On the Descriptive Complexity of Color Coding

Max Bannach
Institute for Theoretical Computer Science, Universität zu Lübeck, Germany
bannach@tcs.uni-luebeck.de

Till Tantau
Institute for Theoretical Computer Science, Universität zu Lübeck, Germany
tantau@tcs.uni-luebeck.de

Abstract

Color coding is an algorithmic technique used in parameterized complexity theory to detect "small" structures inside graphs. The idea is to derandomize algorithms that first randomly color a graph and then search for an easily-detectable, small color pattern. We transfer color coding to the world of descriptive complexity theory by characterizing – purely in terms of the syntactic structure of describing formulas – when the powerful second-order quantifiers representing a random coloring can be replaced by equivalent, simple first-order formulas. Building on this result, we identify syntactic properties of first-order quantifiers that can be eliminated from formulas describing parameterized problems. The result applies to many packing and embedding problems, but also to the long path problem. Together with a new result on the parameterized complexity of formula families involving only a fixed number of variables, we get that many problems lie in FPT just because of the way they are commonly described using logical formulas.

2012 ACM Subject Classification Theory of computation → Finite Model Theory; Theory of computation → Fixed parameter tractability

Keywords and phrases color coding, descriptive complexity, fixed-parameter tractability, quantifier elimination, para-AC^0

Digital Object Identifier 10.4230/LIPIcs.STACS.2019.11

Related Version `https://arxiv.org/abs/1901.03364`

1 Introduction

Descriptive complexity provides a powerful link between logic and complexity theory: We use a logical formula to *describe* a problem and can then infer the computational complexity of the problem just from the *syntactic structure* of the formula. As a striking example, Fagin's Theorem [9] tells us that 3-colorability lies in NP just because its describing formula ("there exist three colors such that all adjacent vertex pairs have different colors") is an existential second-order formula. In the context of fixed-parameter tractability theory, methods from descriptive complexity are also used a lot – but commonly to show that problems are *difficult.* For instance, the A- and W-hierarchies are defined in logical terms [11], but their hard problems are presumably "beyond" the class FPT of fixed-parameter tractable problems.

The methods of descriptive complexity are only rarely used to show that problems are *in* FPT. More precisely, the syntactic structure of the natural logical descriptions of standard parameterized problems found in textbooks are not known to imply that the problems lie in FPT – even though this is known to be the case for many of them. To appreciate the underlying difficulties, consider the following three parameterized problems: p-MATCHING, p-TRIANGLE-PACKING, and p-CLIQUE. In each case, we are given an undirected graph as input and a number k and we are then asked whether the graph contains k vertex-disjoint edges (a size-k matching), k vertex-disjoint triangles, or a clique of size k, respectively. The problems are known to have widely different complexities (maximal matchings can actually be found in polynomial time, triangle packing lies at least in FPT, while finding cliques is

W[1]-complete) but *very* similar logical descriptions:

$$\alpha_k = \exists x_1 \cdots \exists x_{2k} \big(\textstyle\bigwedge_{i \neq j} x_i \neq x_j \wedge \bigwedge_{i=1}^{k} Ex_{2i-1}x_{2i}\big), \tag{1}$$

$$\beta_k = \exists x_1 \cdots \exists x_{3k} \big(\textstyle\bigwedge_{i \neq j} x_i \neq x_j \wedge \bigwedge_{i=1}^{k} (Ex_{3i-2}x_{3i-1} \wedge Ex_{3i-2}x_{3i} \wedge Ex_{3i-1}x_{3i})\big), \tag{2}$$

$$\gamma_k = \exists x_1 \cdots \exists x_k \big(\textstyle\bigwedge_{i \neq j} x_i \neq x_j \wedge \bigwedge_{i \neq j} Ex_ix_j\big). \tag{3}$$

The family $(\alpha_k)_{k \in \mathbb{N}}$ of formulas is clearly a natural "slicewise" description of the matching problem: A graph $\mathcal{G}$ has a size-k matching if, and only if, $\mathcal{G} \models \alpha_k$. The families $(\beta_k)_{k \in \mathbb{N}}$ and $(\gamma_k)_{k \in \mathbb{N}}$ are natural parameterized descriptions of the triangle packing and the clique problems, respectively. Well-known results on the descriptive complexity of parameterized problems allow us to infer [11] from the above descriptions that all three problems lie in W[1], but offer no hint why the first two problems actually lie in the class FPT – syntactically the clique problem arguably "looks like the easiest one" when in fact it is semantically the most difficult one. The results of this paper will remedy this: We will show that the syntactic structures of the formulas α_k and β_k imply membership of p-MATCHING and p-TRIANGLE-PACKING in FPT.

The road to deriving the computational complexity of parameterized problems just from the syntactic properties of slicewise first-order descriptions involves three major steps: First, a characterization of when the color coding technique is applicable in terms of syntactic properties of second-order quantifiers. Second, an exploration of how these results on second-order formulas apply to first-order formulas, leading to the notion of *strong* and *weak* quantifiers and to an elimination theorem for weak quantifiers. Third, we add a new characterization to the body of known characterizations of how classes like FPT can be characterized in a slicewise fashion by logical formulas.

Our Contributions I: A Syntactic Characterization of Color Coding. The hard triangle packing problem from above becomes almost trivial when we just wish to check whether a *vertex-colored* graph contains a red triangle, a green triangle, a blue triangle, a yellow triangle, and so on for k different colors. The ingenious idea behind the color coding technique of Alon, Yuster, and Zwick [1] is to reduce the original problem to the much simpler colored version by simply *randomly coloring the graph.* Of course, even if there are k disjoint triangles, we will most likely *not* color them monochromatically and differently, *but* the probability of "getting lucky" is nonzero and depends only on the parameter k. Even better, Alon et al. point out that one can *derandomize the coloring easily by using universal hash functions to color each vertex with its hash value.*

Applying this idea in the setting of descriptive complexity was recently pioneered by Chen et al. [6]. Transferred to the triangle packing problem, their argument would roughly be: "Testing for each color i whether there is a monochromatic triangle of color i can be done in first-order logic using something like $\bigwedge_{i=1}^{k} \exists x \exists y \exists z (Exy \wedge Eyz \wedge Exz \wedge C_ix \wedge C_iy \wedge C_iz)$. Next, instead of testing whether x has color i using the formula C_ix, we can test whether x gets hashed to i by a hash function. Finally, since computing appropriate universal hash functions only involves addition and multiplication, we can express the derandomized algorithm using an arithmetic first-order formula of low quantifier rank." Phrased differently, Chen et al. would argue that $\bigwedge_{i=1}^{k} \exists x \exists y \exists z (Exy \wedge Eyz \wedge Exz \wedge C_ix \wedge C_iy \wedge C_iz)$ together with the requirement that the C_i are pairwise disjoint is (ignoring some details) equivalent to $\delta_k = \exists p \exists q \bigwedge_{i=1}^{k} \exists x \exists y \exists z (Exy \wedge Eyz \wedge Exz \wedge \textsc{hash}_k(x,p,q) = i \wedge \textsc{hash}_k(y,p,q) = i \wedge \textsc{hash}_k(z,p,q) = i)$, where $\textsc{hash}_k(x,p,q) = i$ is a formula that is true when "x is hashed to i by a member of a universal family of hash functions indexed by q and p."

The family $(\delta_k)_{k\in\mathbb{N}}$ may seem rather technical and, indeed, its importance becomes visible only in conjunction with another result by Chen et al. [6]: They show that a parameterized problem lies in para-AC^0, one of the smallest "sensible" subclasses of FPT, if it can be described by a family $(\phi_k)_{k\in\mathbb{N}}$ of $\mathrm{FO}[+,\times]$ formulas of *bounded quantifier rank* such that the finite models of ϕ_k are exactly the elements of the kth slice of the problem. Since the triangle packing problem can be described in this way via the family $(\delta_k)_{k\in\mathbb{N}}$ of formulas, all of which have a quantifier rank 5 plus the constant number of quantifiers used to express the arithmetics in the formulas $\textsc{hash}_k(x,p,q)=i$, we get p-TRIANGLE-PACKING $\in$ FPT.

Clearly, this beautiful idea cannot work in all situations: If it also worked for the formula mentioned earlier expressing 3-colorability, 3-colorability would be first-order expressible, which is known to be impossible. Our first main contribution is a *syntactic characterization of when the color coding technique is applicable,* that is, of why color coding works for triangle packing but not for 3-colorability: For triangle packing, the colors C_i are applied to variables only inside *existential scopes* ("$\exists x\exists y\exists z$") while for 3-colorability the colors R, G, and B are also applied to variables inside universal scopes ("for all adjacent vertices"). In general, see Theorem 3.1 for the details, we show that a second-order quantification over an arbitrary number of disjoint colors C_i can be replaced by a fixed number of first-order quantifiers whenever none of the C_i is used in a universal scope.

Our Contributions II: New First-Order Quantifier Elimination Rules. The "purpose" of the colors C_i in the formulas $\bigwedge_{i=1}^k \exists x\exists y\exists z(Exy\wedge Eyz\wedge Exz\wedge C_ix\wedge C_iy\wedge C_iz)$ is not that the three vertices of a triangle get a particular color, but just that they get a color *different* from the color of all other triangles. Indeed, our "real" objective in these formulas is to ensure that the vertices of a triangle are *distinct* from the vertices in the other triangles – and giving vertices different colors is "just a means" of ensuring this.

In our second main contribution we explore this idea further: If the main (indeed, the only) use of colors in the context of color coding is to ensure that certain vertices are different, let us do away with colors and instead focus on the notion of *distinctness.* To better explain this idea, consider the following family, also describing triangle packing, where the only change is that we now require (a bit superfluously) that even the vertices inside a triangle get different colors: $\bigwedge_{j=1}^k \exists x\exists y\exists z(Exy\wedge Eyz\wedge Exz\wedge C_{3j-2}x\wedge C_{3j-1}y\wedge C_{3j}z)$. Observe that each C_i is now applied to exactly one variable (x, y, or z in one of the many literals) and the only "effect" that all these applications have is to ensure that the variables are different. In particular, the formula is equivalent to

$$\exists x_1\cdots\exists x_{3k}\bigwedge_{i\neq j} x_i\neq x_j\wedge\bigwedge_{j=1}^k \exists x\exists y\exists z(Exy\wedge Eyz\wedge Exz\wedge x_{3j-2}=x\wedge x_{3j-1}=y\wedge x_{3j}=z) \tag{4}$$

and these formulas are clearly equivalent to the almost identical formulas from (2).

In a sense, in (4) the many existential quantifiers $\exists x_i$ and the many $x_i\neq x_j$ literals come "for free" from the color coding technique, while $\exists x$, $\exists y$, and $\exists z$ have nothing to do with color coding. Our key observation is a syntactic property that tells us whether a quantifier comes "for free" in this way (we will call it *weak*) or not (we will call it *strong*): Definition 3.4 states (essentially) that weak quantifiers have the form $\exists x(\phi)$ such that x is not free in any universal scope of ϕ and x is used in at most one literal that is not of the form $x\neq y$. To make weak quantifiers easier to spot, we mark their bound variables with a dot (note that this is a "syntactic hint" without semantic meaning). Formulas (4) now read $\exists\dot x_1\cdots\exists\dot x_{3k}\bigwedge_{i\neq j}\dot x_i\neq\dot x_j\wedge\bigwedge_{j=1}^k\exists x\exists y\exists z(Exy\wedge Exz\wedge Eyz\wedge\dot x_{3j-2}=x\wedge\dot x_{3j-1}=y\wedge\dot x_{3j}=z)$. Observe that x, y, and z are not weak since each is used in three literals that are not inequalities.

We show in Theorem 3.5 that each ϕ is equivalent to a ϕ' whose quantifier rank depends only on the *strong quantifier rank* of ϕ (meaning that we ignore the weak quantifiers) and whose number of variables depends only on the number of strong variables in ϕ'. For instance, the formulas from (4) all have strong quantifier rank 3 and, thus, the triangle packing problem can be described by a family of constant (normal) quantifier rank. Applying Chen et al.'s characterization yields membership in para-AC^0.

As a more complex example, let us sketch a "purely syntactic" proof of the result [3, 5] that the embedding problem for graphs H of tree depth at most d lies in para-AC^0 for each d. Once more, we construct a family (ϕ_H) of formulas of constant strong quantifier rank that describes the problem. For a graph H and a rooted tree T of depth d such that H is contained in T's transitive closure (this is the definition of "H has tree depth d"), let c_1 be the root of T and let children(c) be the children of c in T. Then the following formula of strong quantifier rank d describes that H can be embedded into a structure:

$$\exists \dot{x}_1 \cdots \exists \dot{x}_{|H|} \big(\textstyle\bigwedge_{i\neq j} \dot{x}_i \neq \dot{x}_j \land \exists n_1 (n_1 = \dot{x}_{c_1} \land \bigwedge_{c_2 \in \mathrm{children}(c_1)} \exists n_2 (n_2 = \dot{x}_{c_2} \land$$
$$\textstyle\bigwedge_{c_3 \in \mathrm{children}(c_2)} \exists n_3 (n_3 = \dot{x}_{c_3} \land \bigwedge_{c_4 \in \mathrm{children}(c_3)} \exists n_4 (n_4 = \dot{x}_{c_4} \land \dots$$
$$\textstyle\bigwedge_{c_d \in \mathrm{children}(c_{d-1})} \exists n_d (n_d = \dot{x}_{c_d} \land \bigwedge_{i,j \in \{1,\dots,d\} : (c_i, c_j) \in E(H)} E n_i n_j) \dots)))\big).$$

Our Contributions III: Slicewise Descriptions and Variable Set Sizes. Our third contribution is a new result in the same vein as the already repeatedly mentioned result of Chen et al. [6]: Theorem 2.3 states that a parameterized problem can be described slicewise by a family $(\phi_k)_{k\in\mathbb{N}}$ of arithmetic first-order formulas that all use only a *bounded number of variables* if, and only if, the problem lies in para-AC$^{0\uparrow}$ – a class that has been encountered repeatedly in the literature [2, 3, 8, 14], but for which no characterization was known. It contains all parameterized problems that can be decided by AC-circuits whose depth depends only on the parameter and whose size is of the form $f(k) \cdot n^c$.

As an example, consider the problem of deciding whether a graph contains a path of length k (no vertex may be visited twice). It can be described (for odd k) by: $\exists s \exists t \exists x (Esx \land \exists \dot{x}_1 (\dot{x}_1 = x \land \exists y (Exy \land \exists \dot{x}_2 (\dot{x}_2 = y \land \exists x (Eyx \land \exists \dot{x}_3 (\dot{x}_3 = x \land \exists y (Exy \land \exists \dot{x}_4 (\dot{x}_4 = y \land \dots \land \exists x (Eyx \land x = t \land \exists \dot{x}_k (\dot{x}_k = x \land \bigwedge_{i\neq j} \dot{x}_i \neq \dot{x}_j) \dots))))$. Note that, now, the strong quantifier rank depends on k and, thus, is not constant. However, there are now only four strong variables, namely s, t, x, and y. By Theorem 3.5 we see that the above formulas are equivalent to a family of formulas with a bounded number of variables and by Theorem 2.3 we see that p-LONG-PATH $\in$ para-AC$^{0\uparrow}$ $\subseteq$ FPT. These ideas also generalize easily and we give a purely syntactic proof of the seminal result from the original color coding paper [1] that the embedding problem for graphs of bounded tree *width* lies in FPT. The core observation – which unifies the results for tree width and depth – is that for each graph with a given tree decomposition, the embedding problem can be described by a formula whose strong nesting structure mirrors the tree structure and whose strong variables mirror the bag contents.

Related Work. Flum and Grohe [10] were the first to give characterizations of FPT and of many subclasses in terms of the syntactic properties of formulas describing their members. Unfortunately, these syntactic properties do not hold for the descriptions of parameterized problems found in the literature. For instance, they show that FPT contains exactly the problems that can be described by families of FO[LFP]-formulas of bounded quantifier rank – but actually describing problems like p-VERTEX-COVER in this way is more or less hopeless and yields little insights into the structure or complexity of the problem. We believe that it is no coincidence that no applications of these beautiful characterizations to concrete problems could be found in the literature – at least prior to very recent work by Chen and Flum [7], who study slicewise descriptions of problems on structures of bounded tree depth, and the already cited article of Chen et al. [6], who *do* present a family of formulas that describe the vertex cover problem. This family internally uses the color coding technique and is thus closely related to our results. The crucial difference is, however, that we identify syntactic properties of logical formulas that imply that the color coding technique can be applied. It then suffices to find a family describing a given problem that meets the syntactic properties to establish the complexity of the problem: there is no need to actually construct the color-coding-based formulas – indeed, there is not even a need to understand how color coding works in order to decide whether a quantifier is weak or strong.

Organization of this Paper. In Section 2 we first review some of the existing work on the descriptive complexity of parameterized problems. We add to this work in the form of the mentioned characterization of the class para-$\mathrm{AC}^{0\uparrow}$ in terms of a bounded number of variables. Our main technical results are then proved in Section 3, where we establish and prove the syntactic properties that formulas must have in order for the color coding method to be applicable. In Section 4 we then apply the findings and show how membership of different natural problems in para-AC^0 and para-$\mathrm{AC}^{0\uparrow}$ (and, thus, in FPT) can be derived entirely from the syntactic structure of the formulas describing them. Full proofs can be found in the full version, but we include proof sketches in the text.

2 Describing Parameterized Problems

A happy marriage of parameterized complexity and descriptive complexity was first presented in [10]. We first review the most important definitions from [10] and then prove a new characterization, namely of the class para-$\mathrm{AC}^{0\uparrow}$ that contains all problems decidable by AC-circuits of parameter-dependent depth and “FPT-like” size. Since the results and notions will be useful later, but do not lie at the paper's heart, we keep this section brief.

Logical Terminology. We only consider first-order logic and use standard notations, with the perhaps only deviations being that we write relational atoms briefly as Exy instead of $E(x, y)$ and that the literal $x \neq y$ is an abbreviation for $\neg x = y$ (recall that a *literal* is an atom or a negated atom). Signatures, typically denoted τ, are always finite and may only contain relation symbols and constant symbols – with one exception: The special unary function symbol $\textsc{succ}$ may also be present in a signature. Let us write $\textsc{succ}^k$ for the k-fold application of $\textsc{succ}$, so $\textsc{succ}^3(x)$ is short for $\textsc{succ}(\textsc{succ}(\textsc{succ}(x)))$. It allows us to specify any fixed non-negative integer without having to use additional variables. An alternative is to dynamically add constant symbols for numbers to signatures as done in [6], but we believe that following [10] and adding the successor function gives a leaner formal framework. Let $\operatorname{arity}(\tau)$ be the maximum arity of relation symbols in τ.

We denote by $\textsc{struc}[\tau]$ the class of all τ-structures and by $|\mathcal{A}|$ the universe of $\mathcal{A}$. As is often the case in descriptive complexity theory, we only consider ordered structures in which the ternary predicates $\textsc{add}$ and $\textsc{mult}$ are available and have their natural meaning. Formally, we say τ is *arithmetic* if it contains all of the predicates $<$, $\textsc{add}$, $\textsc{mult}$, the function symbol $\textsc{succ}$, and the constant symbol 0 (it is included for convenience only). In this case, $\textsc{struc}[\tau]$ contains only those $\mathcal{A}$ for which $<^{\mathcal{A}}$ is a linear ordering of $|\mathcal{A}|$ and the other operations have their natural meaning relative to $<^{\mathcal{A}}$ (with the successor of the maximum element of the universe being itself and with 0 being the minimum with respect to $<^{\mathcal{A}}$). We write $\phi \in \mathrm{FO}[+, \times]$ when ϕ is a τ-formula for an arithmetic τ.

A τ*-problem* is a set $Q \subseteq \textsc{struc}[\tau]$ closed under isomorphisms. A τ-formula ϕ *describes* a τ-problem Q if $Q = \{\mathcal{A} \in \textsc{struc}[\tau] \mid \mathcal{A} \models \phi\}$ and it *describes* Q *eventually* if ϕ describes a set Q' that differs from Q only on structures of a certain maximum size.

▶ **Lemma 2.1.** *For each $\phi \in \mathrm{FO}[+, \times]$ that describes a τ-problem Q eventually, there are quantifier-free formulas α and β such that $(\alpha \wedge \phi) \vee \beta$ describes Q.*

Proof Sketch. Setup α to test structure size. "Hardwire" into β which "small" structures lie in Q. Use $\textsc{succ}$ to address the elements of small structures without using quantifiers. ◀

We write $\mathrm{qr}(\phi)$ for the quantifier rank of a formula and $\mathrm{bound}(\phi)$ for the set of its bound variables. For instance, for $\phi = \bigl(\exists x \exists y(Exz)\bigr) \vee \forall y(Px)$ we have $\mathrm{qr}(\phi) = 2$, since the maximum nesting is caused by the two nested existential quantifiers, and $\mathrm{bound}(\phi) = \{x, y\}$.

Let us say that ϕ is *in negation normal form* if negations are applied only to atomic formulas.

Describing Parameterized Problems. When switching from classical complexity theory to descriptive complexity theory, the basic change is that "words" get replaced by "finite structures." The same idea works for parameterized complexity theory and, following Flum and Grohe [10], let us define *parameterized problems* as subsets $Q \subseteq \textsc{struc}[\tau] \times \mathbb{N}$ where Q is closed under isomorphisms. In a pair $(\mathcal{A}, k) \in \textsc{struc}[\tau] \times \mathbb{N}$ the number k is, of course, the *parameter value* of the pair. Flum and Grohe now propose to describe such problems slicewise using formulas. Since this will be the only way in which we describe problems, we will drop the "slicewise" in the phrasings and just say that a computable family $(\phi_k)_{k \in \mathbb{N}}$ of formulas *describes* a problem $Q \subseteq \textsc{struc}[\tau] \times \mathbb{N}$ if for all $(\mathcal{A}, k) \in \textsc{struc}[\tau] \times \mathbb{N}$ we have $(\mathcal{A}, k) \in Q$ if, and only if, $\mathcal{A} \models \phi_k$. One can also define a purely logical notion of reductions between two problems Q and Q', but we will need this notion only inside the proof of Theorem 4.2 and postpone the definition till then.

For a class Φ of computable families $(\phi_k)_{k \in \mathbb{N}}$, let us write $\mathrm{X}\Phi$ for the class of all parameterized problems that are described by the members of Φ (we chose "X" to represent a "slicewise" description, which seems to be in good keeping with the usual use of X in other classes such as XP or XL). For instance, the mentioned characterization of FPT in logical terms by Flum and Grohe can be written as $\mathrm{FPT} = \mathrm{X}\{(\phi_k)_{k \in \mathbb{N}} \mid \phi_k \in \mathrm{FO}[\textsc{lfp}], \max_k \mathrm{qr}(\phi_k) < \infty\}$.

We remark that instead of describing parameterized problems using families, a more standard and at the same time more flexible way is to use reductions to model checking problems. Clearly, if a family $(\phi_k)_{k \in \mathbb{N}}$ of $\mathcal{L}$-formulas describes $Q \subseteq \textsc{struc}[\tau] \times \mathbb{N}$, then there is a very simple parameterized reduction from Q to the model checking problem p_ϕ-$\textsc{mc}(\mathcal{L})$, where the input is a pair $(\mathcal{A}, \mathrm{num}(\phi))$ and the question is whether both $\mathcal{A} \models \phi$ and $\phi \in \mathcal{L}$ hold. (The function num encodes mathematical objects like ϕ or later tuples like (ϕ, δ) as unique natural numbers.) The reduction simply maps a pair $(\mathcal{A}, k)$ to $(\mathcal{A}, \mathrm{num}(\phi_k))$. Even more interestingly, without going into any of the technical details, it is also not hard to see that as long as a reduction is sufficiently simple, the reverse implication holds, that is, we can replace a reduction to the model checking problem by a family of formulas that describe the problem. We can, thus, use whatever formalism seems more appropriate for the task at hand and – as we hope that this paper shows – it is sometimes quite natural to write down a family that describes a problem.

Parameterized Circuits. For our descriptive setting, we need to slightly adapt the definition of the circuit classes para-AC^0 and para-$\mathrm{AC}^{0\uparrow}$ from [2, 3]: Let us say that a problem $Q \subseteq \textsc{struc}[\tau] \times \mathbb{N}$ is in para-AC^0, if there is a family $(C_{n,k})_{n,k \in \mathbb{N}}$ of AC-circuits (Boolean circuits with unbounded fan-in) such that for all $(\mathcal{A}, k) \in \textsc{struc}[\tau] \times \mathbb{N}$ we have, first, $(\mathcal{A}, k) \in Q$ if, and only if, $C_{|x|,k}(x) = 1$ where x is a binary encoding of $\mathcal{A}$; second, the size of $C_{n,k}$ is at most $f(k) \cdot n^c$ for some computable function f; third, the depth of $C_{n,k}$ is bounded by a constant; and, fourth, the circuit family satisfies a DLOGTIME-uniformity condition. The class para-$\mathrm{AC}^{0\uparrow}$ is defined the same way, but the depth may be $g(k)$ for some computable g instead of only $O(1)$. The following fact and theorem show how these two circuit classes are closely related to descriptions of parameterized problems using formulas:

▶ **Fact 2.2** ([6]). $\text{para-AC}^0 = \mathrm{X}\bigl\{(\phi_k)_{k\in\mathbb{N}} \bigm| \phi_k \in \mathrm{FO}[+,\times], \max_k \mathrm{qr}(\phi_k) < \infty\bigr\}$.

▶ **Theorem 2.3.** $\text{para-AC}^{0\uparrow} = \mathrm{X}\bigl\{(\phi_k)_{k\in\mathbb{N}} \bigm| \phi_k \in \mathrm{FO}[+,\times], \max_k |\mathrm{bound}(\phi_k)| < \infty\bigr\}$.

Proof Sketch. Basically, this follows from the well-known link between circuit depth and size and the number of variables used in a formula, see for instance [15]: The *quantifier rank* of a first-order formula naturally corresponds to the *depth* of a circuit that solves the model checking problem for the formula. The *number of variables* corresponds to the *exponent of the polynomial* that bounds the size of the circuit (the paper [13] is actually entitled “$\mathrm{DSPACE}[n^k] = \mathrm{VAR}[k{+}1]$”). A simple new observation (but needed for the theorem – usually only one formula is considered) is that the *length* of the formula is linked *multiplicatively* to the size of the circuit. ◀

3 Syntactic Properties Allowing Color Coding

The color coding technique [1] is a powerful method from parameterized complexity theory for “discovering small objects” in larger structures. Recall the example from the introduction: While finding k disjoint triangles in a graph is difficult in general, it is easy when the graph is colored with k colors and the objective is to find for each color one triangle having this color. The idea behind color coding is to reduce the (hard) uncolored version to the (easy) colored version by *randomly* coloring the graph and then “hoping” that the coloring assigns a different color to each triangle. Since the triangles are “small objects,” the probability that they do, indeed, get different colors depends only on k. Even more importantly, Alon et al. noticed that we can derandomize the coloring procedure simply by coloring each vertex by its hash value with respect to a simple family of universal hash functions that only use addition and multiplication [1]. This idea is beautiful and works surprisingly well in practice [12], but using the method inside proofs can be tricky: On the one hand, we need to “keep the set sizes under control” (they must stay roughly logarithmic in size) and we “need to actually identify the small set based just on its random coloring.” Especially for more complex proofs this can lead to rather subtle arguments.

In the present section, we identify *syntactic* properties of formulas that guarantee that the color coding technique can be applied. The property is that the colors (the predicates C_i in the formulas) are not in the scope of a universal quantifier (this restriction is necessary, as the example of the formula describing 3-colorability shows).

As mentioned already in the introduction, the main “job” of the colors in proofs based on color coding is to ensure that vertices of a graph are different from other vertices. This leads us to the idea of focusing entirely on the notion of distinctness in the second half of this section. This time, there will be syntactic properties of existentially bounded first-order variables that will allow us to apply color coding to them.

3.1 Formulas With Color Predicates

In graph theory, a *coloring* of a graph can either refer to an arbitrary assignment that maps each vertex to a color or to such an assignment in which vertices connected by an edge must get different colors (sometimes called *proper* colorings). For our purposes, colorings need not be proper and are thus partitions of the vertex set into *color classes.* From the logical point of view, each color class can be represented by a unary predicate. A *k-coloring of a τ-structure* $\mathcal{A}$ is a structure $\mathcal{B}$ over the signature $\tau_{k\text{-colors}} = \tau \cup \{C_1^1, \dots, C_k^1\}$, where the C_i are fresh unary relation symbols, such that $\mathcal{A}$ is the τ-restriction of $\mathcal{B}$ and such that the sets $C_1^{\mathcal{B}}$ to $C_k^{\mathcal{B}}$ form a partition of the universe $|\mathcal{A}|$ of $\mathcal{A}$.

Let us now formulate and prove the first syntactic version of color coding. An example of a possible formula ϕ in the theorem is $\bigwedge_{i=1}^{k} \exists x \exists y \exists z(Exy \wedge Eyz \wedge Exz \wedge C_i x \wedge C_i y \wedge C_i z)$, for which the theorem tells us that there is a formula ϕ' of constant quantifier rank that is true exactly when there are pairwise disjoint sets C_i that make ϕ true.

▶ **Theorem 3.1.** *Let τ be an arithmetic signature and let k be a number. For each first-order $\tau_{k\text{-colors}}$-sentence ϕ in negation normal form in which no C_i is inside a universal scope, there is a τ-sentence ϕ' such that:*

1. *For all $\mathcal{A} \in \mathrm{STRUC}[\tau]$ we have $\mathcal{A} \models \phi'$ if, and only if, there is a k-coloring $\mathcal{B}$ of $\mathcal{A}$ with $\mathcal{B} \models \phi$.*
2. $\mathrm{qr}(\phi') = \mathrm{qr}(\phi) + O(1)$.
3. $|\mathrm{bound}(\phi')| = |\mathrm{bound}(\phi)| + O(1)$.

(Let us clarify that $O(1)$ represents a global constant that is independent of τ and k.)

Proof. Let τ, k, and ϕ be given as stated in the theorem. If necessary, we modify ϕ to ensure that there is no literal of the form $\neg C_i x_j$, by replacing each such literal by the equivalent $\bigvee_{l \neq i} C_l x_j$. After this transformation, the C_i in ϕ are neither in the scope of universal quantifiers nor of negations – and this is also true for all subformulas α of ϕ. We will now show by structural induction that all these subformulas (and, hence, also ϕ) have two *semantic* properties, which we call the *monotonicity property* and the *small witness property* (with respect to the C_i). Afterwards, we will show that these two properties allow us to apply the color coding technique.

Establishing the Monotonicity and Small Witness Properties. Some notations will be useful: Given a τ-structure $\mathcal{A}$ with universe A and given sets $A_i \subseteq A$ for $i \in \{1, \dots, k\}$, let us write $\mathcal{A} \models \phi(A_1, \dots, A_k)$ to indicate that $\mathcal{B}$ is a model of ϕ where $\mathcal{B}$ is the $\tau_{k\text{-colors}}$-structure with universe A in which all symbols from τ are interpreted as in $\mathcal{A}$ and in which the symbol C_i is interpreted as A_i, that is, $C_i^{\mathcal{B}} = A_i$. Subformulas γ of ϕ may have free variables and suppose that x_1 to x_m are the free variables in γ and let $a_i \in A$ for $i \in \{1, \dots, m\}$. We write $\mathcal{A} \models \gamma(A_1, \dots, A_k, a_1, \dots, a_m)$ to indicate that γ holds in the just-described structure $\mathcal{B}$ when each x_i is interpreted as a_i.

▶ **Definition 3.2.** *Let γ be a $\tau_{k\text{-colors}}$-formula with free variables x_1 to x_m. We say that γ has the* monotonicity and the small witness properties with respect to the C_i *if for all τ-structures $\mathcal{A}$ with universe A and all values $a_1, \dots, a_m \in A$ the following holds:*

1. Monotonicity property: *Let $A_1, \dots, A_k \subseteq A$ and $B_1, \dots, B_k \subseteq A$ be sets with $A_i \subseteq B_i$ for all $i \in \{1, \dots, k\}$. Then $\mathcal{A} \models \gamma(A_1, \dots, A_k, a_1, \dots, a_m)$ implies $\mathcal{A} \models \gamma(B_1, \dots, B_k, a_1, \dots, a_m)$.*

2. Small witness property: *If there are any pairwise disjoint sets $B_1, \dots, B_k \subseteq A$ with $\mathcal{A} \models \gamma(B_1, \dots, B_k, a_1, \dots, a_m)$, then there are sets $A_i \subseteq B_i$ whose sizes $|A_i|$ depend only on γ for $i \in \{1, \dots, k\}$, such that $\mathcal{A} \models \gamma(A_1, \dots, A_k, a_1, \dots, a_m)$.*

We now show that ϕ has these two properties (for $m = 0$). For monotonicity, just note that the C_i are not in the scope of any negation and, thus, if some A_i make ϕ true, so will all supersets B_i of the A_i.

To see that the small witness property holds, we argue by structural induction: If ϕ is any formula that does not involve any C_i, then ϕ is true or false independently of the B_i and, in particular, if it is true at all, it is also true for $A_i = \emptyset$ for $i \in \{1, \dots, k\}$. If ϕ is the atomic formula $C_i x_j$, then setting $A_i = \{a_j\}$ and $A_{i'} = \emptyset$ for $i' \neq i$ makes the formula true.

If $\phi = \alpha \wedge \beta$, then α and β have the small witness property by the induction hypothesis. Let $B_1, \dots, B_k \subseteq A$ make ϕ hold in $\mathcal{A}$. Then they also make both α and β hold in $\mathcal{A}$. Let $A_1^\alpha, \dots, A_k^\alpha \subseteq A$ with $A_i^\alpha \subseteq B_i$ be the witnesses for α and let $A_1^\beta, \dots, A_k^\beta \subseteq A$ be the witnesses for β. Then by the monotonicity property, $A_1^\alpha \cup A_1^\beta, \dots, A_k^\alpha \cup A_k^\beta$ makes both α and β true, that is

$$\mathcal{A} \models \alpha(A_1^\alpha \cup A_1^\beta, \dots, A_k^\alpha \cup A_k^\beta, a_1, \dots, a_m)$$

and the same holds for β. Note that $A_i^\alpha \cup A_i^\beta \subseteq B_i$ still holds and that they have sizes depending only on α and β and thereby on ϕ.

For $\phi = \alpha \vee \beta$ we can argue in exactly the same way as for the logical and.

The last case for the structural induction is $\phi = \exists x_m(\alpha)$. Consider pairwise disjoint $B_1, \dots, B_k \subseteq A$ that make ϕ true. Then there is a value $a_m \in A$ such that $\mathcal{A} \models \alpha(B_1, \dots, B_k, a_1, \dots, a_m)$. Now, since α has the small witness property by the induction hypothesis, we get $A_i \subseteq B_i$ of size depending on α for which we also have $\mathcal{A} \models \alpha(A_1, \dots, A_k, a_1, \dots, a_m)$. But then, by the definition of existential quantifiers, these A_i also witness $\mathcal{A} \models \exists x_m \phi(A_1, \dots, A_k, a_1, \dots, a_{m-1})$. (Observe that this is the point where the argument would *not* work for a universal quantifier: Here, for each possible value of a_m we might have a different set of A_i's as witnesses and their union would then no longer have small size.)

Applying Color Coding. Our next step in the proof is to use color coding to produce the partition. First, let us recall the basic lemma on universal hash functions formulated below in a way equivalent to [11, page 347]:

▶ **Lemma 3.3.** *There is an $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$ and all subsets $X \subseteq \{0, \dots, n-1\}$ there exist a prime $p < |X|^2 \log_2 n$ and a number $q < p$ such that the function $h_{p,q}(m) = (q \cdot m \bmod p) \bmod |X|^2$ is injective on X.*

As has already been observed by Chen et al. [6], if we set $k = |X|$ we can easily express the computation underlying $h_{p,q}\colon \{0, \dots, n-1\} \to \{0, \dots, k^2-1\}$ using a fixed $\mathrm{FO}[+, \times]$-formula $\rho(k, p, q, x, y)$. That is, if we encode the numbers $k, p, q, x, y \in \{0, \dots, n-1\}$ as corresponding elements of the universe with respect to the ordering of the universe, then $\rho(k, p, q, x, y)$ holds if, and only if, $h_{p,q}(x) = y$. Note that the p and q from the lemma could exceed n for very large X (they can reach up to $n^2 \log_2 n \leq n^3$), but, first, this situation will not arise in the following and, second, this could be fixed by using three variables to encode p and three variables to encode q. Trivially, $\rho(k, p, q, x, y)$ has some constant quantifier rank (the formula explicitly constructed by Chen et al. has $\mathrm{qr}(\rho) = 9$, assuming $k^2 < n$).

Next, we will need the basic idea or “trick” of Alon et al.'s [1] color coding technique: While for appropriate p and q the function $h_{p,q}$ will “just” be injective on $\{0, \dots, k^2-1\}$, we actually want a function that maps each element $x \in X$ to a specific element ("the color of x") of $\{1, \dots, k\}$. Fortunately, this is easy to achieve by concatenating $h_{p,q}$ with an appropriate function $g\colon \{0, \dots, k^2 - 1\} \to \{1, \dots, k\}$.

In detail, to construct ϕ' from the claim of the theorem, we construct a family of formulas $\phi^g(p, q)$ where p and q are new free variables and the formulas are indexed by all possible functions $g\colon \{0, \dots, k^2 - 1\} \to \{1, \dots, k\}$: In ϕ, replace every occurrence of $C_i x_j$ by the following formula $\pi_i^g(p, q, x_j)$:

$$\bigvee\nolimits_{y \in \{0,\dots,k^2-1\}, g(y)=i} \exists \hat{k} \exists \hat{y} \big(\textsc{succ}^k(0) = \hat{k} \land \textsc{succ}^y(0) = \hat{y} \land \rho(\hat{k}, p, q, x_j, \hat{y})\big)$$

where $\hat{k}$ and $\hat{y}$ are fresh variables that we bind to the numbers k and y (if the universe is large enough). Note that the formula $C_i x_j$ has x_j as a free variable, while $\pi_i^g(p, q, x_j)$ additionally has p and q as free variables. As an example, for the formula $\phi = \exists x(C_2 x \lor \exists y C_5 y)$ we would have $\phi^g = \exists x(\pi_2^g(p, q, x) \lor \exists y \pi_5^g(p, q, y))$. Clearly, each ϕ^g has the property $\operatorname{qr}(\phi^g) = \operatorname{qr}(\phi) + O(1)$.

The desired formula ϕ' is (almost) simply $\bigvee_{g:\{0,\dots,k^2-1\}\to\{1,\dots,k\}} \exists p \exists q(\phi^g(p, q))$. The "almost" is due to the fact that this formula works only for structures with a sufficiently large universe – but by Lemma 2.1 it suffices to consider only this case. Let us prove that for every σ-structure $\mathcal{A}$ with universe $A = \{0, \dots, n - 1\}$ and $n \geq c$ for some to-be-specified constant c, the following two statements are equivalent:

1. There is a k-coloring $\mathcal{B}$ of $\mathcal{A}$ with $\mathcal{B} \models \phi$.

2. $\mathcal{A} \models \bigvee_{g:\{0,\dots,k^2-1\}\to\{1,\dots,k\}} \exists p \exists q(\phi^g(p, q))$.

Let us start with the implication of item 2 to 1. Suppose there is a function $g\colon \{0, \dots, k^2 - 1\} \to \{1, \dots, k\}$ and elements $p, q \in \{0, \dots, n - 1\}$ such that $\mathcal{A} \models \phi^g(p, q)$. We define a partition $A_1 \mathbin{\dot\cup} \cdots \mathbin{\dot\cup} A_k = A$ by $A_i = \{x \in A \mid g(h_{p,q}(x)) = i\}$. In other words, A_i contains all elements of A that are first hashed to an element of $\{0, \dots, k^2 - 1\}$ that is then mapped to i by the function g. Trivially, the A_i form a partition of the universe A.

Assuming that the universe size is sufficiently large, namely for $k^2 \log_2 n < n$, inside ϕ^g all uses of $\rho(\hat{k}, p, q, x, \hat{y})$ will have the property that $\mathcal{A} \models \rho(\hat{k}, p, q, x, \hat{y})$ if, and only if, $h_{p,q}(x) = \hat{y}$. Clearly, there is a constant c depending only on k such that for all $n > c$ we have $k^2 \log_2 n < n$.

With the property established, we now see that $\pi_i^g(p, g, x_j)$ holds inside the formula ϕ^g if, and only if, the interpretation of x_j is an element of A_i. This means that if we interpret each C_i by A_i, then we get $\mathcal{A} \models \phi(A_1, \dots, A_k)$ and the A_i form a partition of the universe. In other words, we get item 1.

Now assume that item 1 holds, that is, there is a partition $B_1 \mathbin{\dot\cup} \cdots \mathbin{\dot\cup} B_k = A$ with $\mathcal{A} \models \phi(B_1, \dots, B_k)$. We must show that there are a $g\colon \{0, \dots, k^2 - 1\} \to \{1, \dots, k\}$ and $p, q \in A$ such that $\mathcal{A} \models \phi^g(p, q)$.

At this point, we use the small witness property that we established earlier for the partition. By this property there are pairwise disjoint sets $A_i \subseteq A$ such that, first, $|A_i|$ depends only on ϕ and, second, $\mathcal{A} \models \phi(A_1, \dots, A_k)$. Let $X = \bigcup_{i=1}^k A_i$. Then $|X|$ depends only on ϕ and let s_ϕ be a ϕ-dependent upper bound on this size. By the universal hashing lemma, there are now p and q such that $h_{p,q}\colon \{0, \dots, n - 1\} \to \{0, \dots, s_\phi^2 - 1\}$ is injective on X. But, then, we can set $g\colon \{0, \dots, s_\phi^2 - 1\} \to \{1, \dots, k\}$ to $g(v) = i$ if there is an $x \in A_i$ with $h_{p,q}(x) = v$ and setting $g(v)$ arbitrarily otherwise. Note that this is, indeed, a valid definition of g since $h_{p,q}$ is injective on X.

With these definition, we now define the following sets D_1 to D_k: Let $D_i = \{x \in A \mid g(h_{p,q}(\hat{x})) = i\}$ where $\hat{x}$ is the index of x in A with respect to the ordering (that is, $\hat{x} = |\{y \in A \mid y <^{\mathcal{A}} x\}|$ and for the special case that $A = \{0, \dots, n - 1\}$ and that $<^{\mathcal{A}}$ is the natural ordering, $\hat{x} = x$). Observe that $D_i \supseteq A_i$ holds for all D_i and that the D_i form a partition of the universe A. By the monotonicity property, $\mathcal{A} \models \phi(A_1, \dots, A_k)$ implies $\mathcal{A} \models \phi(D_1, \dots, D_k)$. However, by definition of the D_i and of the formulas π_i^g, for a sufficiently large universe size n (namely $s_\phi^2 \log_2 n < n$), we now also have $\mathcal{A} \models \phi^g(p, q)$, which in turn implies $\mathcal{A} \models \bigvee_g \exists p \exists q \phi^g$. ◀

In the theorem we assumed that ϕ is a sentence to keep the notation simple, both the theorem and later theorems still hold when $\phi(x_1, \dots, x_n)$ has free variables x_1 to x_n. Then there is a corresponding $\phi'(x_1, \dots, x_n)$ such that first item becomes that for all $\mathcal{A} \in \text{STRUC}[\tau]$ and all $a_1, \dots, a_n \in |\mathcal{A}|$ we have $\mathcal{A} \models \phi'(a_1, \dots, a_n)$ if, and only if, there is a k-coloring $\mathcal{B}$ of $\mathcal{A}$ with $\mathcal{B} \models \phi(a_1, \dots, a_n)$. Note that the syntactic transformations in the theorem do not add dependencies of universal quantifiers on the free variables.

3.2 Formulas With Weak Quantifiers

If one has a closer look at proofs based on color coding, one cannot help but notice that the colors are almost exclusively used to ensure that certain vertices in a structure are distinct from certain other vertices: recall the introductory example $\bigwedge_{j=1}^{k} \exists x \exists y \exists z (Exy \wedge Eyz \wedge Exz \wedge C_{3j-2}x \wedge C_{3j-1}y \wedge C_{3j}z)$, which describes the triangle packing problem when we require that the C_i form a partition of the universe. Since the C_i are only used to ensure that the many different x, y, and z are different, we already rewrote the formula in (4) as $\exists x_1 \cdots \exists x_{3k} \bigwedge_{i \neq j} x_i \neq x_j \wedge \bigwedge_{j=1}^{k} \exists x \exists y \exists z (Exy \wedge Eyz \wedge Exz \wedge x_{3j-2} = x \wedge x_{3j-1} = y \wedge x_{3j} = z)$. While this rewriting gets rid of the colors and moves us back into the familiar territory of simple first-order formulas, the quantifier rank and the number of variables in the formula have now "exploded" (from the constant 3 to the parameter-dependent value $3k+3$) – which is exactly what we need to avoid in order to apply Fact 2.2 or Theorem 2.3.

We now define a syntactic property that the x_i have that allows us to remove them from the formula and, thereby, to arrive at a family of formulas of constant quantifier rank. For a (sub)formula α of the form $\forall d(\phi)$ or $\exists d(\phi)$, we say that d *depends* on all free variables in ϕ (at the position of α in a larger formula). For instance, in $Exy \wedge \forall b(Exb \wedge \exists z(Eyz)) \wedge \exists b(Exx)$, the variable b depends on x and y at its first binding ($\forall b$) and on x at the second binding ($\exists b$).

▶ **Definition 3.4.** *We call the leading quantifier in a formula $\exists x(\phi)$ in negation normal form* strong *if*

1. *some universal binding inside ϕ depends on x or*
2. *there is a subformula $\alpha \wedge \beta$ of ϕ such that both α and β contain x in literals that are not of the form $x \neq y$ for some variable y.*

If neither of the above hold, we call the quantifier weak. *The* strong quantifier rank strong-qr(ϕ) *is the quantifier rank of ϕ, where weak quantifiers are ignored;* strong-bound(ϕ) *contains all variables of ϕ that are bound by non-weak quantifiers.*

(Later on we extend the definition to the dual notion of weak *universal* quantifiers, but for the moment let us only call existential quantifiers weak.)

We place a dot on the variables bound by weak quantifiers to make them easier to spot. For example, in $\phi = \exists x \exists y \exists \dot{z}(Rxx\dot{z}\dot{z} \wedge x \neq y \wedge y \neq \dot{z} \wedge Px \wedge \forall w\, Ewyy)$ the quantifier $\exists \dot{z}$ is weak, but neither are $\exists x$ (since x is used in two literals joined by a conjunction, namely in $Rxx\dot{z}\dot{z}$ and Px) nor $\exists y$ (since w depends on y in $\forall w\, Ewyy$). We have qr$(\phi) = 4$, but strong-qr$(\phi) = 3$, and bound$(\phi) = \{x, y, \dot{z}\}$, but strong-bound$(\phi) = \{x, y\}$.

Admittedly, the definition of weakness is a bit technical, but note that there is a rather simple sufficient condition for a variable x to be weak: If it not used in universal binding and used in only one literal that is not an inequality, then x is weak. This condition almost always suffices for identifying the weak variables, although there are of course exceptions like $\exists \dot{x}(P\dot{x} \lor Q\dot{x})$.

▶ **Theorem 3.5.** *Let τ be an arithmetic signature. Then for every τ-formula ϕ in negation normal form there is a τ-formula ϕ' such that*

1. *ϕ' is equivalent to ϕ on finite structures,*
2. $\mathrm{qr}(\phi') = 3 \cdot \text{strong-qr}(\phi) + O(\mathrm{arity}(\tau))$, *and*
3. $|\mathrm{bound}(\phi')| = |\text{strong-bound}(\phi)| + O(\mathrm{arity}(\tau))$.

Proof Sketch. Using simple transformations, we can ensure that all weak quantifiers follow in blocks after universal quantifiers. We can also ensure that inequality literals directly follow the blocks of weak quantifiers and are joined by conjunctions. If the inequality literals following a block happen to require that all weak variables from the block are different (that is, if for all pairs $\dot{x}_i$ and $\dot{x}_j$ of different weak variables there is an inequality $\dot{x}_i \neq \dot{x}_j$), then we can remove the weak quantifiers $\exists \dot{x}_i$ and at the (single) place where $\dot{x}_i$ is used, we use a color C_i instead. For instance, if $\dot{x}_i$ is used in the literal $\dot{x}_i = y$, we replace the literal by $C_i y$. If $\dot{x}_i$ is used for instance in $\neg E\dot{x}_i y$, we replace this by $\exists x(C_i x \land \neg Exy)$. In this way, for each block we get an equivalent formula to which we can apply Theorem 3.1. A more complicated situation arises when the inequality literals in a block "do not require complete distinctness," but this case can also be handled by considering all possible ways in which the inequalities can be satisfied in parallel. In result, all weak quantifiers get removed and for each block a constant number of new quantifiers are introduced. Since each block follows a different universal quantifier, the new total quantifier rank is at most the strong quantifier rank times a constant factor; and the new number of variables is only a constant above the number of original strong variables. ◀

We already mentioned that the notion of weak existential quantifiers begs a dual: By Theorem 3.5, for $\phi = \exists \dot{x}_1 \cdots \exists \dot{x}_k(\psi)$ there is an equivalent formula ϕ' with $\mathrm{qr}(\phi') = O(\text{strong-qr}(\phi))$. Since, trivially, $\mathrm{qr}(\neg\phi') = \mathrm{qr}(\phi')$, the formula $\neg\phi$ is also equivalent to a formula of quantifier rank $O(\text{strong-qr}(\phi))$. The normal form of $\neg\phi$ starts with $\forall x_1 \cdots \forall x_k$ to which Theorem 3.5 does not apply "at all" – but the dual of the theorem applies, where we call the leading quantifier in a (sub)formula $\forall x(\phi)$ *weak* if no *existential* binding inside ϕ depends on x and in all subformulas of ϕ of the form $\alpha \lor \beta$ at most one of α and β may contain a literal that contains x and is not of the form $x = y$ (note that this is now an equality). More interestingly, we can even show that both kinds of weak quantifiers may be present:

▶ **Theorem 3.6.** *Theorem 3.5 still holds when ϕ may contain both existential and universal weak variables, none of which count towards the strong quantifier rank nor count as strong bound variables.*

Proof Sketch. As in the proof of Theorem 3.5, we syntactically transform ϕ so that the weak existential quantifiers follow strong universal quantifiers in block and – this is new – that the weak universal quantifiers follow strong existential quantifiers. The key observation that makes these transformations possible in the mixed case is that weak existential and weak universal quantifiers commute: For instance, $\exists \dot{x}(\alpha \land \forall \dot{y}(\beta)) \equiv \forall \dot{y}(\beta \land \exists \dot{x}(\alpha))$ since $\dot{x}$ and $\dot{y}$ cannot depend on one another by the core property of weak quantifiers (α cannot contain $\dot{y}$ and β cannot contain $\dot{x}$). ◀

4 Syntactic Proofs and Natural Problems

The special allure of descriptive complexity theory lies in the possibility of proving that a problem has a certain complexity just by describing the problem in the right way. The "right way" is, of course, a logical description that has a certain syntax (such as having a bounded strong quantifier rank). In the following we present such descriptions for several natural problems and thereby bound their complexity "in a purely syntactic way." First, however, we present "syntactic tools" for describing problems more easily. These tools are built on top of the notion of strong and weak quantifiers.

4.1 Syntactic Tools: New Operators

It is common in mathematical logic to distinguish between the core syntax and additional "shorthands" built on top of the core syntax. For instance, while $\neg$ and $\vee$ are typically considered to be part of the core syntax of propositional logic, the notation $a \to b$ is often seen as a shorthand for $\neg a \vee b$. In a similar way, we now consider the notions of weak variables and quantifiers introduced in the previous section as our "core syntax" and build a number of useful shorthands on top of them. Of course, just as $a \to b$ has an intended semantic meaning that the expansion $\neg a \vee b$ of the shorthand must reflect, the shorthands we introduce also have an intended semantic meaning, which we specify.

As a first example, consider the common notation $\exists^{\geq k} x(\phi(x))$, whose intended semantics is "there are at least k different elements in the universe that make $\phi(x)$ true." While this notation is often considered as a shorthand for $\exists x_1 \cdots \exists x_k \bigwedge_{i \neq j} x_i \neq x_j \wedge \bigwedge_{i=1}^{k} \phi(x_i)$ we will consider it a shorthand for the equivalent, but slightly more complicated formula $\exists \dot{x}_1 \cdots \exists \dot{x}_k \bigwedge_{i \neq j} \dot{x}_i \neq \dot{x}_j \wedge \bigwedge_{i=1}^{k} \exists x(x = \dot{x}_i \wedge \phi(x))$. The difference is, of course, that the strong quantifier rank is now much lower and, hence, by Theorem 3.5 we can replace any occurrence of $\exists^{\geq k} x(\phi(x))$ by a formula of quantifier rank $\operatorname{qr}(\phi) + O(1)$. In all of the following notations, k and s are arbitrary values. The indicated strong quantifier rank for the notation is that of its expansion. The *semantics* describe which structures $\mathcal{A}$ are models of the formula.

▶ **Notation** ($\exists^{\geq k} x(\phi(x))$). **Strong-qr:** $1 + \text{strong-qr}(\phi)$
Semantics There are k distinct $a_1, \dots, a_k \in |\mathcal{A}|$ with $\mathcal{A} \models \phi(a_i)$ for all i.
Expansion $\exists \dot{x}_1 \cdots \exists \dot{x}_k \bigwedge_{i \neq j} \dot{x}_i \neq \dot{x}_j \wedge \bigwedge_{i=1}^{k} \exists x(x = \dot{x}_i \wedge \phi(x))$

▶ **Notation** ($\exists^{\leq k} x(\phi(x))$). **Strong-qr:** $1 + \text{strong-qr}(\phi)$
Semantics There are at most k distinct $a_1, \dots, a_k \in |\mathcal{A}|$ with $\mathcal{A} \models \phi(a_i)$ for all i.
Expansion $\forall \dot{x}_1 \cdots \forall \dot{x}_{k+1} \bigvee_{i \neq j} \dot{x}_i = \dot{x}_j \vee \bigvee_{i=1}^{k+1} \forall x(x \neq \dot{x}_i \vee \neg\phi(x))$ $(\equiv \neg\exists^{\geq k+1} x(\phi(x)))$

▶ **Notation** ($\exists^{= k} x(\phi(x))$). **Strong-qr:** $1 + \text{strong-qr}(\phi)$
Semantics There are exactly k distinct $a_1, \dots, a_k \in |\mathcal{A}|$ with $\mathcal{A} \models \phi(a_i)$ for all i.
Expansion $\exists^{\geq k} x(\phi(x)) \wedge \exists^{\leq k} x(\phi(x))$

The next notation is useful for "binding" a set of vertices to weak or strong variables. The binding contains the allowed "single use" of the weak variables in the sense of Definition 3.4, but they can still be used in inequality literals. Let $\mathring{x}$ indicate that x may be weak or strong.

▶ **Notation** ($\{\mathring{x}_1, \dots, \mathring{x}_k\} = \{x \mid \phi(x)\}$). **Strong-qr:** $1 + \text{strong-qr}(\phi)$
Semantics Let $a_1, \dots, a_k \in |\mathcal{A}|$ be the assignments to the $\mathring{x}_i$ (note that they need *not* be distinct). Then $\{a_1, \dots, a_k\} = \bigl\{a \in |\mathcal{A}| \bigm| \mathcal{A} \models \phi(a)\bigr\}$ must hold.
Expansion $\bigwedge_{i=1}^{k} \exists x\bigl(x = \mathring{x}_i \wedge \phi(x)\bigr) \wedge$ *// ensure* $\{\mathring{x}_1, \dots, \mathring{x}_k\} \subseteq \{x \mid \phi(x)\}$
$\bigvee_{s=1}^{k} \bigl(\exists^{=s} x(\phi(x)) \wedge$ *// bind* s *to* $|\{x \mid \phi(x)\}|$
$\bigvee_{I \subseteq \{1,\dots,k\}, |I| = s} \bigwedge_{i,j \in I, i \neq j} \mathring{x}_i \neq \mathring{x}_j\bigr).$ *// ensure* $|\{\mathring{x}_1, \dots, \mathring{x}_k\}| \geq s$

The final notation can be thought of as a "generalization of $\exists^{=k}$" where we not only ask whether there are exactly k distinct a_i with a property ϕ, but whether these a_i then also have an *arbitrary* special additional property. Formally, let $Q \subseteq \textsc{struc}[\tau]$ be an arbitrary τ-problem. We write $\mathcal{A}[I]$ for the substructure of $\mathcal{A}$ induced on a subset $I \subseteq |\mathcal{A}|$.

▶ **Notation** ($\textsc{induced}^{\text{size}=k}\{x \mid \phi(x)\} \in Q$). **Strong-qr:** $1 + \text{strong-qr}(\phi) + \text{arity}(\tau)$
Semantics The set $I = \{a \in |\mathcal{A}| \mid \mathcal{A} \models \phi(a)\}$ has size exactly k and $\mathcal{A}[I] \in Q$.
Expansion Assuming for simplicity that τ contains only E^2 as non-arithmetic predicate:

$$\exists^{=k}x(\phi(x)) \wedge \bigvee_{\mathcal{A}\in Q, |\mathcal{A}|=\{1,\dots,k\}} \bigwedge_{(i,j)\in E^{\mathcal{A}}} \exists x \exists y(\pi_i(x) \wedge \pi_j(y) \wedge Exy) \wedge \\ \bigwedge_{(i,j)\notin E^{\mathcal{A}}} \exists x \exists y(\pi_i(x) \wedge \pi_j(y) \wedge \neg Exy),$$

where $\pi_i(x)$ is a shorthand for $\phi(x) \wedge \exists^{=i-1}z(z < x \wedge \phi(z))$, which binds x to the ith element of the universe with property ϕ.

▶ **Notation** ($\textsc{induced}^{\text{size}\le k}\{x \mid \phi(x)\} \in Q$). **Strong-qr:** $1 + \text{strong-qr}(\phi) + \text{arity}(\tau)$
Semantics The set $I = \{a \in |\mathcal{A}| \mid \mathcal{A} \models \phi(a)\}$ has size at most k and $\mathcal{A}[I] \in Q$.
Expansion $\bigvee_{s=0}^{k} \textsc{induced}^{\text{size}=s}\{x \mid \phi(x)\} \in Q$

4.2 Describing Classical Problems

A *vertex cover* of a graph $G = (V, E)$ is a subset $X \subseteq V$ with $e \cap X \neq \emptyset$ for all $e \in E$. The problem p_k-$\textsc{vertex-set}$ asks whether a graph has a cover X with $|X| \le k$.

▶ **Theorem 4.1** ([2, 6]). *p-$\textsc{vertex-cover}$ $\in$ para-AC^0.*

Proof. We describe the problem using a family $(\phi_k)_{k\in\mathbb{N}}$ of constant strong quantifier rank that expresses the well-known Buss kernelization "using logic": Let $\textsc{high}(x) = \exists^{\ge k+1}y(Exy)$ expresses that x is a *high-degree vertex.* Buss observed that all high-degree vertices must be part of a vertex cover of size at most k. Thus, $h \le k$ must hold for the unique h with $\exists^{=h}x(\textsc{high}(x))$. A remaining vertex is *interesting* if it is connected to at least one non-high-degree vertex: $\textsc{interesting}(x) = \neg\,\textsc{high}(x) \wedge \exists y(Exy \wedge \neg\,\textsc{high}(y))$. If there are more than $(k-h)(k+1) \le k^2 + k$ interesting vertices, there cannot be a vertex cover – and if there are less, the graph induced on the interesting vertices must have a vertex cover of size $k - h$. In symbols: $\phi_k = \bigvee_{h=0}^{k}\bigl(\exists^{=h}x(\textsc{high}(x)) \wedge \textsc{induced}^{\text{size}\le k^2+k}\{x \mid \textsc{interesting}(x)\} \in Q_{k-h}\bigr)$ for $Q_s = \{\mathcal{G} \mid \mathcal{G} \text{ has a vertex cover of size } s\}$. ◀

Hitting sets generalize the notion of vertex covers to hypergraphs (V, E). They are still sets $X \subseteq V$ with $e \cap X \neq \emptyset$ for all $e \in E$. The problem $\text{p}_{k,d}$-$\textsc{hitting-set}$ asks whether a hypergraph with $\max_{e\in E}|e| \le d$ has a hitting set X with $|X| \le k$. Note that p-$\textsc{vertex-cover}$ is exactly this problem restricted to $d = 2$.

▶ **Theorem 4.2** ([4]). *$\text{p}_{k,d}$-$\textsc{hitting-set}$ $\in$ para-AC^0.*

Proof Sketch. Instead of the Buss kernelization, "using logic" we describe the kernelization presented by us in [4] for the hitting set problem. While this kernelization is *considerably* more complex, it turns out that it can be expressed quite naturally using weak variables. ◀

Next, we show that the result by Flum and Grohe [10] that the model checking problem for first-order logic lies in FPT on structures whose Gaifman graph has bounded degree can be obtained "syntactically." For simplicity, we only consider graphs and let $\text{p}_{\psi,\delta}$-$\textsc{mc}(\text{FO}) = \bigl\{(\mathcal{G}, \text{num}(\psi,\delta)) \bigm| \mathcal{G} \in \textsc{struc}[(E^2)], \psi \in \text{FO}, \mathcal{G} \models \psi, \text{max-degree}(\mathcal{G}) \le \delta\bigr\}$.

▶ **Theorem 4.3** ([2, 10]). $\mathrm{p}_{\psi,\delta}\text{-}\textsc{mc}(\mathrm{FO}) \in \text{para-AC}^{0\uparrow}$.

Proof Sketch. There is a family $(\phi_{\psi,\delta})_{\psi\in\mathrm{FO},\delta\in\mathbb{N}}$ with a bound on the number of strong variables that describes $\mathrm{p}_{\psi,\delta}\text{-}\textsc{mc}(\mathrm{FO})$: For fixed ψ and δ, using Gaifman's Theorem, rewrite ψ as $\exists x_1 \cdots \exists x_k \bigl(\bigwedge_{i\neq j} \gamma_{\mathrm{dist}(x_i,x_j)>2r} \wedge \bigwedge_i \rho(x_i)\bigr)$ where ρ is r-local. Because of the bounded degree, a ball of radius r can have maximum size δ^r. We can now verify the disjointness of the balls surrounding the x_i by using one weak variable for each element in such a ball. The second part can then be verified by $\textsc{induced}^{\mathrm{size}\le\delta^r}\{x \mid \gamma_{\mathrm{dist}(x,x_i)\le r}\} \in \{\mathcal{G} \mid \mathcal{G} \models \rho(x_i)\}$. ◀

In our final example, $\mathrm{td}(H)$ is the *tree depth* of H and $\mathrm{tw}(H)$ is the *tree width* (see the appendix for detailed definitions). A graph $H = (V(H), E(H))$ *embeds into* a graph $G = (V(G), E(G))$ if there is an injective mapping $\iota\colon V(H) \to V(G)$ such that for all $(u, v) \in E(H)$ we have $(\iota(u), \iota(v)) \in E(G)$. Let $\text{p-}\textsc{emb}_{\mathrm{td}\le c}$ be $\{(G, \mathrm{num}(H)) \mid \mathrm{td}(H) \le c$ and H embeds into $G\}$ and define $\text{p-}\textsc{emb}_{\mathrm{tw}\le c}$ similarly.

▶ **Theorem 4.4** ([2, 5]). $\text{p-}\textsc{emb}_{\mathrm{td}\le c} \in \text{para-AC}^0$ *and* $\text{p-}\textsc{emb}_{\mathrm{tw}\le c} \in \text{para-AC}^{0\uparrow}$ *for each* c.

Proof Sketch. For each graph H together with a tree decomposition (T, B) of H, we present a formula $\phi_{H,T,B}$ with

1. $\text{strong-qr}(\phi_{H,T,B}) = O(\mathrm{depth}(T))$ and
2. $|\text{strong-bound}(\phi_{H,T,B})| = O(\mathrm{width}(B))$,

such that for all graphs $\mathcal{G}$ we have $\mathcal{G} \models \phi_{H,T,B}$ if, and only if, H embeds into $\mathcal{G}$. The idea is to use $|H|$ distinct weak variables to bind the embedding and $\mathrm{width}(B) + 1$ strong variables to keep track of the vertices in the bags. Each time a vertex enters the bags for the first time, bind the corresponding weak variable to one of the strong ones. Recycle strong variables when a vertex leaves a bag. Build a nested formula whose structure mirrors the tree decomposition and check for each bag whether the necessary edges are present. ◀

5 Conclusion

In the present paper, we showed how the color coding technique can be turned into a powerful tool for parameterized descriptive complexity theory. This tool allows us to show that important results from parameterized complexity theory – like the fact that the embedding problem for graphs of bounded tree width lies in FPT – follow just from the syntactic structure of the formulas that describe the problem.

In all our syntactic characterizations it was important that variables or color predicates were not allowed to be within a universal scope. The reason was that literals, disjunctions, conjunctions, and existential quantifiers all have what we called the *small witness property,* which universal quantifiers do *not* have. However, there are other quantifiers, from more powerful logics that we did not explore, that also have the small witness property. An example are operators that test whether there is a path of length at most k from one vertex to another for some fixed k: if such a path exists, its vertices form a "small witness." Weak variables may be used inside these operators, leading to broader classes of problems that can be described by families of bounded strong quantifier rank. On the other hand, we cannot add the full transitive closure operator $\textsc{tc}$ (for which it is well-known that $\mathrm{FO}[\textsc{tc}] = \mathrm{NL}$) and hope that Theorems 3.1 and 3.5 still hold: If this were the case, we should be able to turn a formula that uses two colors C_1 and C_2 to express that there are two vertex-disjoint paths between two vertices into a $\mathrm{FO}[\textsc{tc}]$ formula – thus proving the unlikely result that the NP-hard disjoint path problem is in NL.

Another line of inquiry into the descriptive complexity of parameterized problems was already started in the repeatedly cited paper by Chen et al. [6]: They give first syntactic properties for families of formulas describing weighted model checking problems that imply membership in para-AC^0. We believe that it might be possible to base an alternative notion of weak quantifiers on these syntactic properties. Ideally, we would like to prove a theorem similar to Theorem 3.5 in which there are just more quantifiers that count as weak and, hence, even more families have bounded strong quantifier rank. This would allow us to prove for even more problems that they lie in FPT just because of the syntactic structure of the natural formula families that describe them.

References

1 Noga Alon, Raphael Yuster, and Uri Zwick. Color-Coding. *Journal of the ACM*, 42(4):844–856, 1995. `doi:10.1145/210332.210337`.

2 Max Bannach, Christoph Stockhusen, and Till Tantau. Fast Parallel Fixed-parameter Algorithms via Color Coding. In *Proceedings of the Tenth International Symposium on Parameterized and Exact Computation (IPEC 2015)*, pages 224–235, 2015. `doi:10.4230/LIPIcs.IPEC.2015.224`.

3 Max Bannach and Till Tantau. Parallel Multivariate Meta-Theorems. In *Proceedings of the Eleventh International Symposium on Parameterized and Exact Computation (IPEC 2016)*, pages 4:1–4:17, 2016. `doi:10.4230/LIPIcs.IPEC.2016.4`.

4 Max Bannach and Till Tantau. Computing Hitting Set Kernels By AC^0-Circuits. In *Proceedings of the 35th Symposium on Theoretical Aspects of Computer Science (STACS 2018)*, pages 9:1–9:14, 2018. `doi:10.4230/LIPIcs.STACS.2018.9`.

5 Hubie Chen and Moritz Müller. The Fine Classification of Conjunctive Queries and Parameterized Logarithmic Space. *ACM Transactions on Computation Theory*, 7(2):7:1–7:27, 2015. `doi:10.1145/2751316`.

6 Yijia Chen, Jörg Flum, and Xuangui Huang. Slicewise Definability in First-Order Logic with Bounded Quantifier Rank. In *Proceedings of the 26th EACSL Annual Conference on Computer Science Logic (CSL 2017)*, pages 19:1–19:16, 2017. `doi:10.4230/LIPIcs.CSL.2017.19`.

7 Yijia Chen and Jörg Flum. Tree-depth, quantifier elimination, and quantifier rank. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2018)*, pages 225–234. ACM, 2018. `doi:10.1145/3209108.3209160`.

8 Bireswar Das, Murali Krishna Enduri, and I. Vinod Reddy. On the Parallel Parameterized Complexity of the Graph Isomorphism Problem. In *Proceedings of the Twelfth International Conference and Workshop on Algorithms and Computation (WALCOM 2018)*, pages 252–264. Springer, 2018. `doi:10.1007/978-3-319-75172-6_22`.

9 Ronald Fagin. Generalized First-Order Spectra and Polynomial-Time Recognizable Sets. *Complexity of Computation*, 7:43–74, 1974.

10 Jörg Flum and Martin Grohe. Describing Parameterized Complexity Classes. *Information and Computation*, 187(2):291–319, December 2003. `doi:10.1016/S0890-5401(03)00161-5`.

11 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. Springer, 2006. `doi:10.1007/3-540-29953-X`.

12 Falk Hüffner, Sebastian Wernicke, and Thomas Zichner. Algorithm Engineering for Color-Coding with Applications to Signaling Pathway Detection. *Algorithmica*, 52(2):114–132, 2008. `doi:10.1007/s00453-007-9008-7`.

13 Neil Immerman. $\mathrm{DSPACE}[n^k] = \mathrm{VAR}[k+1]$. In *Proceedings of the Sixth Annual Structure in Complexity Theory Conference*, pages 334–340, 1991. `doi:10.1109/SCT.1991.160278`.

14 Michał Pilipczuk, Sebastian Siebertz, and Szymon Toruńczyk. Parameterized circuit complexity of model-checking on sparse structures. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2018)*, pages 789–798, 2018. `doi:10.1145/3209108.3209136`.

15 Heribert Vollmer. *Introduction to Circuit Complexity – A Uniform Approach*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 1999. `doi:10.1007/978-3-662-03927-4`.