Parameterized Proof Complexity: a Complexity Gap for Parameterized Tree-like Resolution

Stefan Dantchev,* Barnaby Martin and Stefan Szeider

Department of Computer Science
Durham University, Durham, England, UK
[s.s.dantchev,b.d.martin,stefan.szeider]@durham.ac.uk

November 24, 2006

Abstract

We propose a proof-theoretic approach for gaining evidence that certain parameterized problems are not fixed-parameter tractable. We consider proofs that witness that a given propositional formula cannot be satisfied by a truth assignment that sets at most k variables to *true*, considering k as the parameter. One could separate the parameterized complexity classes FPT and W[2] by showing that there is no proof system (for CNF formulas) that admits proofs of size $f(k)n^{O(1)}$ where f is a computable function and n represents the size of the propositional formula. We provide a first step and show that tree-like resolution does not admit such proofs. We obtain this result as a corollary to a meta-theorem, the main result of this paper. The meta-theorem extends Riis' Complexity Gap Theorem for tree-like resolution. Riis' result establishes a dichotomy between polynomial and exponential size tree-like resolution proofs for propositional formulas that uniformly encode a first-order principle over a universe of size n: (1) either there are tree-like resolution proofs of size polynomial in n, or (2) the proofs have size at least $2^{\varepsilon n}$ for some constant ε; the second case prevails exactly when the first-order principle has no finite but some infinite model.

We show that the parameterized setting allows a refined classification, splitting the second case into two subcases: (2a) there are tree-like resolution proofs of size at most $\beta^k n^\alpha$ for some constants α, β; or (2b) every tree-like resolution proof has size at least n^{k^γ} for some constant $0 < \gamma \leq 1$; the latter case prevails exactly if for every infinite model, a certain associated hypergraph has no finite dominating set. We provide examples of first-order principles for all three cases.

*Most of the work was done while visiting the Isaac Newton Institute for Mathematical Sciences as a participant in the Logic and Algorithms (LAA) Programme.

1 Introduction

In recent years parameterized complexity and fixed-parameter algorithms have become an important branch of algorithm design and analysis; hundreds of research papers have been published in the area [1, 4, 5, 8]. In parameterized complexity one considers computational problems in a two-dimensional setting: the first dimension is the usual *input size* n, the second dimension is a positive integer k, the *parameter*. A problem is fixed-parameter tractable if it can be solved in time $O(f(k)n^{O(1)})$ where f denotes a computable, possibly exponential, function. Several NP-hard problems have natural parameterizations that admit fixed-parameter tractability. For example, given a graph with n vertices, one can check in time $O(1.273^k + nk)$ (and polynomial space) whether the graph has a vertex cover of size at most k [2]. On the other hand, several parameterized problems such as CLIQUE (has a given graph a clique of size at least k?) are believed to be *not* fixed-parameter tractable. BOUNDED CNF SATISFIABILITY is a further problem that is believed to be *not* fixed-parameter tractable (and which will play a special role in the sequel): given a propositional formula in conjunctive normal form, is there a satisfying truth assignment that sets at most k variables to *true*?

Parameterized complexity offers also a completeness theory. Numerous parameterized problems that appear to be not fixed-parameter tractable have been classified as being complete under *fpt-reductions* for complexity classes of the so-called *weft hierarchy* W[1] $\subseteq$ W[2] $\subseteq \cdots$. For example, CLIQUE and BOUNDED CNF SATISFIABILITY are complete for the first two levels of the weft hierarchy, respectively. We will outline the basic notions of parameterized complexity in Section 2.1; for an in-depth treatment of parameterized complexity classes and fpt-reduction we refer the reader to Flum and Grohe's monograph [5].

It is widely believed that problems that are hard for the weft hierarchy are not fixed-parameter tractable. Up to now there are mainly three types of evidence:

1. *Accumulative evidence*: numerous problems are known which are hard or complete for classes of the weft hierarchy, and for which no fixed-parameter algorithm has been found in spite of considerable efforts [1].

2. *k-step Halting Problems* for non-deterministic Turing machines are complete for the classes W[1] (single-tape) and W[2] (multi-tape) [5]. A Turing machine is such an opaque and generic object that it does not appear reasonable that we should be able to decide if a given Turing machine on a given input has some accepting path without looking at the paths.

3. If a problem that is hard for a class of the weft hierarchy turns out to be fixed-parameter tractable, then the *Exponential Time Hypothesis* (ETH) fails, i.e., there is a $2^{o(n)}$ time algorithm for the n-variable 3-SAT problem [6]. ETH is closely related to the parameterized complexity class M[1] which lies between FPT and W[1] (see [5]).

We propose a new approach for gaining further evidence that certain parameterized problems are not fixed-parameter tractable. We generalize concepts of proof complexity to the two-dimensional setting of parameterized complexity. This allows us to formulate a parameterized version of the program of Cook and Reckhow [3]. Their program attempts to gain evidence for NP $\neq$ co-NP, and in turn for P $\neq$ NP, by showing that propositional proof systems are not polynomially bounded. We introduce the concept of parameterized proof systems; in our program, lower bounds for the length of proofs in these new systems yield evidence that certain parameterized problems are not fixed-parameter tractable.

In propositional proof complexity one usually constructs a sequence of tautologies (or contradictions), and shows that the sequence requires proofs (or refutations) of super-polynomial size in the proof system under consideration. In the scenario of contradictions and refutations, such sequences of propositional formulas frequently encode a first-order (FO) sentence (such as the pigeon hole principle) where the n-th formula of the sequence states that the FO sentence has no model of size n. S. Riis [10] established a meta-theorem that exactly pinpoints under which circumstances a given FO sentence gives rise to a sequence of propositional formulas that have polynomial-sized refutations in the system of tree-like resolution. Namely, if the sequence has not tree-like resolution refutations of polynomial size, then shortest tree-like resolution refutations have size at least $2^{\varepsilon n}$ for a positive constant ε that only depends on the FO sentence. Hence there is a *gap* between two possible proof complexities. The case of exponential size prevails exactly when the FO sentence has no finite but some infinite model.

In this paper we show a meta-theorem regarding the complexity of parameterized tree-like resolution. To this aim we consider *parameterized contradictions* which are pairs $(\mathcal{F}, k)$ where $\mathcal{F}$ is a propositional formula in CNF and k is an integer, such that $\mathcal{F}$ cannot be satisfied by a truth assignment that sets at most k variables to *true*. Parameterized contradictions form a co-W[2]-complete language. Hence FPT $=$ W[2] would follow if there were a proof system that admits proofs of size at most $f(k)n^{O(1)}$ for parameterized contradictions $(\mathcal{F}, k)$ where n represents the size of $\mathcal{F}$; we call such a (hypothetical) proof system *fpt-bounded*.

In this paper we consider the relatively weak system of tree-like resolution. A tree-like resolution refutation for a parameterized contradiction $(\mathcal{F}, k)$ uses clauses with more than k negated variables as additional axioms. We show a meta-theorem that classifies exactly the complexity of tree-like resolution refutations for parameterized contradictions. Our theorem allows a refined view of the exponential case of Riis' Theorem: Consider the sequence $\langle \mathcal{C}_{\psi,n} \rangle_{n\in\mathbb{N}}$ of propositional formulas generated from a FO sentence ψ that has no finite but some infinite model. For a positive integer k we get a sequence of parameterized contradictions $\langle (\mathcal{C}_{\psi,n}, k) \rangle_{n\in\mathbb{N}}$. We show that exactly one of the following two cases holds (and provide a criterion that decides which one).

- $(\mathcal{C}_{\psi,n}, k)$ has a tree-like resolution refutation of size $\beta^k n^\alpha$ for some constants α and β which depend on ψ only.
- There exists a constant γ, $0 < \gamma \leq 1$, such that for every $n > k$, every tree-like resolution refutation of $(\mathcal{C}_{\psi,n}, k)$ is of size at least n^{k^γ}.

We establish the upper bound $\beta^k n^\alpha$ via certain boolean decision trees. For the lower bound n^{k^γ} we use a game theoretic argument.

We provide examples of FO sentences for each of the above categories. In particular, the examples for the n^{k^γ} case (Examples 6 and 7) show that parameterized tree-like resolution is not fpt-bounded.

2 Preliminaries

2.1 Fixed-parameter Tractability

In the following let Σ denote an arbitrary but fixed finite alphabet. A *parameterized language* is a set $L \subseteq \Sigma^* \times \mathbb{N}$ where $\mathbb{N}$ denotes the set of positive integers. If (I, k) is in a parameterized language L, then we call I the *main part* and k *the parameter*. We identify a parameterized language with the decision problem "$(I, k) \in L$?" and will therefore synonymously use the terms *parameterized problem* and parameterized language. A parameterized problem L is called *fixed-parameter tractable* if membership of (I, k) in L can be deterministically decided in time

$$O(f(k)|I|^{O(1)}) \tag{1}$$

where f denotes a computable function. FPT denotes the class of all fixed-parameter tractable decision problems; algorithms that achieve the time complexity (1) are called *fixed-parameter algorithms*. The key point of this definition is that the exponential growth is confined to the parameter only, in contrast to running times of the form

$$O(|I|^{O(f(k))}). \tag{2}$$

There is theoretical evidence that parameterized problems like CLIQUE are *not* fixed-parameter tractable. This evidence is provided via a completeness theory which is similar to the theory of NP-completeness. This completeness theory is based on the following notion of reductions: Let $L_1 \in \Sigma_1^* \times \mathbb{N}$ and $L_2 \in \Sigma_2^* \times \mathbb{N}$ be parameterized problems. An *fpt-reduction* from L_1 to L_2 is a mapping $R : \Sigma_1^* \times \mathbb{N} \to \Sigma_2^* \times \mathbb{N}$ such that

1. $(I, k) \in L_1$ if and only if $R(I, k) \in L_2$.
2. R is computable by a fixed-parameter algorithm, i.e., there is a computable function f such that $R(I, k)$ can be computed in time $O(f(k)|I|^{O(1)})$.

3. There is a computable function g such that whenever $R(I,k) = (I', k')$, then $k' \le g(k)$.

A parameterized complexity class $\mathcal{C}$ is the equivalence class of a parameterized problem under fpt-reductions. It is easy to see that FPT is closed under fpt-reductions, thus FPT is a parameterized complexity class. Parameterized problems appear to have several degrees of intractability, as manifested by the *weft hierarchy*. The classes W[t] of this hierarchy form a chain

$$\text{FPT} \subseteq \text{W[1]} \subseteq \text{W[2]} \subseteq \cdots \subseteq \text{XP}$$

where all inclusions are assumed to be proper. Here XP denotes the class of problems solvable in time $O(|I|^{f(k)})$; it is known that FPT $\neq$ XP [4]. Each class W[t] is defined as the equivalence class of a certain canonical weighted satisfiability problem for decision circuits. For W[2] the canonical problem is equivalent to the following satisfiability problem:

WEIGHTED CNF SATISFIABILITY

Instance: A propositional formula $\mathcal{F}$ in conjunctive normal form (CNF), and a positive integer k.

Parameter: k.

Question: Can $\mathcal{F}$ be satisfied by a truth assignment τ that sets exactly k variables to *true*? (k is the *weight* of τ.)

Note that if the clauses of the CNF formula are required to contain at most three literals, we get the W[1]-complete problem WEIGHTED 3-CNF SATISFIABILITY. Let BOUNDED CNF SATISFIABILITY denote the problem obtained from WEIGHTED CNF SATISFIABILITY by allowing truth assignments of weight *at most* k. It is easy to see that this relaxation does not change the parameterized complexity of the problem:

Lemma 1. BOUNDED CNF SATISFIABILITY *is complete for the class* W[2] *under fpt-reductions.*

Proof. We provide an fpt-reduction from the language WEIGHTED CNF SATISFIABILITY, which is known to be W[2]-complete [4]. Let $(\mathcal{F}, k)$ be an instance of WEIGHTED CNF SATISFIABILITY in which the variables $v_1', \dots, v_k'$ do not appear. We reduce it to the instance $(\mathcal{F}', k+1)$ of BOUNDED CNF SATISFIABILITY in which $\mathcal{F}' := \mathcal{F} \wedge (v_1' \vee \dots \vee v_k')$. It is transparent that F has a satisfying assignment of weight k if and only if F' has a satisfying assignment of weight at most $k+1$, and the result follows. □

As in classical complexity theory, we can define for a parameterized complexity class $\mathcal{C}$ the complementary complexity class co-$\mathcal{C} = \{\, \overline{L} : L \in \mathcal{C} \,\}$ where $\overline{L} = (\Sigma^* \times \mathbb{N}) \setminus L$ for a parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$. Clearly FPT = co-FPT. It is easy to see that if $\mathcal{C}$ is closed under fpt-reductions, then so is co-$\mathcal{C}$. Thus, in particular, each class W[t] of the weft hierarchy gives rise to a parameterized complexity class co-W[t].

2.2 Parameterized Proof Systems

Definition 1. *Let $L \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized language. A* parameterized proof system for *L is an onto mapping $\Gamma : (\Sigma_1^* \times \mathbb{N}) \to L$ for some alphabet Σ_1 where Γ can be computed by a fixed-parameter algorithm.*

We say that Γ is fpt-bounded *if there exist computable functions f and g such that for every $(I, k) \in L$ there is $(I', k') \in \Sigma_1^* \times \mathbb{N}$ with $\Gamma(I', k') = (I, k)$, $|I'| = O(f(k)|I|^{O(1)})$, and $k' \le g(k)$.*

Note that the problems of the classes W[t] of the weft hierarchy have fpt-bounded proof systems since the yes-instances of these problems have small witnesses. Consider, for example, the W[2]-complete problem L = BOUNDED CNF SATISFIABILITY. Let $S_{\mathcal{F},\tau,k}$ denote a string over some alphabet Σ that encodes a CNF formula $\mathcal{F}$ together with a satisfying truth assignment τ of weight $\le k$ for $\mathcal{F}$. A proof system Γ for L can now be defined by setting $\Gamma(w, k) = (\mathcal{F}, k)$ if w encodes $S_{\mathcal{F},\tau,k}$, and otherwise $\Gamma(w, k) = (\mathcal{F}_0, k_0)$ for some fixed $(\mathcal{F}_0, k_0) \in L$. Evidently, Γ is fpt-bounded.

However, the situation is different for the classes co-W[t]; specifically, in this case, for co-W[2]. We can witness that a CNF formula with n variables has no satisfying assignment of weight $\le k$ by listing all $O(k \cdot n^k)$ assignments of weight $\leq k$, then checking that none is satisfying. However, this listing requires too much space and apparently we cannot use it for the construction of an fpt-bounded proof system.

This next result follows by a standard argument in which the computation of a Turing machine is considered as a proof.

Lemma 2. *Let $\mathcal{C}$ be a parameterized complexity class and let L be a* co-$\mathcal{C}$*-complete parameterized problem. If there is no fpt-bounded proof system for L, then $\mathcal{C} \neq$* FPT.

Proof. Let $L \subseteq \Sigma^* \times \mathbb{N}$ be a co-$\mathcal{C}$-complete parameterized problem. We show the contra-positive of the statement. Assume $\mathcal{C} = \text{FPT}$. Since FPT $=$ co-FPT, co-$\mathcal{C} = \text{FPT}$ follows. Consequently, there is a fixed-parameter algorithm that decides membership in L; let M be a Turing machine that implements this algorithm. For $(I, k) \in L$ let $M_{(I,k)}$ be a string over some alphabet Σ_1 that encodes the computation steps of M with input (I, k). By the fixed-parameter tractability of L, there is a computable function f such that $|M_{(I,k)}| \leq O(f(k)|I|^{O(1)})$. We may assume that (I, k) can be read off from $M_{(I,k)}$, say, by choosing an encoding where (I, k) is encoded as a prefix of $M_{(I,k)}$ where k is presented in unary. We define a mapping $\Gamma : \Sigma_1^* \times \mathbb{N} \to L$ as follows. Consider $(I', k') \in \Sigma_1^* \times \mathbb{N}$. If I' encodes a computation of M for the input (I, k), i.e. if $I' = M_{(I',k')}$, then we let $\Gamma(I', k') = (I, k)$. Otherwise, if (I', k') does not encode a computation of M for some input (I, k), we put $\Gamma(I', k') = (I_0, k_0)$ for some arbitrary fixed $(I_0, k_0) \in L$. Clearly Γ is a proof system for L as $\Gamma(I', k')$ can be computed in linear time. Furthermore, Γ is fpt-bounded, since $|M_{(I,k)}| \leq O(f(k)|I|^{O(1)})$ holds for $(I, k) \in L$. ☐

In view of this lemma we suggest a program à la Cook-Reckhow for gaining evidence that the complexity classes from the weft hierarchy are distinct from FPT. This program consists of showing that particular parameterized proof systems are not fpt-bounded. For such an approach we would start with a weak system such as a parameterized version of tree-like resolution. The consideration of stronger systems is left for future research.

2.3 From First-Order to Propositional Logic

Next we describe a translation of a FO sentence to a sequence of propositional CNF formulas. We use the language of FO logic with equality but with neither function nor constant symbols. We omit functions and constants only for the sake of a clearer exposition; note that we may simulate constants in a single FO sentence with added *outermost* existential quantification on new variables replacing those constants. We assume that the FO sentence is given as a conjunction of FO sentences, each of which is in prenex normal form; thus, we need only explain the translation of a single FO sentence in prenex normal form. The case of a purely universal sentence is easy – a sentence ψ of the form

$$\forall x_1, x_2, \ldots x_k \; \mathcal{F}(x_1, x_2, \ldots x_k),$$

where $\mathcal{F}$ is quantifier-free, is translated into a sequence of propositional formulas in CNF $\langle \mathcal{C}_{\psi,n} \rangle_{n \in \mathbb{N}}$, of which the n-th member $\mathcal{C}_{\psi,n}$ is constructed as follows. Let $[n] = \{1, 2, \ldots n\}$. For instantiations $x_1, x_2, \ldots x_k \in [n]$, we can consider $\mathcal{F}(x_1, x_2, \ldots x_k)$ to be a propositional formula over propositional variables of two different kinds: $R(x_{i_1}, x_{i_2}, \ldots x_{i_p})$, where R is a p-ary predicate symbol, and $(x_i = x_j)$. We transform $\mathcal{F}$ into CNF and then take the union of all such CNF formulas for $(x_1, x_2, \ldots x_k)$ ranging over $[n]^k$. The variables of the form $(x_i = x_j)$ evaluate to either true or false, thus we are left with variables of the form $R(x_{i_1}, x_{i_2}, \ldots x_{i_p})$ only.

The general case, a sentence ψ of the form

$$\forall x_1 \exists y_1 \forall x_2 \exists y_2 \ldots \forall x_k \exists y_k \; \mathcal{F}(x_1, x_2, \ldots x_k, y_1, y_2, \ldots y_k),$$

can be reduced to the previous case by Skolemization. We introduce *Skolem relations* $S_i(x_1, x_2, \ldots x_i, y_i)$ for $1 \leq i \leq k$. $S_i(x_1, x_2, \ldots x_i, y_i)$ witnesses y_i for any given $x_1, x_2, \ldots x_i$, so we need to add *Skolem clauses* stating that such a witness always exists, i.e.,

$$\bigvee_{y_i=1}^{n} S_i(x_1, x_2, \ldots x_i, y_i) \qquad \text{for all } (x_1, x_2, \ldots x_i) \in [n]^i \,.$$

The original sentence can be transformed into the following purely universal sentence

$$\forall x_1, x_2, \ldots x_k, y_1, y_2, \ldots y_k \ \bigwedge_{i=1}^{k} \neg S_i(x_1, x_2, \ldots x_i, y_i) \vee \mathcal{F}(x_1, x_2, \ldots x_k, y_1, y_2, \ldots y_k).$$

By construction it is clear that, for FO sentences ψ, the CNF formula $\mathcal{C}_{\psi,n}$ is satisfiable if and only if ψ has a model of size n. Thus satisfiability questions on the sequence $\langle \mathcal{C}_{\psi,n} \rangle_{n \in \mathbb{N}}$ relate to questions on the existence of non-empty finite models for ψ.

Example 1. We consider (the negation of) the pigeonhole principle to be defined by the following sentence ψ^{PHP} of FO.

$$\forall x \exists y R(x, y) \ \wedge \ \exists y \forall x \neg R(x, y) \ \wedge \ (\forall x \forall w \forall y \ \neg R(x, y) \vee \neg R(w, y) \vee x = w).$$

We translate this to the conjunction of the following universal clauses

$$\begin{array}{l} \forall x \forall y \ \neg S_2(x, y) \vee R(x, y) \\ \forall y \forall x \ \neg S_1(y) \vee \neg R(x, y) \\ \forall x \forall y \forall w \ \neg R(x, y) \vee \neg R(w, y) \vee x = w \end{array}$$

together with the Skolem clauses

$$\begin{array}{c} \forall x \exists y S_2(x, y) \\ \exists y S_1(y). \end{array}$$

For $x, y \in [n]$ we now consider $R(x, y)$, $S_2(x, y)$ and $S_1(y)$ to be propositional variables. $\mathcal{C}_{\psi,n}$ is therefore the system of clauses

$$\begin{array}{l} \neg S_2(x, y) \vee R(x, y), \ \neg S_1(y) \vee \neg R(x, y) \ \text{ and} \\ \neg R(x, y) \vee \neg R(w, y), \text{ for } x, y, w \in [n],\, w \neq x, \end{array}$$

together with the Skolem clauses

$$\bigvee_{i=1}^{n} S_2(x, i), \text{ for } x \in [n], \text{ and } \bigvee_{i=1}^{n} S_1(i).$$

Note that the size of $C_{\psi,n}$, with respect to a reasonable encoding, is polynomial in n. Let us briefly explain this point. In our translation of ψ we generate a constant number p of predicate CNF clauses, each involving at most p' free FO variables (once the universal quantifiers are dropped). We also generate a constant number q of Skolem relations, each involving at most q' free FO variables (once the universal quantifiers are dropped; we consider the final, witness variable to remain existentially quantified). After instantiating these free variables with the elements of $[n]$, we deduce that $C_{\psi,n}$ involves at most $p \cdot n^{p'} + q \cdot n^{q'}$ (propositional) clauses. Since these (propositional) clauses are of length bound by $\max\{p', q' \cdot n\}$, the result follows.

2.4 Parameterized Tree-like Resolution

A *literal* is either a propositional variable or the negation of a propositional variable. A *clause* is a disjunction of literals (and a propositional variable can appear only once in a clause). A set of clauses is a conjunction, i.e., it is *satisfiable* if there exists a truth assignment satisfying simultaneously all the clauses. *Resolution* is a proof system designed to *refute* a given set of clauses, i.e., to prove that it is unsatisfiable. This is done by means of a single derivation rule

$$\frac{C \vee v \quad \neg v \vee D}{C \vee D},$$

which we use to obtain a new clause from two already existing ones. The goal is to derive the empty clause – resolution is known to be sound and complete, i.e., we can derive the empty clause from the initial clauses if and only if the initial set of clauses was unsatisfiable.

In this paper, we shall work with a restricted version of resolution, namely *tree-like resolution*. In tree-like resolution we are not allowed to reuse any clause that has already been derived, i.e., we need to derive a clause as many times as we use it (this, of course, does not apply to the initial clauses). In other words, a tree-like resolution refutation can be viewed as a binary tree whose nodes are labeled with clauses. Every leaf is labeled with one of the original clauses, every clause at an internal node is obtained by a resolution step from the clauses at its two children nodes, and the root of the tree is labeled with the empty clause. We measure the *size* of a tree-like resolution refutation by the number of nodes.

It is not hard to see that a tree-like resolution refutation of a given set of clauses is equivalent to a *boolean decision tree* solving the *search problem* for that set of clauses. The search problem for an unsatisfiable set of clauses is defined as follows (see, e.g., Krajíček's book [7]): given a truth assignment, find a clause which is falsified under the assignment. A boolean decision tree solves the search problem by querying values of propositional variables and then branching on the answer. Without loss of generality, we may assume that no propositional variable is questioned twice on the same branch and that a branch of the tree is closed as soon as a falsified clause is found, under the partial assignment – conjunction of facts – obtained so far along that branch. When a branch is thus closed we say that an *elementary contradiction* has been obtained. Note that we consider a node of the decision tree to be labeled by the conjunction of facts thus far obtained together with the propositional variable there questioned. This is analogous to a node in a tree-like resolution refutation being labeled with its clause together with the variable about to be resolved. Given the equivalence between tree-like resolution refutations and boolean decision trees, we shall concentrate on the latter. Whenever we need to show that there is a certain tree-like resolution refutation of some unsatisfiable set of clauses, we shall construct a boolean decision tree for the corresponding search problem. On the other hand, whenever we claim a tree-like resolution lower bound, we shall prove it by an adversary argument against any boolean decision tree which solves the search problem.

We give working definitions of parameterized contradiction and parameterized tree-like resolution, which we shall use to state and prove the complexity gap for parameterized tree-like resolution.

Definition 2. *A* parameterized contradiction *is a pair* $(\mathcal{F}, k)$ *where* $\mathcal{F}$ *is a propositional CNF formula and* k *is a positive integer such that* $\mathcal{F}$ *has no satisfying assignment of weight at most* k*.*

Example 2. Let us consider an undirected graph $G = (V, E)$ that does not have a vertex cover of size $\leq k$. We introduce a propositional variable p_v for every vertex $v \in V$. Then the pair

$$\left(\bigwedge_{\{u,v\}\in E}(p_u \vee p_v), k\right)$$

is a parameterized contradiction.

Let PARAMETERIZED CONTRADICTIONS be the language of parameterized contradictions. Note that PARAMETERIZED CONTRADICTIONS is the complement of BOUNDED CNF SATISFIABILITY and, as such, is co-W[2]-complete under fpt-reductions. *Parameterized tree-like resolution* is a proof system designed to refute a parameterized contradiction $(\mathcal{F}, k)$. It should be viewed as a tree-like resolution refutation on $\mathcal{F} \wedge \mathcal{G}_k$, where $\mathcal{G}_k$ is the conjunction of *all* clauses that contain at least $k+1$ negated variables (where the variables occurring in the clauses of $\mathcal{G}_k$ range over those occurring in $\mathcal{F}$). The conjunction $\mathcal{G}_k$ should be seen neither as part of the given parameterized contradiction nor as part of the refutation (except for those individual clauses of $\mathcal{G}_k$ that are actually used, which do form part of the refutation). As we have explained, we prefer the equivalent formulation in terms of boolean decision trees. It is straightforward to verify that a parameterized tree-like resolution refutation is equivalent to a parameterized boolean decision tree (in which all the branches are closed), defined as follows.

Definition 3. *Given a parameterized contradiction* $\mathcal{P} = (\mathcal{F}, k)$*, a* parameterized boolean decision tree *is a decision tree that queries values of propositional variables and branches on the answers; a branch of the tree is closed as soon as (1) or (2) happens:*

(1) an elementary contradiction is reached, i.e. the partial assignment obtained along the branch falsifies $\mathcal{F}$*;*

(2) the partial assignment obtained along the branch has more than k *propositional variables set to true, i.e., has weight* $> k$*.*

3 Complexity Gap for Parameterized Tree-like Resolution

We first recall the complexity gap theorem for tree-like resolution proven by Riis [10].

Theorem 1. *Given a FO sentence ψ which fails in all finite models, consider its translation into a sequence of propositional CNF contradictions $\langle \mathcal{C}_{\psi,n} \rangle_{n\in\mathbb{N}}$. Then either 1 or 2 holds:*

1. *$\mathcal{C}_{\psi,n}$ has polynomial-size in n tree-like resolution refutations.*
2. *There exists a positive constant ε such that for every n, every tree-like resolution refutation of $\mathcal{C}_{\psi,n}$ is of size at least $2^{\varepsilon n}$.*

Furthermore, 2 holds if and only if ψ has an infinite model.

In the parameterized setting, one can hope that the second case above, the hard one, splits into two subcases. This is indeed true as we shall prove the following complexity gap theorem for *parameterized* tree-like resolution:

Theorem 2. *Given a FO sentence ψ, which fails in all finite models but holds in some infinite model, consider the sequence of parameterized contradictions $\langle \mathcal{D}_{\psi,n,k} \rangle_{n\in\mathbb{N}} = \langle (\mathcal{C}_{\psi,n}, k) \rangle_{n\in\mathbb{N}}$ where $\langle \mathcal{C}_{\psi,n} \rangle_{n\in\mathbb{N}}$ is the translation of ψ already defined. Then either 2a or 2b holds:*

2a. *$\mathcal{D}_{\psi,n,k}$ has a parameterized tree-like resolution refutation of size $\beta^k n^\alpha$ for some constants α and β which depend on ψ only.*

2b. *There exists a constant γ, $0 < \gamma \leq 1$, such that for every $n > k$, every parameterized tree-like resolution refutation of $\mathcal{D}_{\psi,n,k}$ is of size at least n^{k^γ}.*

Furthermore, 2b holds if and only if ψ has an infinite model whose induced hypergraph has no finite dominating set.

By proving that Case 2b can be attained (see Examples 6 and 7), and bearing in mind the remark from the end of Section 2.3, we derive the following as a corollary.

Corollary 1. *Parameterized tree-like resolution is not fpt-bounded.*

If we could prove that no parameterized proof system for PARAMETERIZED CONTRADICTIONS is fpt-bounded, then we would have derived W[2] $\neq$ FPT.

Before we prove Theorem 2, we need to give some definitions. For a model M, let $|M|$ denote the universe of M. Given a model M of a FO sentence ψ, either finite or infinite, the *hypergraph induced by the model* M has the elements of $|M|$ as vertices and as hyperedges those sets $\{y_1, \dots y_l\}$ such that $(y_1, \dots, y_l)$ appears as a tuple in some relation. (Recall that there are two kinds of relations – the extensional R relations which are present in the original FO sentence, and the S relations that we introduce when Skolemizing the sentence – both give rise to hyperedges.) A set of vertices is *independent* if it contains no hyperedge as a subset. Given a set X of vertices, a vertex $y \notin X$, and a set A such that $X \cup \{y\} \subseteq A \subseteq |M|$, we say that *$y$ is A-independent from X* if and only if (i) there is no self-loop $\{y\}$ at y, and (ii) there is no hyperedge $E \subseteq A$ which contains y and intersects with X. We say that y is independent from X if y is $|M|$-independent from X; otherwise we say that X *dominates* y. Finally, a *dominating set* is a set X of vertices that dominates every other vertex of the hypergraph.

3.1 Case 2a of Theorem 2

We now prove Case 2a of Theorem 2. We shall start by reproving Case 1 of Theorem 1. Note that our proof is different from Riis' proof [10] as our translation, though equivalent, is slightly different.

Proof of Case 1, Theorem 1. The idea is to take a (finite) resolution refutation of the FO formula ψ (such a refutation exists as the formula has no model), and to transform it into a polynomial size in n tree-like resolution refutation of $\mathcal{C}_{\psi,n}$.

As we have explained, we can consider a boolean decision tree instead of a tree-like resolution refutation. In the FO case, constructing a boolean decision tree is very similar to producing a tableau refutation. (Our method therefore differs slightly from simply inverting the classical FO resolution, as we consider only instantiations of terms as opposed to terms themselves.) The decision tree tries to build up a model of ψ, starting by witnessing some unary Skolem relation ψ with the constant 1 and deriving further constants as Skolem witnesses of already derived constants as and when necessary[1]. Note that, while we do not allow constants in our signatures, we refer to those elements that have been mentioned in decision tree questions as constants.

Let C be the set of constants thus far witnessed, and let $\overline{c}$ be some tuple over C. At each point two kinds of queries are allowed: (I) querying the boolean value of some $R_i(\overline{c})$ and (II) querying the witness y of some $S_j(\overline{c}, y)$. In the latter case there are two possibilities for y: it could be a constant that is already known or it could be a new one, thus extending the set of constants. For Case I, the branching factor is 2: corresponding to $R_i(\overline{c})$ being true ($\top$) or false ($\bot$). For Case II, the branching factor is $|C| + 1$: we label these branches with the elements of C or a new constant c' according to the conceded witness for $S_j(\overline{c}, y)$.

The order in which the boolean decision tree performs these queries is as follows. We start with the single constant 1, witnessing a unary Skolem relation of ψ, i.e. set $C := \{1\}$, and first query all possible R_i relations on all possible tuples over C, closing any branch as soon as a contradiction is reached. We then pick up a Skolem relation $S_j(\overline{c}, y)$ and a j-tuple $\overline{c}$ of constants of C and query the witness y. There are $|C|+1$ possible outcomes – y is either one of the already known constants from C or a different constant, which we denote by c'. If $y \in C$, we pick another $S_{j'}(\overline{c}', y)$ and do the same (we assume a reasonable order over the Skolem relations S_j and tuples in C). In the case where y is a new constant which is not in C, we extend the set of constants, i.e. set $C := C \cup \{c'\}$ and repeat the same procedure, i.e. query all possible R_i relations over all possible tuples in the expanded C and so on.

It is easy to see that the boolean decision tree constructed in this way is finite. Indeed, suppose it were infinite. Then, by König's Lemma, there must be an infinite branch which constitutes an infinite model of ψ – a contradiction. Let the depth of this tree be h and the maximum size of C along any of its branches be m. Let us now turn this finite refutation of ψ into polynomial size in n refutation of $\mathcal{C}_{\psi,n}$. We note that a node, which queries an R_i relation in the FO case, remains the same in the propositional case, and, in particular, has a branching factor 2. A node, which witnesses a Skolem relation $S_j(\overline{c}, y)$, is of constant branching factor in the FO case (bounded by m). In the propositional case, such a node can be translated into a sequence of n nodes, the l-th node querying the $S_j(\overline{c}, l)$ only if all the nodes $S_j(\overline{c}, 1)$, $S_j(\overline{c}, 2), \ldots, S_j(\overline{c}, l-1)$ got negative answers. If the answers to all queries were negative, we arrive at a contradiction with the clause $\bigvee_{y=1}^{n} S_j(\overline{c}, y)$, while a positive answer gives us the desired witness. Thus a node querying an S relation in the FO case can be thought as a single node of branching factor n in the propositional case. As the FO tree is of constant height h that depends on the formula ψ only, the boolean decision tree in the propositional case is of size at most $(\max\{m, n\})^h$ which is $O(n^h)$, i.e., polynomial in n as claimed. ☐

We can now modify the proof above in order to prove Case 2a of Theorem 2.

Proof of Case 2a, Theorem 2. We shall construct a boolean decision tree for the parameterized FO case in a similar manner, but with the following modification: whenever we witness a new constant and extend the set of constants by adding it, we add *another* new constant that is *independent* from all the others. That is, we actually introduce new constants to C in pairs, c' and c'', where c' is a Skolem witness for some constant in C and c'' is assumed independent from $C \cup \{c'\}$ (we make no assumption of the independence of c' from C). Thereafter, we may also close branches whenever we directly contradict the independence of c'' from $C \cup \{c'\}$. Now, suppose for the sake of contradiction that the boolean decision tree constructed in this way is infinite. Again, by König's Lemma, there must be an infinite branch which constitutes an infinite model of ψ with the additional property that it has no finite dominating set. Indeed, by the construction, for every finite set of constants, we always add a new constant that is independent from the set. This gives us the desired contradiction, thus showing that the decision tree we have constructed is finite. Let the depth of this tree be h and the maximum size of C along any of its branches be m.

What remains is to estimate the branching factor of the queries in the propositional case. The R and S queries have branching factors 2 and n as before. The only problem is in finding a new constant that is independent from all existing constants. The boolean decision tree in the propositional case can "search" for such a constant in the following way. Denote the set of elements of the finite universe $[n]$ that have not been queried at all so far by $Z = \{z_1, z_2, \ldots z_p\}$ and the set of already

[1]As is customary in Proof Complexity, we discount the empty model. It is, therefore, possible to have ψ with no finite models and no outermost existential quantifier. In this case we may instantiate a single constant at the outset to get us going.

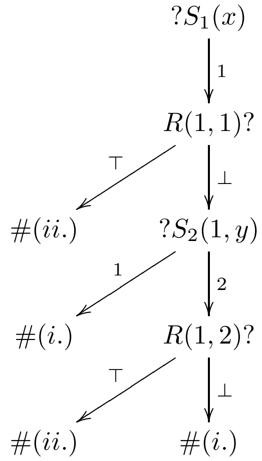

Figure 1: Decision tree for Example 3.

known constants by C. The decision tree first queries all possible R and S relations with arguments over $C \cup \{z_1\}$ that could possibly make z_1 dominated by C. If all answers are negative then z_1 is independent from C, so it is success – z_1 is added to C and we proceed further according to the decision tree in the FO case. Otherwise, on the first positive answer (i.e., having found out that z_1 is dominated by C), we abandon z_1 and proceed the same way with z_2 and so on. For every z_i which we query the branching factor is bounded by m^{ab} where a is the maximum arity of any relation of ψ and b is the number of relations of ψ (including Skolem relations in both cases). On the other hand, we do not need to test more than k elements of Z as we are now in the parameterized setting where the boolean decision tree cannot take more than k positive answers and we need to move onto a new element of Z on a positive answer only. This gives us a subtree of height k and branching factor m^{ab}, which is equivalent to a single node of branching factor m^{abk}. To conclude, let us recall that the parameterized FO tree was of constant height h that depends on the formula ψ only, and thus, the boolean decision tree in the parameterized propositional case is of size at most $(\max\{m^{abk}, n\})^h$ which is not greater than $(m^{abh})^k n^h$ as claimed. $\square$

Example 3. We give an example of a decision tree constructed as in Case 1, Theorem 1. We consider the following sentence ψ which has no models:

$$\forall x \exists y\ R(x,y) \ \ \wedge \ \ \exists x \forall y\ \neg R(x,y).$$

As per our translation to propositional clauses, this is equivalent to the conjunction of the universal clauses

$$(i.) \quad \forall x \forall y\ \neg S_2(x,y) \vee R(x,y) \text{ and}$$
$$(ii.) \quad \forall x \forall y\ \neg S_1(x) \vee \neg R(x,y),$$

together with the Skolem clauses

$$\forall x \exists y\ S_2(x,y) \text{ and}$$
$$\exists x\ S_1(x).$$

Figure 1 shows a FO decision tree for this system of clauses. The number following each # specifies the clause that has been contradicted. For example, the bottom right # comes from the knowledge $S_2(1,2)$ and $\neg R(1,2)$ – which contradicts the first universal clause.

Example 4. We give an example of a decision tree constructed as in Case 2a, Theorem 2. We consider the sentence ψ which is the conjunction of the following.

$$\begin{array}{cr}
\exists x\ U(x) & U\text{-existence} \\
\forall x\ \neg U(x) \vee \neg R(x,x) & U\text{-antireflexivity} \\
\forall x \forall y\ \neg U(x) \vee \neg U(y) \vee \neg R(x,y) \vee \neg R(y,x) & U\text{-antisymmetry} \\
\forall x \forall y \forall z\ \neg U(x) \vee \neg U(y) \vee \neg U(z) \vee \neg R(x,y) \vee \neg R(y,z) \vee R(x,z) & U\text{-transitivity} \\
\forall x \forall y\ \neg U(x) \vee \neg U(y) \vee R(x,y) \vee R(y,x) & U\text{-totality} \\
\forall y \exists x\ U(y) \rightarrow (U(x) \wedge R(x,y)) & U\text{-non-minimality} \\
\exists x \forall y\ U(y) \vee R(x,y) & \neg U\text{-dominator}
\end{array}$$

The sentence ψ asserts the existence of a bipartition, in which the U-part is a non-empty strict total R-order without minimal element, and such that there is a single element with an R-edge to all the elements of the $\neg U$-part. Depending on which part this single element is in, a model of ψ will have a dominating set of size 1 or 2. As per our translation, this is equivalent to the universal clauses

$$\begin{array}{lc}
(i.) & \forall x\ \neg S_1(x) \vee U(x) \\
(ii.) & \forall x\ \neg U(x) \vee \neg R(x,x) \\
(iii.) & \forall x \forall y\ \neg U(x) \vee \neg U(y) \vee \neg R(x,y) \vee \neg R(y,x) \\
(iv.) & \forall x \forall y \forall z\ \neg U(x) \vee \neg U(y) \vee \neg U(z) \vee \neg R(x,y) \vee \neg R(y,z) \vee R(x,z) \\
(v.) & \forall x \forall y\ \neg U(x) \vee \neg U(y) \vee R(x,y) \vee R(y,x) \\
(vi.) & \forall y \forall x\ \neg S_2(x,y) \vee \neg U(y) \vee U(x) \\
(vi'.) & \forall y \forall x\ \neg S_2(x,y) \vee \neg U(y) \vee R(x,y) \\
(vii.) & \forall x \forall y\ \neg S_3(x) \vee U(y) \vee R(x,y),
\end{array}$$

together with the Skolem clauses

$$\begin{array}{c}
\exists x\ S_1(x) \\
\forall y \exists x\ S_2(x,y) \\
\exists x S_3(x).
\end{array}$$

Note that the Skolem relation S_1 is somewhat redundant and is included for the sake of formality (it would preserve meaning if we were to remove clause $(i.)$ and substitute $\exists x\ U(x)$ for the Skolem clause $\exists x\ S_1(x)$). Figure 2 shows a FO decision tree for this system in the parameterized case. (Note that we have questioned constants and relations in an intelligent, rather than natural, order. This is so that we might keep the size of the tree to a minimum; the tree would still close if we chose a natural order.) The bullet points (•) indicate where, having just witnessed a new constant, we introduce another new, independent constant. In the decision tree, we know that 2 must be independent from 1, and that 4 must be independent from 1, 2 and 3; we do not know that 3 is independent from either 2 or 1. The contradictions labeled with square brackets arise from violating the independence condition. For example, at $\#[1,4]$ we have just learned the truth of $R(1,4)$, which violates the assumed independence of 1 and 4.

The height of our tree is $h = 9$ and we never involve more than $m = 4$ constants; the maximum arity is $a = 2$ and there are $b = 5$ involved relations. As in the previous proof, using the bound $(m^{abh})^k n^h$, we can state that $\mathcal{D}_{\psi,n,k}$ has a parameterized tree-like resolution refutation of size bounded by $2^{180k} n^9$.

Owing to the rules that allow us to introduce independent constants, the character of the FO decision tree in the parameterized case is different from the ordinary FO decision tree. Notice that we have closed our tree without witnessing the Skolem relation $S_1(x)$. It would not be possible to close an ordinary FO decision tree without this, since, without the U-existence clause $(i.)$, ψ has finite models.

We conclude this section with a further example of Case 2a of Theorem 2. This specimen provides a somewhat trivial instance, having, as it does, parameterized tree-like resolution refutations not just polynomial in n, but actually independent of n (in contrast to Example 4 where the size of a smallest tree-like refutation depends on n).

Example 5. We consider the (negation of the) least number principle for total orders. Let ψ^{LNP_1} be

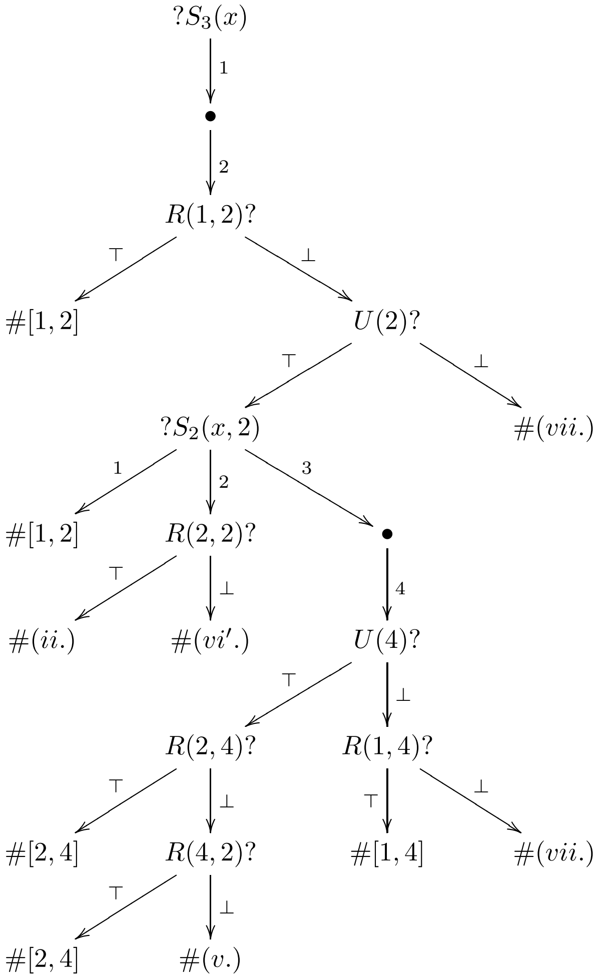

Figure 2: Decision tree for Example 4

.

the conjunction of the following.

$\forall x\, \neg R(x,x)$	antireflexivity
$\forall x \forall y\, \neg R(x,y) \vee \neg R(y,x)$	antisymmetry
$\forall x \forall y \forall z\, \neg R(x,y) \vee \neg R(y,z) \vee R(x,z)$	transitivity
$\forall x \forall y R(x,y) \vee R(y,x)$	totality
$\forall y \exists x\, R(x,y)$	no least element

All models of ψ^{LNP_1} have a dominating set of size 1; moreover, every element of the model constitutes such a dominating set. It is straightforward to verify that $\langle \mathcal{D}_{\psi^{\mathrm{LNP}_1},n,k} \rangle_{n\in\mathbb{N}}$ has tree-like resolution refutations of size $2k$, independent from n.

3.2 Case 2b of Theorem 2

We now turn our attention to proving Case 2b of Theorem 2. Our argument will be facilitated by a game based on those described by Pudlák [9] and Riis [10] in which *Prover* (female) plays against *Adversary* (male). In this game, a strategy for Prover gives rise to a boolean decision tree on a set of clauses. Prover questions the propositional variables that label the nodes of the tree and Adversary attempts to answer these so as neither to violate any specific clause nor to have conceded that more than k variables are true ($\top$), for in either of these situations Prover is deemed the winner. Of course, assuming the set of clauses was unsatisfiable, Adversary is destined to lose: the question is how large he can make the tree in the process of losing. Note that each branch of the tree corresponds to a play of this game, hence each decision tree corresponds to a Prover strategy. We will be concerned with Adversary strategies that perform well over all Prover strategies, and hence induce a lower bound on all decision trees and, consequently, all parameterized tree-like resolution refutations.

When considering a certain Prover strategy – a decision tree – we will actually consider only a certain subtree in which the missing branches correspond to places where Adversary has simply given up, already conceding the imminent violation of a clause. In this way, there are two types of non-leaf nodes in this subtree, those of out-degree 1 in which Adversary's decision was *forced* (because he conceded defeat on the alternative valuation) and those of out-degree 2 in which he is happy to continue on either outcome. In the latter case, we may consider that he has given Prover a *free choice* as to the value of the relevant variable. The free choice nodes play a vital role in ensuring the large size of this subtree, which in turn places a lower bound on the size of the decision tree of which it is a subset.

Let $\mathcal{C}_{\psi,n}$ be the propositional translation of some FO sentence ψ which has no finite models, but holds in some infinite model. We formally define the game $\mathcal{G}(\mathcal{C}_{\psi,n}, k)$ as follows. At each turn Prover selects a propositional variable of $\mathcal{C}_{\psi,n}$ that she has not questioned before, and Adversary responds either by answering that the variable is true ($\top$) or that it is false ($\bot$), or by allowing Prover a free choice over those two. The Prover wins if at any point she holds information that contradicts a clause of $\mathcal{C}_{\psi,n}$ or she holds more than k variables evaluated true. In this formalism, given a Prover strategy on her moves, and considering both possibilities on the free choice nodes, we generate a *game tree*, the subtree of the decision tree alluded to in the previous paragraph.

Henceforth, we consider only the case in which some model of ψ has no finite dominating set. We will give a strategy for Adversary in the game $\mathcal{G}(\mathcal{C}_{\psi,n}, k)$ that guarantees a large game tree for all opposing Prover strategies.

Adversary's Strategy At any point in the game – node in the game tree – Adversary will have conceded certain information to Prover. He always has in mind two disjoint sets of already mentioned constants P and Q on which he has conceded certain information: initially these sets are both empty. The set Q is to be a ($P \cup Q$)-independent set whose members are also ($P \cup Q$)-independent from P. In some sense P is the only set of constants for which Adversary has actually conceded an interpretation; all he concedes of Q is that it is a floating set with certain independence properties. If X is a set of constants, let $\mathcal{M}_X$ be the class of models of ψ that are consistent with the information Adversary has conceded on X. At each point Prover will ask Adversary a question of the form $R_i(\overline{c})$ or $S_j(\overline{c})$. The Adversary answers as follows:

I. If all constants of $\overline{c}$ are in P, then Adversary should choose some model in $\mathcal{M}_P$ and answer according to that.

II. If all constants of $\overline{c}$ are in $P \cup Q$, and there is at least one from Q, then Adversary should answer false ($\bot$).

III. If some constant in $\overline{c}$ is not in $P \cup Q$ then

- if no model in $\mathcal{M}_P$ satisfies the question, then Adversary should answer false ($\bot$), otherwise
- he should give Prover a free choice on the question.

In all cases the sets P and Q remain the same, except in Case III Part 2. If the Prover chooses true ($\top$), then Adversary places all the constants of $\overline{c}$ in P, possibly removing some from Q in the process. If the Prover chooses false ($\bot$), then Adversary places any constants in $\overline{c}$ that are not already in $P \cup Q$ into Q. It turns out that, in Cases II and III, the situation never arises in which Adversary is forced to answer true. In particular, in Case III, it will never be the case that all models in $\mathcal{M}_P$ satisfy the question. This is vital to the success of Adversary's strategy, and we will return to it later. We must now prove that this strategy leads to a large decision tree; we will need the following two lemmas.

Lemma 3. *Let ψ be a sentence of FO in which no model has a finite dominating set. Let M be a model of ψ and let P be a finite subset of $|M|$. For any positive integer q, there exists an independent set Q of size q such that all elements of Q are independent from P.*

Proof. Suppose for contradiction some M fails to have this property. Consider any finite P, of size p, in $|M|$. If there is a q such that all sets $Q \subseteq |M| \setminus P$ are either not independent or some element in Q is not independent from P, then there is a maximal q_0, the cardinality of a set Q_0, that is independent and whose elements are independent from P. But, $P \cup Q_0$ is now a finite dominating set of M by the maximality of q_0. □

Lemma 4. *Consider any path in the game tree of $\mathcal{G}(\mathcal{C}_{\psi,n}, k)$ from the root to a leaf. If there are k or fewer propositional variables evaluated to true by the leaf, then every one of the n constants must have appeared in a free choice node along that path.*

Proof. We give a sketch proof of the lemma; for a fuller explanation, see Riis' paper [10]. It is important to see that Adversary plays faithfully according to some (infinite) models of ψ, because this means that an elementary contradiction can only be reached by the violation of a Skolem clause. In order to see that Adversary plays so, it becomes necessary to explain why in Case II of his strategy he never loses any of his putative models $\mathcal{M}_P$ and why in Case III he is never forced to answer true ($\top$).

In Case II, Adversary never loses a model M in $\mathcal{M}_P$ because Q can always be chosen to be independent, and independent from P, by Lemma 3. Indeed, if such an interpretation is put on Q in M, then Adversary's answer is forced to be false ($\bot$).

Suppose, in Case III, that Adversary were forced to answer true ($\top$), i.e., all models M in $\mathcal{M}_P$ satisfy the question $R_i(\overline{c})$ or $S_j(\overline{c})$. By the floating nature of all elements that are not in P this would generate a finite dominating set of $P \cup Q$ on M. Let us dwell on this point further. Let $\overline{c}'$ be the subtuple of $\overline{c}$ consisting of those constants of the latter that are not in $P \cup Q$. Some of the constants of $\overline{c}'$ could have been mentioned in questions before, but only in ones for which Adversary's response had been forced false. Suppose that $P \cup Q$ were not a dominating set for M, then there exists an element $x \in M$, independent from $P \cup Q$. But this element is such that it can fill the tuple $\overline{c}'$ and falsify $R_i(\overline{c})$ or $S_j(\overline{c})$ in M (and falsify any questions that previously involved it, which had already been answered false). This contradicts the question having been forced true in the first place.

Recalling that we can only reach an elementary contradiction by the violation of a Skolem clause, we can now complete the proof. Let c' be a constant that never appears in a free choice node in our game tree. In order to violate a Skolem clause, Adversary must have denied some $S(\overline{c}, x)$, for each of the n constants substituted for x. But that his denial of $S(\overline{c}, c')$ was forced implies a contradiction. Since c' is uninterpreted in any of the models in $\mathcal{M}_P$, it follows that $S(\overline{c}, c'')$ is false for all c'' in any model in $\mathcal{M}_P$. This tells us that $\mathcal{M}_P$ is empty and, consequently, that ψ had no infinite model. □

We are now in a position to argue the key lemma in this section.

Lemma 5. *Let a be the maximum arity of any relation in ψ and suppose that there are no more than b different relations in the propositional translation of ψ (including Skolem relations in both cases). Following the strategy that we have detailed for the game $\mathcal{G}(\mathcal{C}_{\psi,n}, k)$, and with p and q the cardinality of the sets P and Q, respectively, Adversary cannot lose while both $p < k^{1/ab}$ and $p + q < n$.*

Proof. Consider the game tree of $\mathcal{G}(\mathcal{C}_{\psi,n}, k)$. Note that Adversary only answers true in the case that all involved constants are then added to his set P, or, of course, were already there. Thus, at a certain node in the game tree, the number of true answers given is trivially bounded by the size of the set of all possible questions on P, which is certainly bound by p^{ab}. Hence, whilst $p^{ab} < k$, there must be fewer than k propositional variables evaluated to true. Furthermore, if $p + q < n$ at this node, then not all of the n constants can have appeared in a free choice (since constants that have appeared in a free choice are necessarily added to either P or Q). It follows from the previous lemma that Adversary has not yet lost. □

We are now in a position to settle Case 2b.

Proof of Case 2b, Theorem 2. We aim to provide a lower bound on the size of any game tree for $\mathcal{G}(\mathcal{C}_{\psi,n}, k)$. Since a lower bound on the size of a game tree induces a lower bound on the size of a boolean decision tree, the result follows.

Consider a game tree for $\mathcal{G}(\mathcal{C}_{\psi,n}, k)$. Recall that, at any node in this tree, Adversary has in mind two sets P and Q, of size p and q, respectively, and, by the previous lemma, whilst $p < k^{1/ab}$ and $p + q < n$, he has not lost. Consider, therefore any node in this game tree and the sets P and Q that Adversary there has in mind. Let $S(p, q)$ be some monotonic decreasing function that provides a lower bound on the size of the subtree of the game tree rooted at the chosen node; whence $S(0, 0)$ is a lower bound on the size of the game tree itself. In showing that $S(p, q)$ satisfies the recurrence relation

- $S(p, q) \geq S(p + a, q) + S(p, q + a) + 1$, with
- $S(p, q) \geq 0$, when $p \geq k^{1/ab}$ or $p + q \geq n$,

we are able to derive the following statement whose full proof appears in the appendix.

Let n, k, a and b be positive integers such that

$$\text{(i.) } a \geq 2 \quad \text{(ii.) } n > k \quad \text{(iii.) } n \geq 7\,a + 1 \quad \text{(iv.) } k^{1/ab} \geq (16a^2)^2,$$

then

$$S(0, 0) \geq n^{k^\gamma} \text{ where } \gamma := 1/16a^3b.$$

The result follows immediately from this statement for sufficiently large k ($\geq (16a2)^{2ab}$) and n ($\geq 7a + 1$). By noting that all boolean decision trees of Case 2b are of size ≥ 2, we can modify the given γ to one that works for all $n, k \geq 1$. Note that the assumption that (maximum arity) $a \geq 2$ is innocuous – there are no unary FO sentences φ which have no finite models but possess an infinite one, therefore we would be in neither Case 2a nor Case 2b. □

Example 6. We consider the (negation of the) least number principle for partial orders. Let $\psi^{\mathrm{LNP}_\infty}$ be the conjunction of the FO clauses given in Example 5 without the forth clause (totality). $\psi^{\mathrm{LNP}_\infty}$ has models without a finite dominating set. For example, if $\mathbb{Z}$ is the set of integers, then $\mathbb{N} \times \mathbb{Z}$ under the strict partial ordering

$$(n, z) \prec (n', z') \text{ if and only if } n = n' \text{ and } z < z'$$

provides such a model.

Example 7. We return to the sentence ψ^{PHP} defined in Example 1. This has models without a finite dominating set: for example the positive integers $\mathbb{N}$, with $R(x, y) \Leftrightarrow y = x + 1$, provides such a model.

References

[1] Cesati, M.: *Compendium of parameterized problems.* `http://bravo.ce.uniroma2.it/home/cesati/research/compendium.pdf` (September 2006)

[2] Chen, J., Kanj, I.A., Xia, G.: Simplicity is beauty: Improved upper bounds for vertex cover. Technical Report TR05-008, DePaul University, Chicago IL (2005)

[3] Cook, S., Reckhow, R.: On the lengths of proofs in the propositional calculus: preliminary version. In: *Sixth Annual ACM Symposium on Theory of Computing* (Seattle, Wash., 1974). Assoc. Comput. Mach., New York (1974) 135–148

[4] Downey, R.G., Fellows, M.R.: *Parameterized Complexity.* Monographs in Computer Science. Springer Verlag (1999)

[5] Flum, J., Grohe, M.: *Parameterized Complexity Theory.* Volume XIV of Texts in Theoretical Computer Science. An EATCS Series. Springer Verlag (2006)

[6] Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? *J. of Computer and System Sciences* **63**(4) (2001) 512–530

[7] Krajíček, J.: *Bounded Arithmetic, Propositional Logic, and Complexity Theory.* Cambridge University Press, New York, NY, USA (1995)

[8] Niedermeier, R.: *Invitation to Fixed-Parameter Algorithms.* Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press (2006)

[9] Pudlák, P.: Proofs as Games. *American Mathematical Monthly*, **107**(6) (2000) 541–550

[10] Riis, S.: A complexity gap for tree-resolution. *Computational Complexity* **10**(3) (2001) 179–209

Appendix

Lemma 6. *Let n, k, a and b be positive integers such that*

$$\text{(i.) } a \geq 2 \quad \text{(ii.) } n > k \quad \text{(iii.) } n \geq 7a + 1 \quad \text{(iv.) } k^{1/ab} \geq (16a^2)^2,$$

then

$$S(0,0) \geq n^{k^\gamma} \text{ where } \gamma := 1/16a^3b.$$

The reminder of the appendix is devoted to proving this lemma.

We consider the combinatorial choose function $\binom{m}{r}$ to be defined on the integers and to be $m!/(r!(m-r!))$, when $m \geq r$ and m and r are non-negative, and to be 0 otherwise. The proof of Lemma 6 requires several technical lemmas.

Lemma 7. *Consider a game tree for $\mathcal{G}(\mathcal{C}_{\psi,n}, k)$. Let a be the maximum arity of any relation of ψ and let b be the number of relations of ψ (including Skolem relations in both cases). Then a monotonic decreasing subtree size-bounding function $S(p,q)$ satisfies the following properties:*

- $S(p,q) \geq S(p+a,q) + S(p,q+a) + 1$*, with*
- $S(p,q) \geq 0$*, when $p \geq k^{1/ab}$ or $p + q \geq n$.*

Proof. The second part follows from Lemma 5. For the first part, we consider only the free choice branching points in the game tree – that is we consider the binary tree that is a minor of the game tree in the natural way. At these points, on answering true, some constants – at most a – may be added to P. Some may have been taken from Q, but since the function S is monotonic decreasing the bound still holds. If the answer is false then at most a constants may be added to Q and the bound holds for similar reasons. □

Lemma 8. *The recurrence relation of the previous lemma satisfies:*

$$S(p,q) \geq \binom{\lfloor \frac{n-p-q}{a} \rfloor}{\lfloor \frac{k^{1/ab}-p}{a} \rfloor} - 1$$

Proof. By induction on the (binary tree minor of the) game tree for $\mathcal{G}(\mathcal{C}_{\psi,n}, k)$, starting from the the leaves.

(Base case.) The choose function evaluates to 1 or 0 when $p+q \geq n$ or $p \geq k^{1/ab}$. Since $S(p,q)$ is defined and is always ≥ 0, the bound holds.

(Inductive step.) Assume the solution holds for m steps, or less, in from a leaf. We will prove that it holds for $m+1$ steps in. Consider such a (free choice) node. Then $S(p,q) := S(p+a,q) + S(p,q+a) + 1$ where the two child nodes are m or fewer steps from the leaves. So, by the inductive hypothesis, we have

$$\begin{aligned} S(p,q) &\geq \binom{\lfloor \frac{n-p-q-a}{a} \rfloor}{\lfloor \frac{k^{1/ab}-p-a}{a} \rfloor} + \binom{\lfloor \frac{n-p-q-a}{a} \rfloor}{\lfloor \frac{k^{1/ab}-p}{a} \rfloor} - 2 + 1 \\ &= \binom{\lfloor \frac{n-p-q}{a} \rfloor - 1}{\lfloor \frac{k^{1/ab}-p}{a} \rfloor - 1} + \binom{\lfloor \frac{n-p-q}{a} \rfloor - 1}{\lfloor \frac{k^{1/ab}-p}{a} \rfloor} - 2 + 1 \\ &\geq \binom{\lfloor \frac{n-p-q}{a} \rfloor}{\lfloor \frac{k^{1/ab}-p}{a} \rfloor} - 1. \end{aligned}$$

□

Lemma 9. *Let m and r be positive integers such that $m \geq r^2$; $m \geq 7$. Then*

$$\binom{m}{r} \geq m^{r/4}.$$

Proof. From $m \geq r^2$; $m \geq 7$, we may derive $(m - r) \geq (m - m^{1/2}) \geq m^{3/4}$. Thence

$$\binom{m}{r} \geq \frac{(m-r)^r}{r^r} \geq \frac{(m - m^{1/2})^r}{m^{r/2}} \geq \frac{m^{3r/4}}{m^{r/2}} = m^{r/4}.$$

□

Lemma 10. *Let m, r and c be non-negative reals such that $c \geq 1$.*

A. If $m \geq c^{c/(c-1)}$ then $\left(\frac{m}{c}\right)^r \geq m^{r/c}$.

B. If $r \geq 2$ and $m \geq 3$ then $(m-1)^r - 1 \geq m^{r/2}$.

C. If $m^r \geq c^{c/(c-1)}$ then $\frac{m^r}{c} \geq m^{r/c}$.

Proof. May be easily verified. □

We are now in a position to proceed with the proof of Lemma 6.

Proof of Lemma 6. By Lemma 9, and the knowledge that the preconditions yield both $\lfloor \frac{n}{a} \rfloor \geq 7$ and $\lfloor \frac{n}{a} \rfloor \geq \lfloor \frac{k^{1/ab}}{a} \rfloor^2$ we have that

$$\binom{\lfloor \frac{n}{a} \rfloor}{\lfloor \frac{k^{(1/ab)}}{a} \rfloor} - 1 \geq \left\lfloor \frac{n}{a} \right\rfloor^{\frac{1}{4}\lfloor \frac{k^{1/ab}}{a} \rfloor} - 1.$$

Noting that the preconditions yield $n \geq 1$ and $k^{1/ab} \geq 1$, we derive

$$\left\lfloor \frac{n}{a} \right\rfloor^{\frac{1}{4}\lfloor \frac{k^{1/ab}}{a} \rfloor} - 1 \geq \frac{(n-1)}{a}^{\frac{k^{1/ab} - a}{4a}} - 1.$$

Now, by Part A of the previous lemma, together with the knowledge that the preconditions yield $n - 1 \geq a^{a/(a-1)}$, we have that

$$\frac{(n-1)}{a}^{\frac{k^{1/ab} - a}{4a}} - 1 \geq (n-1)^{\frac{k^{1/ab} - a}{4a^2}} - 1.$$

By Part B of the previous lemma, together with the fact that the preconditions yield $\frac{k^{1/ab} - a}{4a} \geq 2$ and $n - 1 \geq 3$, we have

$$(n-1)^{\frac{k^{1/ab} - a}{4a^2}} - 1 \geq n^{\frac{k^{1/ab} - a}{8a^2}}.$$

Noting that the preconditions yield $\frac{k^{1/ab}}{8a^2} - \frac{a}{8a^2} \geq \frac{k^{1/ab}}{16a^2}$ we derive

$$n^{\frac{k^{1/ab} - a}{8a^2}} \geq n^{\frac{k^{1/ab}}{16a^2}}.$$

Finally, we deploy Part C of the previous lemma, together with the knowledge that precondition $(iv.)$ yields $k^{1/ab} \geq (16a^2)^{16a^2/(16a^2-1)}$, to demonstrate that

$$n^{\frac{k^{1/ab}}{16a^2}} \geq n^{k^{1/16a^3 b}}.$$

Hence we have shown that

$$\binom{\lfloor \frac{n}{a} \rfloor}{\lfloor \frac{k^{(1/ab)}}{a} \rfloor} - 1 \geq n^{k^{\gamma}} \text{ where } \gamma := 1/16a^3 b.$$

However, it follows from Lemma 8 that

$$S(0,0) \geq \binom{\lfloor \frac{n}{a} \rfloor}{\lfloor \frac{k^{1/ab}}{a} \rfloor} - 1,$$

so our proof of Lemma 6 is concluded. □