then ρ is a run *with input* x. Any path from the root of a run ρ to a leaf is called a *computation path*. The *language (or problem) accepted by* $\mathbb{M}$ is the set $Q_{\mathbb{M}}$ of all $x \in \Sigma^*$ such that there is an accepting run of $\mathbb{M}$ with initial configuration $C_0(x)$. $\mathbb{M}$ *runs in time* $t : \mathbb{N} \to \mathbb{N}$ if for every $x \in \Sigma^*$ the length of every run of $\mathbb{M}$ with input x is at most $t(|x|)$. A *step* $C \to C'$ is an *alternation* if either C is existential and C' is universal, or vice versa. A run ρ of $\mathbb{M}$ is *ℓ-alternating*, for an $\ell \in \mathbb{N}$, if on every path in the tree associated with ρ, there are less than ℓ alternations between existential and universal configurations. The machine $\mathbb{M}$ is *ℓ-alternating* if every run of $\mathbb{M}$ is ℓ-alternating.

3 A First-Order Model Checking Characterization

In this section, we characterize the class $\exists^k\forall^*$ in terms of first-order model checking. Consider the following parameterized problem.

> $\exists^k\forall^*$-MC
> *Instance:* A first-order logic sentence $\varphi = \exists x_1, \dots, x_k.\forall y_1, \dots, y_n.\psi$ over a vocabulary τ, where ψ is quantifier-free, and a finite τ-structure $\mathcal{A}$.
> *Parameter:* The number k of existentially quantified variables of φ.
> *Question:* Does $\mathcal{A} \models \varphi$?

We show that this problem is complete for the class $\exists^k\forall^*$. This result does not imply that $\exists^k\forall^* \subseteq$ A[2] (cf. [11]), because the parameter of the problem $\exists^k\forall^*$-MC is only the number of existential variables, not the size of the entire first-order formula.

Theorem 1. $\exists^k\forall^*$-MC *is* $\exists^k\forall^*$*-complete.*

Proof. We show $\exists^k\forall^*$-membership by giving an fpt-reduction to $\exists^k\forall^*$-WSAT. Let $(\varphi, \mathcal{A})$ be an instance of $\exists^k\forall^*$-MC, where $\varphi = \exists x_1, \dots, x_k.\forall y_1, \dots, y_n.\psi$ is a first-order logic sentence over vocabulary τ, and $\mathcal{A}$ is a τ-structure with domain A. We may assume without loss of generality that ψ contains only connectives $\wedge$ and $\neg$.

We construct an instance (φ', k) of $\exists^k\forall^*$-WSAT, where φ is of the form $\exists X'.\forall Y'.\psi'$. We define $X' = \{\, x'_{i,a} : 1 \leq i \leq k, a \in A \,\}$, and $Y' = \{\, y'_{j,a} : 1 \leq j \leq n, a \in A \,\}$. Intuitively, the variable $x'_{i,a}$ denotes whether the variable x_i is assigned to value a, and similarly, the variable $y'_{j,a}$ denotes whether y_j is assigned to value a. In order to define ψ', we will use the auxiliary function μ on subformulas of ψ, defined by letting $\mu(\chi_1 \wedge \chi_2) = \mu(\chi_1) \wedge \mu(\chi_2)$, $\mu(\neg\chi_1) = \neg\mu(\chi_1)$, and $\mu(\chi) = \bigvee_{1 \leq i \leq u}(\psi_{z_1,a^i_1} \wedge \dots \wedge \psi_{z_m,a^i_m})$ if $\chi = R(z_1, \dots, z_m)$ and $R^{\mathcal{A}} = \{(a^1_1, \dots, a^1_m), \dots, (a^u_1, \dots, a^u_m)\}$, where for each $z \in X \cup Y$ and each $a \in A$ we let $\psi_{z,a} = x'_{i,a}$ if $z = x_i$, and we let $\psi_{z,a} = y'_{j,a}$ if $z = y_j$. Now, we define ψ' by letting $\psi' = \psi'_{\text{unique-}X'} \wedge (\psi'_{\text{unique-}Y'} \rightarrow \mu(\psi))$, where $\psi'_{\text{unique-}X'} = \bigwedge_{1 \leq i \leq k} (\bigvee_{a \in A} x'_{i,a} \wedge \bigwedge_{a,a' \in A, a \neq a'} (\neg x'_{i,a} \vee \neg x'_{i,a'}))$, and $\psi'_{\text{unique-}Y'} = \bigwedge_{1 \leq j \leq n} (\bigvee_{a \in A} y'_{j,a} \wedge \bigwedge_{a,a' \in A, a \neq a'} (\neg y'_{j,a} \vee \neg y'_{j,a'}))$. Intuitively, the formula $\psi'_{\text{unique-}X'}$ represents whether the variables $x'_{i,a}$ encode a unique assignment for each variable x_i. Similarly, the formula $\psi'_{\text{unique-}Y'}$ represents whether the variables $y'_{i,a}$ encode a unique assignment for each variable y_i. We claim that $(\mathcal{A}, \varphi) \in \exists^k\forall^*$-MC if and only if $(\varphi', k) \in \exists^k\forall^*$-WSAT.

Hardness can be shown by means of an fpt-reduction from $\exists^k\forall^*$-WSAT. A detailed proof of both membership and hardness can be found in the technical report [15]. □

4 Alternating Turing Machine Characterizations

Next, we characterize $\exists^k\forall^*$ in terms of ATMs. In particular, we consider parameterized problems related to the halting problem for a particular class of ATMs, and show that these problems are $\exists^k\forall^*$-complete. Moreover, we show that $\exists^k\forall^*$ is exactly the class of parameterized decision problems that can be decided by a certain class of ATMs.

We consider the following restrictions on ATMs. An $\exists\forall$-*Turing machine* (or simply $\exists\forall$-machine) is a 2-alternating ATM $(S_\exists, S_\forall, \Sigma, \Delta, s_0, F)$, where $s_0 \in S_\exists$. Let $\ell, t \geq 1$ be positive integers. We say that an $\exists\forall$-machine $\mathbb{M}$ *halts (on the empty string) with existential cost* ℓ *and universal cost* t if: (1) there is an accepting run of $\mathbb{M}$ with input ϵ, and (2) each computation path of $\mathbb{M}$ contains at most ℓ existential configurations and at most t universal configurations.

Let P be a parameterized problem. An $\exists^k\forall^*$-*machine for* P is a $\exists\forall$-machine $\mathbb{M}$ such that there exists a computable function f and a polynomial p such that: (1) $\mathbb{M}$ decides P in time $f(k) \cdot p(|x|)$; and (2) for all instances (x, k) of P and each computation path R of $\mathbb{M}$ with input (x, k), at most $f(k) \cdot \log |x|$ of the existential configurations of R are nondeterministic. We say that a parameterized problem P *is decided by some* $\exists^k\forall^*$-*machine* if there exists a $\exists^k\forall^*$-machine for P.

Let $m \in \mathbb{N}$ be a positive integer. We consider the following parameterized problem.

> $\exists^k\forall^*$-TM-HALT^m.
> *Instance:* An $\exists\forall$-machine $\mathbb{M}$ with m tapes, and positive integers $k, t \geq 1$.
> *Parameter:* k.
> *Question:* Does $\mathbb{M}$ halt on the empty string with existential cost k and universal cost t?

In addition, we consider the parameterized problem $\exists^k\forall^*$-TM-HALT$^* = \bigcup_{m \in \mathbb{N}} \exists^k\forall^*$-TM-HALT^m, i.e., the variant of the above problem where the number of tapes is given as part of the input, rather than being a fixed constant.

In the remainder of this section, we show that the class $\exists^k\forall^*$ is characterized by alternating Turing machines in the way specified by the following two theorems.

Theorem 2. $\exists^k\forall^*$ *is exactly the class of parameterized decision problems that are decided by some* $\exists^k\forall^*$*-machine.*

Theorem 3. *The problem* $\exists^k\forall^*$-TM-HALT* *is* $\exists^k\forall^*$*-complete, and so is the problem* $\exists^k\forall^*$-TM-HALT^m *for each* $m \in \mathbb{N}$.

Proof (Theorems 2 and 3). In order to show these results, concretely, we will prove the following claims:

1. $\exists^k\forall^*$-TM-HALT* $\leq_{\text{fpt}}$ $\exists^k\forall^*$-MC.
2. For any parameterized problem P that is decided by some $\exists^k\forall^*$-machine with m tapes, it holds that $P \leq_{\text{fpt}} \exists^k\forall^*$-TM-HALT^{m+1}.
3. There is an $\exists^k\forall^*$-machine with a single tape that decides $\exists^{\leq k}\forall^*$-WSAT.
4. Let A and B be parameterized problem. If there is an $\exists^k\forall^*$-machine for B with m tapes, and if $A \leq_{\text{fpt}} B$, then there is an $\exists^k\forall^*$-machine for A with m tapes.
5. $\exists^k\forall^*$-TM-HALT^2 $\leq_{\text{fpt}}$ $\exists^k\forall^*$-TM-HALT^1.

These claims imply the desired results in the following way.

By Claims 1 and 2, by Theorem 1, and by transitivity of fpt-reductions, we have that any parameterized problem P that is decided by an $\exists^k\forall^*$-machine is fpt-reducible to $\exists^k\forall^*$-WSAT, and thus is in $\exists^k\forall^*$. Conversely, let P be any parameterized problem in $\exists^k\forall^*$. Then, by $\exists^k\forall^*$-hardness of $\exists^{\leq k}\forall^*$-WSAT, we know that $P \leq_{\text{fpt}} \exists^{\leq k}\forall^*$-WSAT. By Claims 3 and 4, we know that P is decided by some $\exists^k\forall^*$-machine with a single tape. From this we conclude that $\exists^k\forall^*$ is exactly the class of parameterized problems P decided by some $\exists^k\forall^*$-machine.

Together, Claims 2 and 3 imply that $\exists^{\leq k}\forall^*$-WSAT $\leq_{\text{fpt}}$ $\exists^k\forall^*$-TM-HALT^2. Clearly, for all $m \geq 2$, $\exists^k\forall^*$-TM-HALT^2 $\leq_{\text{fpt}}$ $\exists^k\forall^*$-TM-HALT^m. This gives us $\exists^k\forall^*$-hardness of $\exists^k\forall^*$-TM-HALT^m, for all $m \geq 2$. $\exists^k\forall^*$-hardness of $\exists^k\forall^*$-TM-HALT^1 follows from Claim 5, which states that there is an fpt-reduction from $\exists^k\forall^*$-TM-HALT^2 to $\exists^k\forall^*$-TM-HALT^1. This also implies that $\exists^k\forall^*$-TM-HALT* is $\exists^k\forall^*$-hard. Then, by Claim 1, and since $\exists^k\forall^*$-MC is in $\exists^k\forall^*$ by Theorem 1, we obtain $\exists^k\forall^*$-completeness of $\exists^k\forall^*$-TM-HALT* and $\exists^k\forall^*$-TM-HALT^m, for each $m \geq 1$.

For Claims 1–3, we describe the main idea and intuition behind the proof. A full detailed proof of these claims can be found in the technical report [15].

Proof of Claim 1 (sketch). Given an $\exists\forall$-machine $\mathbb{M}$ with m tapes and positive integers $k, t \geq 1$, we construct a structure $\mathcal{A}$ and a first-order sentence $\exists x_1, \dots, x_{k'}. \forall y_1, \dots, y_u.\psi$ such that $\mathcal{A} \models \varphi$ if and only if $\mathbb{M}$ halts on the empty string with existential cost k and universal cost t. In order to do so, firstly, we transform $\mathbb{M}$ in such a way that each computation path contains exactly k existential configurations and exactly t universal configurations (rather than at most k existential configurations and at most t universal configurations) by adding a "clock" to it, i.e., by indexing the existential and universal states with time steps i and allowing $\mathbb{M}$ to be "idle" at each time step.

Then, we use the existential variables x_i (and the structure $\mathcal{A}$) to guess the first k many (existential) configurations and transitions of $\mathbb{M}$, and we use universal variables (and the structure $\mathcal{A}$) to represent the subsequent t many (universal) configurations and transitions. The position of the tape heads and the tape contents for the first k many configurations can be represented by formulas whose size depends only on k. This is not entirely straightforward, but can be done by adapting a technique used by Flum and Grohe [11, Theorem 7.28] to our setting. In order to represent the position of the tape heads and the tape contents for the universal configurations, we can use additional universally quantified variables, since the number of universal variables is not bounded by the parameter. Finally, it is straightforward to encode into ψ the condition that the computation path of $\mathbb{M}$ that is represented by the variables x_i and y_i must be an accepting run.

Proof of Claim 2 (sketch). Let P be a parameterized problem, and let $\mathbb{M}$ be an $\exists^k\forall^*$-machine that decides it, i.e., there exists a computable function f and a polynomial p such that for any instance (x,k) of P we have that any computation path of $\mathbb{M}$ with input (x,k) has length at most $f(k) \cdot p(|x|)$ and contains at most $f(k) \cdot \log|x|$ nondeterministic existential configurations. Let (x,k) be an instance of P. We construct an $\exists\forall$-machine $\mathbb{M}^{(x,k)}$ and positive integers $k', t \geq 1$ such that $\mathbb{M}^{(x,k)}$ accepts the empty string with existential cost k' and universal cost t if and only if $\mathbb{M}$ accepts (x,k).

In order to do so, we add symbols σ to the alphabet that represent sequences of u many nondeterministic transitions of $\mathbb{M}$, for $u \leq \lceil \log|x| \rceil$. The machine $\mathbb{M}^{(x,k)}$ firstly guesses $f(k)$ of such symbols σ. This can be done using $k' = f(k)$ many existential steps. Then, $\mathbb{M}^{(x,k)}$ simulates using (deterministic) universal steps the existential steps of $\mathbb{M}$ on input (x,k), where it simulates the nondeterministic existential steps of $\mathbb{M}$ by "reading off" the transitions of the guessed symbols σ. Finally, $\mathbb{M}^{(x,k)}$ simulates the (nondeterministic) universal steps of $\mathbb{M}$. The entire simulation of $\mathbb{M}$ on input (x,k) requires at most $t = f(k) \cdot p(|x|)$ universal steps.

Proof of Claim 3 (sketch). We describe the working of an $\exists^k\forall^*$-machine $\mathbb{M}$ for $\exists^{\leq k}\forall^*$-WSAT. Let (φ, k) be an instance of $\exists^{\leq k}\forall^*$-WSAT, where $\varphi = \exists X.\forall Y.\psi$, and $X = \{x_1, \dots, x_n\}$. Firstly, $\mathbb{M}$ determines the size of X, and nondeterministically guesses k many bitstrings of length $\lceil \log|X| \rceil$, which it appends to the tape contents. This can be done using fpt-many existential steps, of which at most $k \cdot \lceil \log|X| \rceil$ many are nondeterministic. These bitstrings represent an assignment $\alpha : X \to \{0,1\}$ of weight at most k in the following way: α sets exactly those x_i to true for which the tape contains a bitstring that is the binary representation of index i. Then, $\mathbb{M}$ uses polynomially many nondeterministic universal steps to guess an assignment $\beta : Y \to \{0,1\}$. Finally, it applies the assignment $\alpha \cup \beta$ to the formula ψ and simplifies the resulting formula, using polynomially many deterministic steps. The machine $\mathbb{M}$ accepts if and only if $\psi[\alpha \cup \beta]$ evaluates to true.

Proof of Claim 4 (sketch). Let R be the fpt-reduction from A to B, and let M be an algorithm that decides B and that can be implemented by an $\exists^k\forall^*$-machine with m tapes. Then, the composition of R and M is an algorithm that decides A. It is straightforward to verify that the composition of R and M can be implemented by an $\exists^k\forall^*$-machine with m tapes.

Proof of Claim 5 (sketch). The claim follows by the following statement, which is known from the literature [16, Thm 8.9 and Thm 8.10]. Let $m \geq 1$ be a (fixed) positive integer. For each ATM $\mathbb{M}$ with m tapes, there exists an ATM $\mathbb{M}'$ with 1 tape such that: (1) $\mathbb{M}$ and $\mathbb{M}'$ are equivalent, i.e., they accept the same language; (2) $\mathbb{M}'$ simulates n many steps of $\mathbb{M}$ using $O(n^2)$ many steps; and (3) $\mathbb{M}'$ simulates existential steps of $\mathbb{M}$ using existential steps, and simulates universal steps of $\mathbb{M}$ using universal steps. ◻

5 Showcase Application to a Combinatorial Problem

In this section, we will exemplify our new complexity toolbox by applying it to various parameterizations of a well-known Π_2^{P}-complete problem, as considered by Ajtai, Fagin, and Stockmeyer [1].

Let $G = (V, E)$ be a graph. We will denote those vertices v that have degree 1 by *leaves*. We call a (partial) function $c : V \to \{1,2,3\}$ a 3-*coloring (of G)*. Moreover, we say that a 3-coloring c is *proper* if c assigns a color to every vertex $v \in V$, and if for each edge $e = \{v_1, v_2\} \in E$ holds that $c(v_1) \neq c(v_2)$. Now consider the following Π_2^{P}-complete decision problem.

3-COL-EXT
Instance: a graph $G = (V, E)$ with n many leaves, and an integer m.
Question: can any 3-coloring that assigns a color to exactly m leaves of G (and to no other vertices) be extended to a proper 3-coloring of G?

We consider several parameterizations $\mathbf{p}$ for this problem, denoted 3-COL-EXT$(\mathbf{p})$.

$\mathbf{p}$	**parameter** (k)
degree	the degree of G, i.e., $k = \deg(G)$
#leaves	the number of leaves of G, i.e., $k = n$
#col.leaves	the number of leaves that are pre-colored, i.e., $k = m$
#uncol.leaves	the number of leaves that are not pre-colored, i.e., $k = n - m$

For most of these parameterizations, the existing parameterized complexity toolbox suffices to determine whether or not an fpt-reduction to SAT exists. The following results witness this (proofs of these results can be found in the technical report [15]). For parameterized problems that are in para-NP, an fpt-reduction to SAT exists, whereas this is not the case for problems that are hard for para-Π_2^{P} (unless the PH collapses).

Proposition 1. *The problems* 3-COL-EXT(degree) *and* 3-COL-EXT(#uncol.leaves) *are* para-Π_2^{P}*-complete. The problem* 3-COL-EXT(#leaves) *is* para-NP*-complete.*

For the remaining parameterization of the problem 3-COL-EXT the classes para-NP and para-Π_2^{P} seem to be of little help. On the one hand, 3-COL-EXT(#col.leaves) is unlikely to be hard for the class para-Π_2^{P}, for the following reason. It is straightforward to construct an xp-reduction from 3-COL-EXT(#col.leaves) to SAT. However, problems that are hard for para-Π_2^{P} do not allow xp-reductions to SAT, unless the PH collapses [14]. Therefore, 3-COL-EXT(#col.leaves) is not para-Π_2^{P}-hard, unless the PH collapses. On the other hand, at first sight it is unclear how one can come up with a more efficient reduction from 3-COL-EXT(#col.leaves) to SAT than the obvious xp-reduction. To back up this conjecture of the non-existence of an fpt-reduction to SAT for the problem 3-COL-EXT(#col.leaves), we will use the class $\exists^k\forall^*$.

In order to give evidence that the problem 3-COL-EXT(#col.leaves) does not allow an fpt-reduction to SAT, we can show that it is hard for the class $\forall^k\exists^*$. In addition, we can illustrate the use of the characterization of the parameterized complexity class $\exists^k\forall^*$ in terms of first-order model checking (Theorem 1), by using the problem $\forall^k\exists^*$-MC (which is the complement of the problem $\exists^k\forall^*$-MC) to show $\forall^k\exists^*$-membership, characterizing the complexity of 3-COL-EXT(#col.leaves) as $\forall^k\exists^*$-complete.

Theorem 4. 3-COL-EXT(#col.leaves) *is* $\forall^k\exists^*$*-complete.*

Proof. To show membership, we give an fpt-reduction from 3-COL-EXT(#col.leaves) to $\forall^k\exists^*$-MC. Let (G, m) be an instance of 3-COL-EXT(#col.leaves), where V' denotes the set of leaves of G, and where $k = m$ is the number of edges that can be pre-colored. Moreover, let $V' = \{v_1, \dots, v_n\}$ and let $V = V' \cup \{v_{n+1}, \dots, v_u\}$. We construct an instance $(\mathcal{A}, \varphi)$ of $\forall^k\exists^*$-MC. We define the domain $A = \{\, a_{v,i} : v \in V', 1 \leq i \leq 3 \,\} \cup \{1, 2, 3\}$. Next, we define $C^{\mathcal{A}} = \{1, 2, 3\}$, $S^{\mathcal{A}} = \{\, (a_{v,i}, a_{v,i'}) : v \in$

$V', 1 \leq i, i' \leq 3\}$, and $F^{\mathcal{A}} = \{(j, j') : 1 \leq j, j' \leq 3, j \neq j'\}$. Then, we can define the formula φ, by letting $\varphi = \forall x_1, \ldots, x_k.\exists y_1, \ldots, y_u.(\psi_1 \rightarrow (\psi_2 \wedge \psi_3 \wedge \psi_4))$, where $\psi_1 = \bigwedge_{1 \leq j < j' \leq k} \neg S(x_i, x_{i'})$, and $\psi_2 = \bigwedge_{1 \leq j \leq u} C(y_j)$, and $\psi_3 = \bigwedge_{v_j \in V', 1 \leq i \leq 3}((\bigvee_{1 \leq \ell \leq k}(x_\ell = a_{v_j,i})) \rightarrow (y_j = i))$, and $\psi_4 = \bigwedge_{\{v_j, v_{j'}\} \in E} F(y_j, y_{j'})$. It is straightforward to verify that $(G, m) \in$ 3-COL-EXT if and only if $\mathcal{A} \models \varphi$.

Intuitively, the assignments to the variables x_i correspond to the pre-colorings of the vertices in V'. This is done by means of elements $a_{v,i}$, which represent the coloring of vertex v with color i. The subformula ψ_1 is used to disregard any assignments where variables x_i are not assigned to the intended elements. Moreover, the assignments to the variables y_i correspond to a proper 3-coloring extending the pre-coloring. The subformula ψ_2 ensures that the variables y_i are assigned to a color in $\{1, 2, 3\}$, the subformula ψ_3 ensures that this coloring extends the pre-coloring encoded by the assignment to the variables x_i, and the subformula ψ_4 ensures that this coloring is proper.

Hardness can be shown by means of an fpt-reduction from $\forall^k\exists^*$-WSAT. A proof of hardness can be found in the technical report [15]. ☐

6 Conclusion

The classes $\exists^k\forall^*$ and $\forall^k\exists^*$ are parameterized complexity classes between the first and the second level of the PH, that can be used to give evidence that certain parameterized problems do not allow an fpt-reduction to SAT. By definition, $\exists^k\forall^*$ and $\forall^k\exists^*$ are characterized in terms of weighted variants of the quantified Boolean satisfiability problem. We provided characterizations of these classes in terms of a first-order logic model checking problem, and in terms of alternating Turing machines with appropriate time bounds and bounds on the number of alternations. Moreover, we showed how one of these alternative characterizations can be used to show membership in the class $\exists^k\forall^*$, by means of an example problem that is related to extending partial graph 3-colorings to complete, proper 3-colorings. Our alternative characterizations establish the robustness of the classes and provide new ways of showing membership.

Further research includes applying the additional characterizations we provided to show membership in $\exists^k\forall^*$ and $\forall^k\exists^*$ for further parameterized problems. In addition, it would be interesting to obtain similar characterizations for the classes $\exists^*\forall^k$-W[t], which are parameterized complexity classes that are defined analogously to $\exists^k\forall^*$, and that can be used to get similar intractability results [14,15].

References

1. Miklós Ajtai, Ronald Fagin, and Larry J. Stockmeyer. The closure of monadic NP. *J. of Computer and System Sciences*, 60(3):660–716, 2000.
2. Sanjeev Arora and Boaz Barak. *Computational Complexity – A Modern Approach*. Cambridge University Press, 2009.
3. Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.
4. Marco Cesati. The Turing way to parameterized complexity. *J. of Computer and System Sciences*, 67:654–685, 2003.

5. Yijia Chen and Jörg Flum. A parameterized halting problem. In Hans L. Bodlaender, Rod Downey, Fedor V. Fomin, and Dániel Marx, editors, *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *Lecture Notes in Computer Science*, pages 364–397. Springer Verlag, 2012.
6. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer Verlag, New York, 1999.
7. Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer Verlag, 2013.
8. Ulle Endriss, Ronald de Haan, and Stefan Szeider. Parameterized complexity results for agenda safety in judgment aggregation. In *Proceedings of the 5th International Workshop on Computational Social Choice (COMSOC-2014)*. Carnegie Mellon University, June 2014.
9. Johannes Klaus Fichte and Stefan Szeider. Backdoors to normality for disjunctive logic programs. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2013)*, pages 320–327. AAAI Press, 2013.
10. Jörg Flum and Martin Grohe. Describing parameterized complexity classes. *Information and Computation*, 187(2):291–319, 2003.
11. Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science. An EATCS Series*. Springer Verlag, Berlin, 2006.
12. Carla P. Gomes, Henry Kautz, Ashish Sabharwal, and Bart Selman. Satisfiability solvers. In *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pages 89–134. Elsevier, 2008.
13. Ronald de Haan and Stefan Szeider. Fixed-parameter tractable reductions to SAT. In Uwe Egly and Carsten Sinz, editors, *Proceedings of the 17th International Symposium on the Theory and Applications of Satisfiability Testing (SAT 2014) Vienna, Austria, July 14–17, 2014*, volume 8561 of *Lecture Notes in Computer Science*, pages 85–102. Springer, 2014.
14. Ronald de Haan and Stefan Szeider. The parameterized complexity of reasoning problems beyond NP. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*. AAAI Press, 2014.
15. Ronald de Haan and Stefan Szeider. The parameterized complexity of reasoning problems beyond NP. Technical Report 1312.1672v3, arXiv.org, 2014.
16. John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley Series in Computer Science. Addison-Wesley-Longman, second edition, 2001.
17. Sharad Malik and Lintao Zhang. Boolean satisfiability from theoretical hardness to practical success. *Communications of the ACM*, 52(8):76–82, 2009.
18. Albert R. Meyer and Larry J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *SWAT*, pages 125–129. IEEE Computer Soc., 1972.
19. Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and its Applications. Oxford University Press, Oxford, 2006.
20. Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
21. Andreas Pfandler, Stefan Rümmele, and Stefan Szeider. Backdoors to abduction. In Francesca Rossi, editor, *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI 2013*. AAAI Press/IJCAI, 2013.
22. Karem A. Sakallah and João Marques-Silva. Anatomy and empirical evaluation of modern SAT solvers. *Bulletin of the European Association for Theoretical Computer Science*, 103:96–121, 2011.
23. Larry J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.
24. Celia Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):23–33, 1976.