Compressing Permutation Groups into Grammars and Polytopes. A Graph Embedding Approach

Lars Jaffke
University of Bergen, Norway
lars.jaffke@uib.no

Mateus de Oliveira Oliveira
University of Bergen, Norway
mateus.oliveira@uib.no

Hans Raj Tiwary
Charles University, Prague, Czech Republic
hansraj@kam.mff.cuni.cz

Abstract

It can be shown that each permutation group $G \sqsubseteq \mathbb{S}_n$ can be embedded, in a well defined sense, in a *connected* graph with $O(n + |G|)$ vertices. Some groups, however, require much fewer vertices. For instance, $\mathbb{S}_n$ itself can be embedded in the n-clique K_n, a connected graph with n vertices.

In this work, we show that the minimum size of a context-free grammar generating a finite permutation group $G \sqsubseteq \mathbb{S}_n$ can be upper bounded by three structural parameters of *connected* graphs embedding G: the number of vertices, the treewidth, and the maximum degree. More precisely, we show that any permutation group $G \sqsubseteq \mathbb{S}_n$ that can be embedded into a connected graph with m vertices, treewidth k, and maximum degree Δ, can also be generated by a context-free grammar of size $2^{O(k\Delta \log \Delta)} \cdot m^{O(k)}$. By combining our upper bound with a connection established by Pesant, Quimper, Rousseau and Sellmann [33] between the extension complexity of a permutation group and the grammar complexity of a formal language, we also get that these permutation groups can be represented by polytopes of extension complexity $2^{O(k\Delta \log \Delta)} \cdot m^{O(k)}$.

The above upper bounds can be used to provide trade-offs between the index of permutation groups, and the number of vertices, treewidth and maximum degree of connected graphs embedding these groups. In particular, by combining our main result with a celebrated $2^{\Omega(n)}$ lower bound on the grammar complexity of the symmetric group $\mathbb{S}_n$ due to Glaister and Shallit [22] we have that connected graphs of treewidth $o(n/\log n)$ and maximum degree $o(n/\log n)$ embedding subgroups of $\mathbb{S}_n$ of index 2^{cn} for some small constant c must have $n^{\omega(1)}$ vertices. This lower bound can be improved to exponential on graphs of treewidth n^{ε} for $\varepsilon < 1$ and maximum degree $o(n/\log n)$.

2012 ACM Subject Classification Theory of computation → Algebraic language theory

Keywords and phrases Permutation Groups, Context Free Grammars, Extension Complexity, Graph Embedding Complexity

Digital Object Identifier 10.4230/LIPIcs.MFCS.2020.50

Related Version A full version of the paper is available at `https://arxiv.org/abs/2001.05583`.

Funding *Lars Jaffke*: Trond Mohn Foundation (TMS).
Mateus de Oliveira Oliveira: Research Council of Norway (288761) and Trond Mohn Foundation (TMS).
Hans Raj Tiwary: Grant GAČR 17-09142S.

Acknowledgements We thank Manuel Aprile, Laszlo Babai, Peter Cameron, Michael Fellows and Samuel Fiorini for valuable comments and suggestions. We thank Michel Goemans, Kanstantsin Pashkovich and Stefan Weltge for answering some of our questions by email.

1 Introduction

Let $\mathbb{S}_n$ be the set of permutations of the set $\{1, ..., n\}$ and $\text{str}(\mathbb{S}_n)$ be the set of strings in $\{1, ..., n\}^n$ encoding permutations in $\mathbb{S}_n$. The search for minimum size grammars generating the language $\text{str}(\mathbb{S}_n)$ has sparked a lot of interest in the automata theory and in the complexity theory communities, both in the study of lower bounds [18, 31, 20], and in the study of upper bounds [25, 3, 2]. In particular, a celebrated result due to Ellul, Krawetz and Shallit [18] states that any context-free grammar generating the language $\text{str}(\mathbb{S}_n)$ must have size $2^{\Omega(n)}$. In this work, we complement this line of research by showing that the minimum size of a context-free grammar representing a finite permutation group $G \sqsubseteq \mathbb{S}_n$ can be upper bounded by three structural parameters of connected graphs whose automorphism group embed G: number of vertices, treewidth and maximum degree.

We say that a permutation group $G \sqsubseteq \mathbb{S}_n$ can be embedded in a graph X with vertex set $[m] = \{1, ..., m\}$, if $m \geq n$ and G is equal to the restriction of the automorphism group of X to its first n vertices $[n] = \{1, ..., n\}$. A more precise definition of the notion of graph embedding is given in Section 3. For a given class of *connected* graphs $\mathcal{X}$, the $\mathcal{X}$-embedding complexity of G, denoted by $\text{gec}_{\mathcal{X}}(G)$, is defined as the minimum m such that G can be embedded in an m-vertex graph $X \in \mathcal{X}$.

Given an alphabet Σ, the *symmetric grammar complexity* (SGC) of a formal language $L \subseteq \Sigma^n$ measures the minimum size of a context-free grammar accepting a permuted version of L. As a matter of comparison, we note that languages accepted by online Turing machines working in space s and with access to a stack have symmetric grammar complexity $2^{O(s)}$ [24]. In this setting, the machine reads the input string $w \in \Sigma^n$ from left to right, one symbol at a time. While reading this string, symbols can be pushed into or popped from the stack. The transitions relation depends on the current state, on the symbol being read at the input, and on the symbol being read at the top of the stack. The caveat is that the number of symbols used in the stack (which can be up to n) is not counted in the space bound s, which can be much smaller than n (say $s = O(\log n)$). The SGC of a language $L \subseteq \Sigma^n$ is also polynomially related to the minimum size of a read-once branching program with a stack accepting L (see for instance [32]).

Our Results. We show that the automorphism group of any graph with n vertices, maximum degree Δ and treewidth k has symmetric grammar complexity at most $2^{O(k\Delta \log \Delta)} \cdot n^{O(k)}$ (Theorem 3). More generally, we show that the SGC of groups that can be embedded in m-vertex graphs of maximum degree Δ and treewidth k is at most $2^{O(k\Delta \log \Delta)} \cdot m^{O(k)}$ (Theorem 5).

In linear programming theory, it can be shown that there are interesting polytopes $P \subseteq \mathbb{R}^n$, which can only be defined with an exponential (in n) number of inequalities, but which can be cast as a linear projection of a higher dimensional polytope Q that can be defined with polynomially many variables and constraints. Such a polytope Q is called an extended formulation of P. Extended formulations of polynomial size play a crucial role in combinatorial optimization because they provide a unified framework to obtain polynomial time algorithms for a large variety of combinatorial problems. For this reason, extended formulations of polytopes associated with formal languages and with groups have been studied intensively during the past decades, both from the perspective of lower bounds [37, 21, 39, 34, 4, 13, 29], and from the perspective of upper bounds [10, 17, 9, 10, 35, 19, 39, 14, 15].

By combining our main theorem 5 with a connection established by Pesant, Quimper, Rousseau and Sellmann [33] between the extension complexity of a permutation group and the grammar complexity of a formal language, we show that any permutation group that can be embedded in a connected graph with m vertices, treewidth k, and maximum degree Δ can be represented by polytopes of extension complexity $2^{O(k\Delta \log \Delta)} \cdot m^{O(k)}$ (Theorem 14).

By combining our upper bound from Theorem 5 with the $2^{\Omega(n)}$ lower bound from [18], we obtain an interesting complexity theoretic trade-off relating the index of a permutation group with the size, treewidth and maximum degree of a graph embedding this group (Theorem 19). As a corollary of this trade-off, we show that subgroups of $\mathbb{S}_n$ with index up to 2^{cn} for some small constant c have superpolynomial graph embedding complexity on classes of graphs with treewidth $o(n/\log n)$ and maximum degree $o(n/\log n)$ (Corollary 20). Additionally, this lower bound can be improved from super-polynomial to exponential on classes of graphs of treewidth n^{ε} (for $\varepsilon < 1$) and maximum degree $o(n/\log n)$ (Corollary 21). In particular, Corollary 21 implies exponential lower bounds for minor-closed families of connected graphs (which have treewidth $\sqrt{n}$).

Related Work. Proving lower bounds for the size of graphs embedding a given permutation group is a challenging and still not well understood endeavour. It is worth noting that it is still not known whether the alternating group $\mathbb{A}_n$ can be embedded in a graph with $n^{O(1)}$ vertices. We note that by solving an open problem stated by Babai in [7], Liebeck has shown that any graph whose automorphism group is isomorphic to the alternating group (as an abstract group) must have at least $2^{\Omega(n)}$ vertices [30]. Nevertheless, a similar result has not yet been obtained in the setting of graph embedding of groups, and indeed, constructing an explicit sequence of groups that have superpolynomial graph embedding complexity is a long-standing open problem [8]. Our results in Corollary 20 and Corollary 21 provide unconditional lower bounds for interesting classes of graphs for any group of relatively small index (index at most 2^{cn} for some small enough constant c).

The crucial difference between the abstract isomorphism setting considered in [30] and our setting is in the way in which graphs are used to represent groups. In the setting of [30], given a group G, the goal is to construct a graph X whose automorphism group is *isomorphic* to G. On the other hand, in the graph embedding setting, we want the group G to be *equal* to the action of the automorphism group $\mathrm{Aut}(X)$ on its first $[n]$ vertices. In the abstract isomorphism setting it has been shown by Babai that for any class of graphs $\mathcal{X}$ excluding a fixed graph H as a minor, there exists some finite group which is not isomorphic to the automorphism group of any graph in $\mathcal{X}$ [6]. Our Corollary 21 can be regarded as a result in this spirit in the context of graph embedding. While the lower bound stated in Corollary 21 also applies to graphs that are not minor closed, this lower boud is only meaningful for graphs of maximum degree at most $o(n/\log n)$.

We observe that in Theorem 5 an exponential dependence on the maximum degree parameter Δ is unavoidable. Indeed, as stated above, the symmetric grammar complexity of the language $\mathrm{str}(\mathbb{S}_n)$ is $2^{\Theta(n)}$. On the other hand, for each $n \in \mathbb{N}_+$, the symmetric group $\mathbb{S}_n$ can be embedded in the star graph $K_{n,1}$ with vertex set $V(K_{n,1}) = \{1, ..., n+1\}$, and edge set $E(K_{n,1}) = \{\{i, n+1\} \ : \ i \in \{1, \ldots, n\}\}$, which is a connected graph of treewidth 1. Nevertheless, it is not clear to us whether the logarithmic factor $\log \Delta$ can be shaved from the exponent of the upper bound $2^{O(k\Delta \log \Delta)} \cdot m^{O(k)}$. We also note that the connectedness requirement is also crucial for our upper bounds since $\mathbb{S}_n$ can be embedded in the discrete graph D_n with vertex set $D_n = \{1, ..., n\}$, and edge set $E(D_n) = \emptyset$.

2 Preliminaries

Proofs of statements marked with '♠' are deferred to the full version. We let $\mathbb{N}$ denote the set of non-negative integers and $\mathbb{N}_+ = \mathbb{N} \setminus \{0\}$ denote the set of positive integers. For each $n \in \mathbb{N}_+$, we let $[n] = \{1, ..., n\}$. For each finite set S we let $\mathcal{P}(S) = \{S' \ : \ S' \subseteq S\}$ denote the set of all subsets of S. For each set S and each $k \in \mathbb{N}$, we let $\binom{S}{k} = \{S' \subseteq S \ : \ |S'| = k\}$ be the set of subsets of S of size k and $\binom{S}{\leq k} = \bigcup_{i=0}^{k} \binom{S}{i}$ the set of subsets of size at most k. For a function $f \colon X \to Y$ and a set $X' \subseteq X$, we denote by $f|_{X'}$ the *restriction of f to X'*, i.e. the function $f|_{X'} \colon X' \to Y$ with $f|_{X'}(x) = f(x)$ for each $x \in X'$.

Prefix Closed Sets. For each $r \in \mathbb{N}_+$, we let $[r]^*$ be the set of all strings over $[r]$, including the empty string λ. Let p and u be strings in $[r]^*$. We say that p is a *prefix* of u if there exists $q \in [r]^*$ such that $u = pq$. Note that u is a prefix of itself, and that the empty string λ is a prefix of each string in $[r]^*$. A non-empty subset $U \subseteq [r]^*$ is *prefix closed* if for each $u \in U$, each prefix of u is also in U. We note that the empty string λ is an element of any prefix closed subset of $[r]^*$. We say that $U \subseteq [r]^*$ is *well numbered* if for each $p \in [r]^*$ and each $j \in [r]$, the presence of pj in U implies that $p1, ..., p(j-1)$ also belong to U.

Tree-Like Sets. We say that a subset $U \subseteq [r]^*$ is *tree-like* if U is both prefix-closed and well-numbered. Let U be a tree-like subset of $[r]^*$. If $pj \in U$, then we say that pj is a *child* of p, or interchangeably, that p is the *parent* of pj. If $pu \in U$ for $u \in [r]^*$, then we say that pu is a descendant of p. For a node $p \in U$ we let $U|_p = \{pu \in U \ : \ u \in [r]^*\}$ denote the set of all descendants of p. Note that p is a descendant of itself and therefore, $p \in U|_p$. A *leaf* of U is a node $p \in U$ without children. We let $\mathit{leaves}(U)$ be the set of leaves of U, and $\mathit{leaves}(U, p)$ be the set of leaves which are descendants of p.

Terms. Let Σ be a finite set of symbols. An *r-ary term* over Σ is a function $t : \mathit{Pos}(t) \to \Sigma$ whose domain $\mathit{Pos}(t)$ is a tree-like subset of $[r]^*$. We denote by $\mathit{Ter}(\Sigma)$ the set of all terms over Σ. If $t_1, ..., t_r$ are terms in $\mathit{Ter}(\Sigma)$, and $a \in \Sigma$, then we let $t = a(t_1, ..., t_r)$ be the term in $\mathit{Ter}(\Sigma)$ which is defined by setting $t(\lambda) = a$ and $t(jp) = t_j(p)$ for each $j \in [r]$ and each $p \in \mathit{Pos}(t_j)$.

3 Embedding Permutation Groups in Graphs

For each finite set Γ, we let $\mathbb{S}(\Gamma)$ be the group of permutations of Γ. If $\Omega \subseteq \Gamma$ and $\alpha \in \mathbb{S}(\Gamma)$, then we say that α stabilizes Ω setwise if $\alpha(\Omega) = \Omega$. Alternatively, we say that Ω is invariant under α. We let α_Ω be the permutation in $\mathbb{S}(\Omega)$ which is defined by setting $\alpha_\Omega(i) = \alpha(i)$ for each $i \in \Omega$. In other words, α_Ω is the restriction of α to Ω. If G is a subgroup of $\mathbb{S}(\Gamma)$, then we let $\mathrm{stab}(G, \Omega)$ be the set of permutations in G that stabilize Ω setwise. We say that a group G stabilizes Ω if $\mathrm{stab}(G, \Omega) = G$. Alternatively, we say that Ω is invariant under G. We let $G|_\Omega = \{\alpha|_\Omega \ : \ \alpha \in G\}$ be the set of restrictions of permutations in G to Ω. In what follows, for each $n \in \mathbb{N}_+$ we write $\mathbb{S}_n$ to denote $\mathbb{S}([n])$.

Graphs. Let $m \in \mathbb{N}_+$. An m-vertex graph is a pair $X = ([m], E(X))$, where $E(X) \subseteq \binom{[m]}{2}$.

Isomorphisms and Automorphisms. If X and Y are two m-vertex graphs, then an *isomorphism* between X and Y is a permutation $\alpha \in \mathbb{S}_m$ such that for each $\{i, j\} \in \binom{[m]}{2}$, $\{i, j\} \in E(X)$ if and only if $\{\alpha(i), \alpha(j)\} \in E(Y)$. An *automorphism* of X is an isomorphism

between X and X. We let $\mathrm{Iso}(X,Y)$ denote the set of all isomorphisms between X and Y, and let $\mathrm{Aut}(X) = \mathrm{Iso}(X,X)$ be the set of automorphisms of X. If $\Omega \subseteq [m]$ is invariant under $\mathrm{Aut}(X)$ then we define $\mathrm{Aut}(X,\Omega) = \mathrm{Aut}(X)|_\Omega = \{\alpha|_\Omega \ : \ \alpha \in \mathrm{Aut}(X)\}$.

▶ **Definition 1.** *Let G be a subgroup of $\mathbb{S}_n$ and X be a connected m-vertex graph where $m \geq n$. We say that G is embeddable in X if $\mathrm{Aut}(X,[n]) = G$.*

In other words, G is embeddable in X if the image of action of the automorphism group of X on its first n vertices is equal to G. We note that the requirement that the graph X of Definition 1 is connected is crucial for our applications.

Let $\mathcal{X}$ be a class of connected graphs and G be a subgroup of $\mathbb{S}_n$. We say that G is *$\mathcal{X}$-embeddable* if there exists some graph $X \in \mathcal{X}$ such that G is embeddable in X. The *$\mathcal{X}$ embedding complexity* of G, denoted by $\mathrm{gec}_{\mathcal{X}}(G)$ is the minimum m such that G is embeddable in a graph $X \in \mathcal{X}$ with at most m vertices. If no such a graph $X \in \mathcal{X}$ exists, then we set $\mathrm{gec}_{\mathcal{X}}(G) = \infty$.

4 Using Grammars to Represent Finite Permutation Groups

A *context-free grammar* is a 4-tuple $\mathfrak{G} = (\Sigma, \mathcal{B}, R, B_1)$ where Σ is a finite set of symbols, $\mathcal{B}$ is a finite set of variables, $R \subseteq \mathcal{B} \times (\Sigma \cup \mathcal{B})^*$ is a finite set of *production rules*, and $B_1 \in \mathcal{B}$ is the *initial variable* of $\mathfrak{G}$. The notion of a string w generated by $\mathfrak{G}$ can be defined with basis on the notions of *$\mathfrak{G}$-parse-tree* and *yield* of a $\mathfrak{G}$-parse-tree, which are inductively defined as follows.

1. For each $a \in \Sigma \cup \{\lambda\}$ the term $t : \{\lambda\} \to \Sigma \cup \{\lambda\}$ which sets $t(\lambda) = a$ is a $\mathfrak{G}$-parse-tree. Additionally, $\mathrm{yield}(t) = a$.
2. If $t_1, ..., t_r$ are $\mathfrak{G}$-parse-trees and $B \to t_1(\lambda) \cdot t_2(\lambda) \cdot ... \cdot t_r(\lambda)$ is a production rule in R, then the term $t = B(t_1, ..., t_r)$ is a $\mathfrak{G}$-parse-tree. Additionally, $\mathrm{yield}(t) = \mathrm{yield}(t_1) \cdot \mathrm{yield}(t_2) \cdot ... \cdot \mathrm{yield}(t_r)$. In other words, the yield of t is the concatenation of the yields of the subterms $t_1, \ldots, t_r$.

We say that a $\mathfrak{G}$-parse-tree t is *accepting* if $t(\lambda) = B_1$. We say that a string $w \in \Sigma^*$ is generated by $\mathfrak{G}$ if there is an accepting $\mathfrak{G}$-parse-tree with $\mathrm{yield}(t) = w$. The language generated by $\mathfrak{G}$ is the set $\mathcal{L}(\mathfrak{G}) = \{w \in \Sigma^* \ : \ w \text{ is generated by } \mathfrak{G}\}$ of strings generated by $\mathfrak{G}$. The size of $\mathfrak{G}$ is defined as $|\mathfrak{G}| = \sum_{(B,u)\in R}(1 + |u|)\log(|\Sigma| + |\mathcal{B}|)$, where $|u|$ is the number of symbols/variables in u, $|\Sigma|$ is the number of elements in Σ and $|\mathcal{B}|$ is the number of elements in $\mathcal{B}$. We denote by $\mathbb{G}(\Sigma)$ the set of context-free grammars over the alphabet Σ.

A context-free grammar $\mathfrak{G}$ is said to be regular if each production rule is either of the form (B, a) for some $B \in \mathcal{B}$ and $a \in \Sigma$, or of the form (B, aB') for some $B, B' \in \mathcal{B}$ and some $a \in \Sigma$. We denote by $\mathbb{RG}(\Sigma)$ the set of regular context-free grammars over the alphabet Σ.

Complexity Measures. If $\alpha \in \mathbb{S}_n$ and $w \in \Sigma^n$ then we let $\mathrm{Perm}(w,\alpha) \stackrel{def}{=} w_{\alpha(1)}w_{\alpha(2)}...w_{\alpha(n)}$ be the string obtained by permuting the positions of w according to α. If $L \subseteq \Sigma^n$ then we let $\mathrm{Perm}(L,\alpha) \stackrel{def}{=} \{\mathrm{Perm}(w,\alpha) \ : \ w \in L\}$. In other words, $\mathrm{Perm}(L,\alpha)$ is the language obtained by permuting the positions of each string $w \in L$ according to α. The *symmetric grammar complexity* of a language $L \subseteq \Sigma^n$ is defined as the minimum size of a context-free grammar generating $\mathrm{Perm}(L,\alpha)$ for some $\alpha \in \mathbb{S}_n$. More precisely, $\mathrm{sgc}(L) = \min\{|\mathfrak{G}| \ : \ \exists \alpha \in \mathbb{S}_n,\ \mathfrak{G} \in \mathbb{G}(\Sigma),\ \mathcal{L}(\mathfrak{G}) = \mathrm{Perm}(L,\alpha)\}$.

Analogously, the *symmetric regular grammar complexity* of a language $L \subseteq \Sigma^n$ is defined as the minimum size of a *regular* grammar generating $\mathrm{Perm}(L, \alpha)$ for some $\alpha \in \mathbb{S}_n$. $\text{reg-sgc}(L) = \min\{|\mathfrak{G}| : \exists \alpha \in \mathbb{S}_n,\ \mathfrak{G} \in \mathbb{RG}(\Sigma),\ \mathcal{L}(\mathfrak{G}) = \mathrm{Perm}(L, \alpha)\}$.

We note that the *symmetric regular grammar complexity* of a language $L \subseteq \Sigma^n$ is polynomially related to the minimum size of an acyclic non-deterministic finite automaton accepting some permuted version of L, or equivalently to the minimum size of a non-deterministic read-once oblivious branching program accepting L. On the other hand, the symmetric context-free complexity of a language L is polynomially related to the minimum size of a pushdown automaton accepting some permuted version of L.

Let $\alpha : [n] \to [n]$ be a permutation in $\mathbb{S}_n$. We let $str(\alpha) = \alpha(1)\alpha(2)...\alpha(n) \in [n]^n$ be the string associated with α. For each group $G \sqsubseteq \mathbb{S}_n$ we let $str(G) = \{str(\alpha) : \alpha \in G\}$ be the language associated with G. The symmetric grammar complexity of G is defined as $\mathrm{sgc}(G) \stackrel{def}{=} \mathrm{sgc}(str(G))$. Analogously, the regular grammar complexity of G is defined as $\text{reg-sgc}(G) \stackrel{def}{=} \text{reg-sgc}(str(G))$.

If $\beta : [n] \to [n]$ and $\gamma : [n] \to [n]$ are permutations in $\mathbb{S}_n$, then we let $\beta \circ \gamma$ be the permutation that sends each $i \in [n]$ to the number $\beta(\gamma(i))$. If S is a subset of $\mathbb{S}_n$, we let $\beta \circ S \stackrel{def}{=} \{\beta \circ \gamma : \gamma \in S\}$. Note that if G is a subgroup of $\mathbb{S}_n$, H is a subgroup of G, and $\beta \in G$, then $\beta \circ H$ is a left coset of H in G. The following proposition, which will be used in the proofs of Lemma 16 and Theorem 5 follows from the fact that context-free languages are closed under homomorphisms.

▶ **Proposition 2 (♠).** *Let $H \subseteq \mathbb{S}_n$, and α be a permutation in $\mathbb{S}_n$. Let $\mathfrak{G}$ be a context-free grammar such that $\mathcal{L}(\mathfrak{G}) = \mathrm{Perm}(\mathrm{str}(H), \alpha)$. Then for each permutation $\beta \in \mathbb{S}_n$ there is a context-free grammar $\mathfrak{G}_\beta$ of size $|\mathfrak{G}_\beta| = |\mathfrak{G}|$ generating $\mathrm{Perm}(\mathrm{str}(\beta \circ H), \alpha)$.*

The following theorem, which will be crucial to the proof of our main result (Theorem 5), upper bounds the symmetric grammar complexity of the automorphism group of a graph in terms of the number of its vertices, its maximum degree, and its treewidth. If the latter two quantities are bounded, then this upper bound is polynomial in the number of its vertices.

▶ **Theorem 3.** *Let X be a connected graph with n vertices, treewidth k and maximum degree Δ. Then*

$$\mathrm{sgc}(\mathrm{Aut}(X)) \leq 2^{O(k\Delta \log \Delta)} \cdot n^{O(k)}.$$

Additionally, one can construct in time $2^{O(k\Delta \log \Delta)} \cdot n^{O(k)}$ a permutation $\alpha \in \mathbb{S}_n$ and a context-free grammar $\mathfrak{G}(X)$ generating the language $\mathrm{Perm}(\mathrm{str}(\mathrm{Aut}(X)), \alpha)$.

▶ **Remark 4.** *If the graph X of Theorem 3 has pathwidth k, then one may assume that $\mathfrak{G}(X)$ is a regular grammar. In other words, in this case,* $\text{reg-sgc}(\mathrm{Aut}(X)) \leq 2^{O(k\Delta \log \Delta)} \cdot n^{O(k)}$.

Theorem 3 can be simultaneously generalized in two ways. First, by allowing grammars to represent not only the automorphism group of a graph, but also groups that can be embedded in the graph. Second, not only the groups themselves but also left cosets of such groups can be represented in the same way. The result of these generalizations is stated in the next theorem.

▶ **Theorem 5.** *Let $G \sqsubseteq \mathbb{S}_n$, and suppose that G is embeddable in a connected graph X with m vertices ($m \geq n$), maximum degree Δ, and treewidth k. Then, for each $\beta \in \mathbb{S}_n$,*

$$\mathrm{sgc}(\beta \circ G) \leq 2^{O(k\Delta \log \Delta)} \cdot m^{O(k)}.$$

Additionally, given X and β, one can construct in time $2^{O(k\Delta \log \Delta)} \cdot m^{O(k)}$ a permutation $\alpha \in \mathbb{S}_n$ (depending only on X) and a grammar $\mathfrak{G}_\beta$ generating the language $\mathrm{Perm}(\mathrm{str}(\beta \circ G), \alpha)$.

▶ **Remark 6.** *If the graph X of Theorem 5 has pathwidth k, then one may assume that $\mathfrak{G}_\beta(X)$ is a regular grammar. In other words, in this case,* $\text{reg-sgc}(\beta \circ G) \leq 2^{O(k\Delta \log \Delta)} \cdot m^{O(k)}$.

4.1 Proof of Theorem 5

In this section, we will prove Theorem 5, which establishes an upper bound for the symmetric grammar complexity of a permutation group G in function of the size, treewidth and maximum degree of a graph embedding G. On the way to prove Theorem 5, we will first prove Theorem 3. The proofs of Remarks 4 and 6 follow by small adaptations of the proofs of Theorems 3 and 5 respectively.

Subtree-Like Sets and Subterms. Let $r \in \mathbb{N}_+$, $U \subseteq [r]^*$ be a tree-like set, $p, q \in U$ and u be the longest common prefix of p and q. Let $p = up'$ and $q = uq'$. The *distance* between p and q is defined as $|p'| + |q'|$. We call a set $M \subseteq U$ *subtree-like* if there exists a $p \in M$, such that $M = U|_p$. We let $M' = \{u \mid pu \in M\}$ be the tree-like set *induced by* M. For a set $U' \subseteq U$, we call the smallest subtree-like set containing U' the *closest ancestral closure of* U'. For any subtree-like set $M \subseteq Pos(t)$, we call $t|_M$ a *subterm* of t. If M' is the induced tree-like set of M, then we call the corresponding term t' with $Pos(t') = M'$ the *t-term induced by* M. For a position $p \in Pos(t)$, we denote by $t|_p$ the subterm of t rooted at p, i.e. we let $t|_p \stackrel{def}{=} t|_{Pos(t)|_p}$.

Neighborhood of a Vertex, and Induced Subgraphs. Let X be an n-vertex graph. For a vertex $v \in [n]$, we let $N(v) \stackrel{def}{=} \{u \in [n] \ : \ \{v, u\} \in E(X)\}$ be the *neighborhood* of v. If $S \subseteq [n]$ then we let $N(S) = \bigcup_{v \in S} N(v)$ be the neighborhood of S. Finally, we let $\overline{N}(S) = N(S) \cup S$ be the *closed neighborhood* of S. The subgraph of X induced by S is defined as $X[S] = (S, E(X[S]))$ where $E(X[S]) = E(X) \cap \binom{S}{2}$.

Tree decomposition as Terms. If we regard the set $\binom{V(X)}{\leq k+1}$ as an alphabet, then each width-k tree decomposition of a graph X may be regarded as a term over $\binom{V(X)}{\leq k+1}$. More precisely, let X be an n-vertex graph and $k \in \{0, 1, \ldots, n-1\}$. A *width-$k$ tree decomposition* (or simply *tree decomposition*, if k is clear from the context) of X is a term $\mathbf{t} \in Ter(\binom{V(X)}{\leq k+1})$ satisfying the following axioms.

(T1) $\bigcup_{p \in Pos(\mathbf{t})} \mathbf{t}(p) = V(X)$

(T2) For each vertex $v \in V(X)$ and each of its neighbors $u \in N(v)$, there is a position $p \in Pos(\mathbf{t})$ such that $\{v, u\} \subseteq \mathbf{t}(p)$.

(T3) For each vertex $v \in V(X)$, the set $\{p \in Pos(\mathbf{t}) \mid v \in \mathbf{t}(p)\}$ induces a subterm of $\mathbf{t}$.

The *treewidth* of X, is defined as the smallest non-negative integer $k \in \mathbb{N}$ such that X admits a width-k tree decomposition.

Annotated Tree Decompositions. Let X be an n-vertex graph, S and S' be subsets of $[n]$ such that $|S| = |S'|$, and $\nu : S \to S'$ be a bijection. We say that ν is a *partial automorphism* of X if ν is an isomorphism from the subgraph $X[S]$ of X induced by S to the subgraph $X[S']$ of X induced by S'. Next, we define the notion of *annotated tree decomposition* of a graph X. These are tree-decompositions whose bags are annotated with partial automorphisms.

▶ **Definition 7** (Annotated Bags). *Let X be an n-vertex graph and $k \in \{0, \ldots, n-1\}$. A* k-annotated bag *is a pair (S, ν), where $S \in \binom{V(X)}{\leq k+1}$, and $\nu \colon \overline{N}[S] \to V(X)$ is a function satisfying the following two properties.*

1. $\nu(\overline{N}(S)) = \overline{N}(\nu(S))$. *In other words, the image of* $\overline{N}(S)$ *under* ν *is equal to the closed neighborhood of the image of* S *under* ν.
2. ν *is a partial automorphism of* X.

We let $\mathfrak{B}(X,k)$ be the set of all k-annotated bags of X. If b is a k-annotated bag in $\mathfrak{B}(X,k)$, then we denote the first coordinate of b by $b.S$ and the second coordinate of b by $b.\nu$. In other words, $b = (b.S, b.\nu)$. We let $\rho\colon \mathfrak{B}(X,k) \to \binom{V(X)}{\leq w+1}$ be the map that takes an annotated bag $b \in \mathfrak{B}(X,k)$ and sends it to the bag $\rho(b) = b.S \in \binom{V(X)}{\leq k+1}$. In other words, the map ρ erases the second coordinate of the annotated bag b. We extend ρ to terms in $Ter(\mathfrak{B}(X,k))$ positionwise. More precisely, for each term $\hat{t} \in Ter(\mathfrak{B}(X,k))$, we let $\rho(\hat{t})$ be the term in $Ter(\binom{V(X)}{\leq k+1})$ where $Pos(\rho(\hat{t})) \stackrel{def}{=} Pos(\hat{t})$ and $\rho(\hat{t})(p) \stackrel{def}{=} \rho(\hat{t}(p))$ for each $p \in Pos(t)$. We say that a term $\hat{t} \in Ter(\mathfrak{B}(X,k))$ is an *annotation* of a term $t \in Ter(\binom{V(X)}{\leq k+1})$ if $\rho(\hat{t}) = t$. Note that a term $t \in Ter(\binom{V(X)}{\leq k+1})$ may have many annotations.

We give an upper bound on the number of annotated bags, see Definition 7. There are at most $\mathcal{O}(n^{k+1})$ choices for the set S. Once such a set S is fixed, there are (at most) $\mathcal{O}(n^{k+1})$ ways of mapping the vertices in S to vertices in X. (In other words, there are at most $\mathcal{O}(n^{k+1})$ choices for the image of S under the partial automorphism ν.) Once the image of S is fixed, for each vertex $x \in S$ there are at most $\Delta!$ ways of mapping the neighbors of x to the neighbors of $\nu(x)$. Hence there are at most $(\Delta!)^{k+1}$ choices for obtaining a partial automorphism for a fixed image of S. Therefore, by noting that $\Delta! = 2^{\mathcal{O}(\Delta \log \Delta)}$, we have the following observation.

▶ **Observation 8.** *Let* X *be a graph of maximum degree* Δ *and let* $k \in \{0, \ldots, n-1\}$. *Then,* $|\mathfrak{B}(X,k)| \leq 2^{\mathcal{O}(k\Delta \cdot \log \Delta)} \cdot n^{\mathcal{O}(k)}$.

▶ **Definition 9** (Annotated Tree Decomposition)**.** *Let* $\hat{\mathbf{t}}$ *be a term in* $Ter(\mathfrak{B}(X,k))$. *We say that* $\hat{\mathbf{t}}$ *is an* annotated width-k tree decomposition *if the following conditions are satisfied.*
1. $\rho(\hat{\mathbf{t}})$ *is a tree decomposition.*
2. *for each* $p \in Pos(\hat{\mathbf{t}})$ *with children* $p1, \ldots, pd$, *and for each* $j \in [d]$, *the restriction of* $\hat{\mathbf{t}}(p).\nu$ *to* $N[\hat{\mathbf{t}}(p).S] \cap N[\hat{\mathbf{t}}(pj).S]$ *is equal to the restriction of* $\hat{\mathbf{t}}(pj).\nu$ *to* $N[\hat{\mathbf{t}}(p).S] \cap N[\hat{\mathbf{t}}(pj).S]$.

Intuitively, the first condition states that if we take an annotated tree decomposition $\hat{\mathbf{t}}$ and forget annotation then the result is a tree-decomposition of X. The second condition guarantees that the annotation is consistent along the whole tree decomposition, in the sense that for each vertex $x \in V(X)$, if the partial automorphism of one bag sends x to vertex x', then the partial automorphism of each bag sends x to x'. Each annotated tree decomposition $\hat{\mathbf{t}}$ gives rise to a map $\mu(\hat{\mathbf{t}})\colon V(X) \to V(X)$ which sets $\mu(\hat{\mathbf{t}})|_{N[\hat{\mathbf{t}}(p).S]} = \hat{\mathbf{t}}(p).\nu$ for each $p \in Pos(\hat{\mathbf{t}})$. We call the map μ the *annotation morphism* of $\hat{\mathbf{t}}$. The following lemma is the main technical tool of this section.

▶ **Lemma 10 (♠).** *Let* X *be an* n*-vertex graph of treewidth* k *and* $\alpha \in \mathbb{S}_n$. *Then,* α *is an automorphism of* X *if and only if there exists an annotated tree decomposition* $\hat{\mathbf{t}}$ *of* X *such that* $\alpha = \mu(\hat{\mathbf{t}})$.

▶ **Definition 11.** *A tree decomposition* $\mathbf{t}$ *is called* permutation yielding, *if there is a bijection* $\pi\colon leaves(Pos(\mathbf{t})) \to V(X)$ *such that for each leaf* $p \in leaves(Pos(\mathbf{t}))$, $\mathbf{t}(p) = \{\pi(p)\}$.

In other words, a tree decomposition $\mathbf{t}$ is permutation yielding if each vertex occurs in precisely one leaf bag. The next lemma shows that any tree decomposition $\mathbf{t}$ can be transformed in polynomial time into a permutation yielding tree decomposition of the same width. We note that a statement analogous to Lemma 12 can also be obtained by observing that tree-decompositions can be converted in polynomial time into branch decompositions of roughly the same width [36]. We include a proof of Lemma 12 for completeness.

▶ **Lemma 12 (♠).** *Let X be an n-vertex graph, $k \in \{0, \ldots, n-1\}$, and $\mathbf{t}$ a width-k tree decomposition of X. Then, one can construct from $\mathbf{t}$ in polynomial time a permutation yielding width-k tree decomposition.*

Let $\hat{\mathbf{t}}$ be an annotated tree decomposition with r leaves, and let $\text{yield}(\hat{\mathbf{t}}) = (S_1, \nu_1) \ldots (S_r, \nu_r)$ be the yield of $\hat{\mathbf{t}}$. In other words, $\text{yield}(\hat{\mathbf{t}})$ is the sequence of annotated bags obtained by reading the leaves of $\hat{\mathbf{t}}$ from left to right. We define the *annotation yield* of $\hat{\mathbf{t}}$ as the sequence $\text{yield}_\nu(\hat{\mathbf{t}}) \stackrel{def}{=} \nu_1(S_1) \ldots \nu_m(S_r)$. Note that if $\mathbf{t}$ is a permutation yielding tree decomposition of X, and $\hat{\mathbf{t}}$ is an annotation of $\mathbf{t}$, then $r = |V(X)|$, and $\text{yield}_\nu(\hat{\mathbf{t}})$ is a string of singletons of the form $\{v_1\}\{v_2\} \ldots \{v_r\}$ where $v_i \in V(X)$ for each $i \in [r]$, and $v_i \neq v_j$ for $i \neq j$.

▶ **Restatement of Theorem 3.** *Let X be a connected graph with n vertices, treewidth k and maximum degree Δ. Then $\text{sgc}(\text{Aut}(X)) \leq 2^{O(k\Delta \log \Delta)} \cdot n^{O(k)}$. Additionally, one can construct in time $2^{O(k\Delta \log \Delta)} \cdot n^{O(k)}$ a permutation α and a context-free grammar $\mathfrak{G}(X)$ generating $\text{Perm}(\text{str}(\text{Aut}(X)), \alpha)$.*

Proof. Since the graph X has treewidth k, one can construct in time $2^{O(k)} \cdot n^{O(1)}$ a width $O(k)$ tree decomposition $\mathbf{t}$ of X. Additionally, from Lemma 12, one can assume that $\mathbf{t}$ is yielding. Let $\text{yield}(\mathbf{t}) = \{v_1\}\{v_2\}...\{v_n\}$. Then we let $\alpha_{\mathbf{t}}$ be the permutation in $\mathbb{S}_n$ with $\text{str}(\alpha_{\mathbf{t}}) = v_1 v_2 ... v_n$. We set $\alpha = \alpha_{\mathbf{t}}^{-1}$. Since $\mathbf{t}$ can be constructed in time $2^{O(k)} \cdot n^{O(1)}$, so can the permutation α. We show that from $\mathbf{t}$ one can construct a context-free grammar $\mathfrak{G}$ accepting the language $\mathcal{L}(\mathfrak{G}) = \text{Perm}(str(\text{Aut}(X)), \alpha)$. Intuitively, the parse trees accepted by the grammar $\mathfrak{G}$ correspond to annotations of $\mathbf{t}$, and by Lemma 10, these annotations correspond to automorphisms of X. Formally, the grammar $\mathfrak{G} = (\Sigma, \mathcal{B}, R, B_1)$ is defined as follows. We let $\Sigma = V(X) = [n]$ and $\mathcal{B} = (Pos(\mathbf{t}) \times \mathfrak{B}(X,k)) \cup \{B_1\}$ where B_1 is the initial variable of $\mathfrak{G}$. Recall that $\rho \colon \mathfrak{B}(X,k) \to \binom{V(X)}{\leq k+1}$ is the map that erases the second coordinate from each annotated bag $b \in \mathfrak{B}$. The set R contains the following rules.

1. A rule $B_1 \to (\lambda, b)$ for each annotated bag $b \in \mathfrak{B}(X,k)$ such that $\rho(b) = \mathbf{t}(\lambda)$. Intuitively, each such b is an annotated bag corresponding to the bag at the root of $\mathbf{t}$.
2. For each non-leaf position $p \in Pos(\mathbf{t}) \setminus leaves(Pos(\mathbf{t}))$, with children $p1, \ldots, pd$, we have a rule $(p, b) \to (p1, b_1)(p2, b_2) \ldots (pd, b_d)$, for each sequence $b, b_1, ..., b_d$ of annotated bags in $\mathfrak{B}(X,k)$ satisfying the following conditions:
   - **(i)** $\rho(b) = \mathbf{t}(p)$ and for $j \in [d]$, $\rho(b_j) = \mathbf{t}(pj)$, and
   - **(ii)** for each $j \in [d]$, $b.\nu|_{S^*} = b_j.\nu|_{S^*}$ where $S^* = b.S \cap b_j.S$.
3. A rule $(p, b) \to j$ for each leaf position $p \in Pos(\mathbf{t})$ with $b.S = \{i\}$ and $b.\nu(i) = j$.

These rules defined above ensure that if we take an accepting parse tree t of $\mathfrak{G}$ and remove its root (i.e the variable B_1) and its leaves (which are labeled with numbers in $[n]$) then we are left with an annotated version $\hat{\mathbf{t}}$ of the tree decomposition $\mathbf{t}$. By Lemma 10, $\hat{\mathbf{t}}$ is an annotation of $\mathbf{t}$ if and only if the map $\mu(\hat{\mathbf{t}}) : V(X) \to V(X)$ is an automorphism of X. Therefore, since $\text{str}(\mu(\hat{\mathbf{t}})) = \text{yield}(t)$, we have that $\mathcal{L}(\mathfrak{G}) = \text{Perm}(str(\text{Aut}(X)), \alpha)$. Since we can assume that $|Pos(\mathbf{t})| = \mathcal{O}(kn)$ (see e.g. [16, Lemma 7.4]), and for each bag, there are at most $2^{\mathcal{O}(k\Delta \log \Delta)} \cdot n^{\mathcal{O}(k)}$ annotations, we have that $|\mathfrak{G}| = 2^{\mathcal{O}(k\Delta \log \Delta)} \cdot n^{\mathcal{O}(k)}$, as claimed. ◀

▶ **Restatement of Theorem 5.** *Let $G \sqsubseteq \mathbb{S}_n$, and suppose that G is embeddable in a connected graph X with m vertices ($m \geq n$), maximum degree Δ, and treewidth k. Then, for each $\beta \in \mathbb{S}_n$, $\text{sgc}(\beta \circ G) \leq 2^{O(k\Delta \log \Delta)} \cdot m^{O(k)}$. Additionally, given X and β, one can construct in time $2^{O(k\Delta \log \Delta)} \cdot m^{O(k)}$ a permutation $\alpha \in \mathbb{S}_n$ (depending only on X) and a grammar $\mathfrak{G}_\beta$ generating the language $\text{Perm}(\text{str}(\beta \circ G), \alpha)$.*

Proof. This is a consequence of Theorem 3, together with the fact that context free grammars are closed under homomorphisms. More precisely, we first construct in time $2^{O(k\Delta \log \Delta)} \cdot m^{O(k)}$ a permutation $\alpha' \in \mathbb{S}_m$, and a context-free grammar $\mathfrak{G}'$ such that $\mathcal{L}(\mathfrak{G}) = \mathrm{Perm}(str(\mathrm{Aut}(X)), \alpha)$. Now let $h : [m]\backslash[n] \to \{\lambda\}$ be the map that sends each number in $[m]\backslash[n]$ to the empty string λ. Then using $\mathfrak{G}$, one can construct in time polynomial in $|\mathfrak{G}'|$ a context-free grammar $\mathfrak{G}''$ whose language $\mathcal{L}(\mathfrak{G}'')$ is the homomorphic image of $\mathcal{L}(\mathfrak{G}')$ under h. Additionally, one may assume that the grammar $\mathfrak{G}''$ has no production rule containing the empty string λ. Let $\alpha = \alpha'|_{[n]}$ be the permutation in $\mathbb{S}_n$ obtained by restricting α' to $[n]$. Note that α is well defined, since the fact that $G \sqsubseteq \mathbb{S}_n$ is embeddable in X implies that $\alpha'([n]) = [n]$, and therefore that $\alpha([n]) = [n]$. Then we have that the language accepted by $\mathfrak{G}''$ is $\mathcal{L}(\mathfrak{G}'') = \mathrm{Perm}(\mathrm{str}(\mathrm{Aut}(X)), \alpha)$.

Finally, let $\beta : [n] \to [n]$ be a permutation in $\mathbb{S}_n$. Then we can regard β as a usual map from $[n]$ to $[n]$, and using again the fact that context-free languages are closed under homomorphism, we can construct in time $O(|\mathfrak{G}''|)$ a context-free grammar $\mathfrak{G}$ accepting the homomorphic image of $\mathcal{L}(\mathfrak{G}'')$ under β. This homomorphic image is simply the language $\mathrm{Perm}(str(\beta \circ G), \alpha)$. ◀

5 Polytopes for Permutation Groups

In linear-programming theory, the n-permutahedron is the polytope $P(\mathbb{S}_n)$ formed by the convex-hull of the set of permutations of the set $\{1, \ldots, n\}$. It can be shown that to define the permutahedron on the n-dimensional space, $2^{\Omega(n)}$ constraints are required. On the other hand, a celebrated result from Goemans states that the n-permutahedron has extended formulations with $O(n \log n)$ variables and constraints [23].

More generally, given a subgroup $G \sqsubseteq \mathbb{S}_n$, one can define the G-hedron as the convex-hull of the permutations in G. The technique used in [23] to upper bound the extension complexity of polytope $P(\mathbb{S}_n)$, which is based on the existence of sorting networks of size $O(n \log n)$ [1], has been used to show that polytopes corresponding to certain families of groups have small extension complexity. This includes polytopes corresponding to the alternating group [38], and to finite reflection groups [27, 28, 26, 11]. Nevertheless, techniques to prove non-trivial upper bounds on the extension complexity of polytopes associated with general permutation groups based on structural properties of these groups are still lacking. We note that a trivial upper bound of $|G|$ can be obtained from the fact that the extension complexity of a polytope is upper bounded by its number of vertices. Nevertheless, $|G|$ may have up to $n! = 2^{\Omega(n \log n)}$ elements.

In this section, by combining our main theorem (Theorem 5) with a connection established in [33] between the grammar complexity of a given formal language $L \subseteq [n]^r$ (for $n, r \in \mathbb{N}_+$) and the extension complexity of the polytope $P(L)$ associated with L, we obtain a new approach for proving upper bounds on the extension complexity of a general permutation group $G \sqsubseteq \mathbb{S}_n$ based on structural parameters of graphs embedding G (Theorem 14). We note that Theorem 14 is more general in the sense that it also can be used to upper bound the extension complexity of polytopes associated with cosets of G.

Let $\mathcal{X}$ be a set of real variables. A *real vector* over $\mathcal{X}$ is a function $v : \mathcal{X} \to \mathbb{R}$. We let $\mathbb{R}^{\mathcal{X}}$ be the set of all real vectors over $\mathcal{X}$. Given a set $W = \{v_1, \ldots, v_r\}$ of real vectors, the convex-hull of W is the set $\mathrm{conv}(W) = \{\sum_{i=1}^{r} \alpha_i v_i \ : \ \alpha_i \geq 0, \sum_{i=1}^{r} \alpha_i = 1\}$ of all convex linear-combinations of vectors in W. A subset $P \subseteq \mathbb{R}^{\mathcal{X}}$ is a *polytope over* $\mathcal{X}$ if $P = \mathrm{conv}(W)$ for some finite set W of real vectors over $\mathcal{X}$. For each such a polytope P, there is a finite set $\mathcal{E}$ of linear inequalities over $\mathcal{X}$ such that P is the set of vectors in $\mathbb{R}^{\mathcal{X}}$ which satisfy each inequality in $\mathcal{E}$.

Let $\mathcal{X}$ and $\mathcal{Y}$ be sets of real variables with $\mathcal{X} \cap \mathcal{Y} = \emptyset$. We say that a $(\mathcal{X} \cup \mathcal{Y})$-polytope Q is an *extended formulation* of P if there exists a linear projection $\rho : \mathbb{R}^{\mathcal{X}\cup\mathcal{Y}} \to \mathbb{R}^{\mathcal{X}}$ such that $P = \rho(Q)$. The *extension complexity of* P, denoted by $\mathrm{xc}(P)$, is defined as the least number of inequalities necessary to define an extended formulation of P.

For each $n \in \mathbb{N}_+$, we let $[n]^{\mathcal{X}}$ be the set of real vectors over $\mathcal{X}$ whose coordinates are chosen from the set $[n]$. For $r \in \mathbb{N}_+$, let $w = w_1 \dots w_r$ be a string in $[n]^r$, and let $\mathcal{X}_r = \{x_1, ..., x_r\}$ be an ordered set of real variables. We let $\hat{w} : \mathcal{X}_r \to [n]$ be the real vector over $\mathcal{X}_r$ which sets $\hat{w}_i = w_i$ for each $i \in [r]$. Given a subset $L \subseteq [n]^r$, the $\mathcal{X}_r$-polytope associated with L is defined as $P(L) = \mathrm{conv}(\{\hat{w} \ : \ w \in L\})$.

The following theorem, proved in [33], relates the grammar complexity of a subset $L \subseteq [n]$ with the extension complexity of the polytope $P(L)$.

▶ **Theorem 13** ([33]). *Let $\mathfrak{G}$ be a context-free grammar such that $\mathcal{L}(\mathfrak{G}) \subseteq [n]^r$ for some $n, r \in \mathbb{N}_+$. Then the extension complexity of the polytope $P(\mathcal{L}(\mathfrak{G}))$ is upper bounded by $|\mathfrak{G}|^{O(1)}$. A system of inequalities defining $P(\mathcal{L}(\mathfrak{G}))$ can be constructed in time $|\mathfrak{G}|^{O(1)}$.*

If G is a subgroup of $\mathbb{S}_n$, and $\beta \in G$, then we let $P(\beta \circ G) := P(\mathrm{str}(\beta \circ G))$ be the polytope associated with the coset $\beta \circ G$. The following theorem, which is the main result of this section, follows by a direct combination of Theorem 5 with Theorems 13.

▶ **Theorem 14.** *Let $G \sqsubseteq \mathbb{S}_n$, and suppose that G is embeddable on a graph X with m vertices ($m \geq n$), maximum degree Δ, and treewidth k. Then, for each $\beta \in \mathbb{S}_n$, the extension complexity of the polytope $P(\beta \circ G)$ is at most $2^{O(k\Delta \log \Delta)} \cdot m^{O(k)}$. Additionally, given X and β, a system of inequalities defining $P(\beta \circ G)$ can be constructed in time $2^{O(k\Delta \log \Delta)} \cdot m^{O(k)}$.*

6 Complexity Theoretic Tradeoffs

In 1969 Babai and Bouwer showed independently that any subgroup G of $\mathbb{S}_n$ can be embedded in a connected graph with $O(n + |G|)$ vertices [5, 12]. Note that $|G|$ can be as large as $n!$. Classifying which groups can, or cannot, be embedded in connected graphs with a much smaller number of vertices is an important problem in algebraic graph theory [8]. Indeed, constructing an explicit class of groups with superpolynomial graph embedding complexity is still an open problem, although a conjecture of Babai states that the alternating group $\mathbb{A}_n$ has graph embedding complexity $2^{\Omega(n)}$ [7]. We note that Liebeck has shown that any graph whose automorphism group is *isomorphic* to the alternating group $\mathbb{A}_n$ (as an abstract group) has an exponential number of vertices [30]. Nevertheless this result does not extend to the graph embedding setting.

In this section we use our main theorem to establish a trade-off between the index of a subgroup G of $\mathbb{S}_n$, and structural parameters of graphs embedding G. In particular, for several classes of graphs $\mathcal{X}$, this trade-off can be used to prove lower bounds on the $\mathcal{X}$-embedding complexity of subgroups of $\mathbb{S}_n$ of small index (i.e index up to 2^{cn} for some small constant c). We start by stating the following immediate observation.

▶ **Observation 15.** *Let $\mathfrak{G}_1$ and $\mathfrak{G}_2$ be context-free grammars. Then there is a context-free grammar $\mathfrak{G}_1 \cup \mathfrak{G}_2$ of size $O(|\mathfrak{G}_1| + |\mathfrak{G}_2|)$ such that $\mathcal{L}(\mathfrak{G}_1 \cup \mathfrak{G}_2) = \mathcal{L}(\mathfrak{G}_1) \cup \mathcal{L}(\mathfrak{G}_2)$.*

The next lemma states that the symmetric grammar complexity of a group G is at most the index of a subgroup H in G times the symmetric grammar complexity of H. Recall that if G is a group and H is a subgroup of G, then the index of H in G is defined as $\mathcal{I}_G(H) = \frac{|G|}{|H|}$.

▶ **Lemma 16 (♠).** *Let $H \sqsubseteq G \sqsubseteq \mathbb{S}_n$. Then $\mathrm{sgc}(G) \leq \mathcal{I}_G(H) \cdot \mathrm{sgc}(H)$.*

It has been shown in [18] (Theorem 30) that the language $str(\mathbb{S}_n)$ cannot be represented by context-free grammars of polynomial size. Since $str(\mathbb{S}_n)$ is invariant under permutation of coordinates, i.e., $str(\mathbb{S}_n) = \mathrm{Perm}(str(\mathbb{S}_n), \alpha)$ for any permutation $\alpha \in \mathbb{S}_n$, we have that the symmetric context-free complexity of $str(\mathbb{S}_n)$ is exponential.

▶ **Theorem 17** (Theorem 30 of [18]). $\mathrm{sgc}(\mathbb{S}_n) \geq 2^{\Omega(n)}$.

Now, by combining Theorem 17 with Lemma 16 (for $G = \mathbb{S}_n$), we have the following immediate corollary.

▶ **Corollary 18.** *Let H be a subgroup of $\mathbb{S}_n$. Then* $\mathrm{sgc}(H) \geq \frac{2^{\Omega(n)}}{\mathcal{I}_{\mathbb{S}_n}(H)}$.

By combining Theorem 5 with Corollary 18, we have a trade-off between the index of a group H, and the number of vertices, the treewidth and the maximum degree of a graph embedding H. Below, we write $\exp_2(x)$ to denote 2^x.

▶ **Theorem 19 (♠).** *There exist positive real constants c_1, c_2 and c_3 such that for large enough n, and each subgroup H of $\mathbb{S}_n$, if H is embeddable in a connected graph with m vertices, maximum degree Δ and treewidth k, then*

$$m \geq \exp_2\left(\frac{c_1 n - c_2 k\Delta \log \Delta - c_3 \log \mathcal{I}_{\mathbb{S}_n}(H)}{k}\right).$$

As a corollary of Theorem 19, we get the following lower bound stating that subgroups of $\mathbb{S}_n$ with small index (i.e. index at most 2^{cn} for some small constant c) cannot be embedded in graphs of treewidth $o(n/\log n)$, maximum degree $o(n/\log n)$ and a polynomial number of vertices.

▶ **Corollary 20.** *Let $\mathcal{X}$ be a class of connected graphs of treewidth $o(n/\log n)$ and maximum-degree $o(n/\log n)$. Then there is a function $f \in \omega(1)$, and a constant $c \in \mathbb{R}$, such that for each sufficiently large n, each subgroup G of $\mathbb{S}_n$ of index $\mathcal{I}_{\mathbb{S}_n}(G) \leq 2^{cn}$ has $\mathcal{X}$-embedding complexity at least $n^{f(n)}$.*

For classes of graphs of treewidth n^ε (for $\varepsilon < 1$), and maximum degree $o(n/\log n)$, Theorem 5 implies exponential lower bounds on the embedding complexity of groups of small index (i.e. index at most 2^{cn} for some small constant c).

▶ **Corollary 21.** *Let $\mathcal{X}$ be a class of connected graphs of treewidth n^ε (for $\varepsilon < 1$) and maximum-degree $o(n/\log n)$. Then there exist constants $c, c' \in \mathbb{R}$, such that for each sufficiently large n, each subgroup G of $\mathbb{S}_n$ of index $\mathcal{I}_{\mathbb{S}_n}(G) \leq 2^{cn}$ has $\mathcal{X}$-embedding complexity at least $2^{c' n^{1-\varepsilon}}$.*

In particular, for some small $c, c' \in \mathbb{R}$, the graph embedding complexity of subgroups of $\mathbb{S}_n$ of index at most $2^{c' n}$ is lower bounded by $2^{c\sqrt{n}}$ for any minor closed class of graphs of maximum degree $o(n \log n)$. Note that these classes of graphs have treewidth at most $\sqrt{n}$.

7 Conclusion and Open Problems

In this work, we have established new connections between three complexity measures for permutation groups: embedding complexity parameterized by treewidth and maximum-degree, symmetric grammar complexity and extension complexity. In particular, we have shown that groups that can be embedded in graphs of small treewidth and degree have small symmetric grammar complexity and small extension complexity. These results can also be used to translate strong lower bounds on the symmetric grammar complexity or on the

extension complexity of a group $G \sqsubseteq \mathbb{S}_n$ into lower bounds on the embedding complexity of G. In particular, using this approach, we have shown that subgroups $G \sqsubseteq \mathbb{S}_n$ of sufficiently small index have superpolynomial embedding complexity on classes of graphs of treewidth $o(n/\log n)$ and maximum degree $o(n/\log n)$.

Below, we state some interesting open problems related to our work.

▶ **Problem 22.** *Construct an explicit family of groups* $\{G_n\}_{n\in\mathbb{N}_+}$ *with superpolynomial graph embedding complexity, that is to say, such that* $\mathrm{gec}(G_n) = n^{\Omega(1)}$.

In particular, it is not known if the graph embedding complexity of the alternating group $\mathbb{A}_n$ is superpolynomial. Note that the graph embdding complexity of the symmetric group $\mathbb{S}_n$ is n, which is witnessed by K_n, the complete graph with vertex set $\{1,\ldots,n\}$.

▶ **Problem 23.** *Does the alternating group* $\mathbb{A}_n$ *have superpolynomial graph embedding complexity?*

The n-alternahedron polytope $P(\mathbb{A}_n)$ is the polytope associated with the alternating group $\mathbb{A}_n$. The technique used in [23] to prove an $O(n\log n)$ upper bound on the extension complexity of the n-permutahedron $P(\mathbb{S}_n)$ was generalized in [38] to show that the extension complexity of the n-alternahedron is $O(n\log n)$.

References

1 Miklós Ajtai, János Komlós, and Endre Szemerédi. Sorting in logn parallel steps. *Combinatorica*, 3(1):1–19, 1983.
2 Peter RJ Asveld. Generating all permutations by context-free grammars in Chomsky normal form. *Theoretical Computer Science*, 354(1):118–130, 2006.
3 Peter RJ Asveld. Generating all permutations by context-free grammars in Greibach normal form. *Theoretical Computer Science*, 409(3):565–577, 2008.
4 David Avis and Hans Raj Tiwary. On the extension complexity of combinatorial polytopes. *Mathematical Programming*, 153(1):95–115, 2015.
5 László Babai. Representation of permutation groups by graphs. *Colloquia Mathematica Societatis Janos Bolyai*, 4:55–80, 1969.
6 László Babai. Automorphism groups of graphs and edge-contraction. *Discrete Mathematics*, 8(1):13–20, 1974.
7 László Babai. On the abstract group of automorphisms. In HNV Temperley, editor, *Combinatorics*, volume 52 of *London Mathematical Society Lecture Note Series*, pages 1–40. Cambridge University Press, 1981.
8 László Babai. Automorphism groups, isomorphism, and reconstruction. In *Handbook of Combinatorics*, pages 1447–1540. North-Holland–Elsevier, 1995.
9 Egon Balas and William Pulleyblank. The perfectly matchable subgraph polytope of a bipartite graph. *Networks*, 13(4):495–516, 1983.
10 Francisco Barahona. On cuts and matchings in planar graphs. *Mathematical Programming*, 60(1-3):53–68, 1993.
11 Aharon Ben-Tal and Arkadi Nemirovski. On polyhedral approximations of the second-order cone. *Mathematics of Operations Research*, 26(2):193–205, 2001.
12 IZ Bouwer. Section graphs for finite permutation groups. *Journal of Combinatorial Theory*, 6(4):378–386, 1969.
13 Gábor Braun, Samuel Fiorini, Sebastian Pokutta, and David Steurer. Approximation limits of linear programs (beyond hierarchies). *Mathematics of Operations Research*, 40(3):756–772, 2015.
14 Kevin King Hin Cheung. *Subtour Elimination Polytopes and Graphs of Inscribable Type*. PhD thesis, University of Waterloo, 2003.

15 Michele Conforti, Marco Di Summa, Friedrich Eisenbrand, and Laurence A Wolsey. Network formulations of mixed-integer programs. *Mathematics of Operations Research*, 34(1):194–209, 2009.

16 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

17 Michel M Deza and Monique Laurent. *Geometry of Cuts and Metrics*. Springer, 1997.

18 Keith Ellul, Bryan Krawetz, Jeffrey Shallit, and Ming-wei Wang. Regular expressions: New results and open problems. *Journal of Automata, Languages and Combinatorics*, 9(2/3):233–256, 2004.

19 Yuri Faenza and Volker Kaibel. Extended formulations for packing and partitioning orbitopes. *Mathematics of Operations Research*, 34(3):686–697, 2009.

20 Yuval Filmus. Lower bounds for context-free grammars. *Information Processing Letters*, 111(18):895–898, 2011.

21 Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald De Wolf. Exponential lower bounds for polytopes in combinatorial optimization. *Journal of the ACM*, 62(2):17, 2015.

22 Ian Glaister and Jeffrey Shallit. A lower bound technique for the size of nondeterministic finite automata. *Information Processing Letters*, 59(2):75–77, 1996.

23 Michel X Goemans. Smallest compact formulation for the permutahedron. *Mathematical Programming*, 153(1):5–11, 2015.

24 Massimiliano Goldwurm, Beatrice Palano, and Massimo Santini. On the circuit complexity of random generation problems for regular and context-free languages. In Afonso Ferreira and Horst Reichel, editors, *Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science, STACS 2001*, volume 2010 of *LNCS*, pages 305–316. Springer, 2001.

25 Hermann Gruber, Markus Holzer, and Simon Wolfsteiner. On minimal grammar problems for finite languages. In Mizuho Hoshi and Shinnosuke Seki, editors, *Proceedings of the 22nd International Conference on Developments in Language Theory, DLT 2018*, volume 11088 of *LNCS*, pages 342–353. Springer, 2018.

26 James E Humphreys. *Reflection Groups and Coxeter Groups*, volume 29 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1992.

27 Volker Kaibel and Andreas Loos. Branched polyhedral systems. In Friedrich Eisenbrand and F. Bruce Shepherd, editors, *Proceedings of the 14th International Conference on Integer Programming and Combinatorial Optimization, IPCO 2010*, volume 6080 of *LNCS*, pages 177–190. Springer, 2010.

28 Volker Kaibel and Kanstantsin Pashkovich. Constructing extended formulations from reflection relations. In Michael Jünger and Gerhard Reinelt, editors, *Facets of Combinatorial Optimization*, pages 77–100. Springer, 2013.

29 Volker Kaibel and Stefan Weltge. A short proof that the extension complexity of the correlation polytope grows exponentially. *Discrete & Computational Geometry*, 53(2):397–401, 2015.

30 Martin W Liebeck. On graphs whose full automorphism group is an alternative group or a finite classical group. *Proceedings of the London Mathematical Society*, 3(2):337–362, 1983.

31 Antonio Molina Lovett and Jeffrey O. Shallit. Optimal regular expressions for permutations. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming, ICALP 2019*, volume 132 of *LIPIcs*, pages 121:1–121:12. Schloss Dagstuhl, 2019.

32 Stefan Mengel. Arithmetic branching programs with memory. In Krishnendu Chatterjee and Jirí Sgall, editors, *Proceedings of the 38th International Symposium Mathematical Foundations of Computer Science, MFCS 2013*, volume 8087 of *LNCS*, pages 667–678. Springer, 2013.

33 Gilles Pesant, Claude-Guy Quimper, Louis-Martin Rousseau, and Meinolf Sellmann. The polytope of context-free grammar constraints. In Willem Jan van Hoeve and John N. Hooker, editors, *Proceedings of the 6th International Conference on Integration of AI and OR Techniques*

in Constraint Programming for Combinatorial Optimization Problems, CPAIOR 2009, volume 5547 of *LNCS*, pages 223–232, 2009.

34 Sebastian Pokutta and Mathieu Van Vyve. A note on the extension complexity of the knapsack polytope. *Operations Research Letters*, 41(4):347–350, 2013.

35 William R Pulleyblank and Bruce Shepherd. Formulations for the stable set polytope. In Giovanni Rinaldi and Laurence A. Wolsey, editors, *Proceedings of the 3rd Conference on Integer Programming and Combinatorial Optimization, IPCO 1993*, pages 267–279, 1993.

36 Neil Robertson and Paul D. Seymour. Graph minors. X. Obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, 1991.

37 Thomas Rothvoß. Some 0/1 polytopes need exponential size extended formulations. *Mathematical Programming*, 142(1-2):255–268, 2013.

38 Stefan Weltge. Erweiterte Formulierungen für das Alternaeder. Diplomarbeit, University of Magdeburg, Germany, 2012.

39 Mihalis Yannakakis. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43(3):441–466, 1991.