Symmetry and Complexity

László Babai[1,2,3], Robert Beals[1,4],
Pál Takácsi-Nagy[1]

Abstract

We examine the effect of symmetry on the complexity of Boolean functions and find a remarkably tight hierarchy. Generalizing the fact that all symmetric Boolean functions belong to (nonuniform) TC^0, we find that the complexity of the class of Boolean functions admitting a given group of symmetries is essentially determined by a single parameter of that group.

Let G be a permutation group acting on the set of n Boolean variables. Let $\mathcal{F}(G)$ denote the set of Boolean functions on n variables which are invariant under G. Let $\mathcal{G}$ denote a sequence $G_n \leq \mathrm{Sym}(\Omega_n)$ of groups ($|\Omega_n| = n$). We say that the language $L \subseteq \{0,1\}^*$ belongs to the symmetry class $\mathcal{F}(\mathcal{G})$ if the indicator function of $L \cap \{0,1\}^n$ belongs to $\mathcal{F}(G_n)$ for every n.

Following Clote and Kranakis, we consider the parameter $s(G)$, the number of orbits of G on the set $\{0,1\}^n$. We show that (a) all functions in $\mathcal{F}(G)$ are computable by circuits of size, polynomial in $s(G)$ and depth, polynomial in $\log s(G)$; (b) there exist functions in $\mathcal{F}(G)$ which cannot be computed by circuits of size $\leq s(G)/(2 \log s(G))$.

While part (b) is obtained by straightforward counting, it demonstrates that part (a) is tight. The result in particular confirms the following conjecture of Clote and Kranakis: if $s(G_n)$ is polynomially bounded (where $G_n \leq S_n$) then $\mathcal{F}(\mathcal{G}) \subseteq NC$ (nonuniform). If in addition the groups are transitive, we prove that the left hand side is actually in TC^0.

[0]This research was partially supported by NSF Grants CCR 8710078 and CCR 9014562

[1]Department of Computer Science, University of Chicago, Chicago, Illinois 60637

[2]Eötvös University, Budapest, Hungary H-1088

[3]laci@cs.uchicago.edu

[4]beals@cs.uchicago.edu

24th ANNUAL ACM STOC - 5/92/VICTORIA, B.C., CANADA
© 1992 ACM 0-89791-512-7/92/0004/0438...$1.50

The proof of the main result involves nontrivial elementary asymptotic structure theory of permutation groups; and a delicate analysis (under new circumstances) of algorithmic techniques developed largely by E. M. Luks in the context of graph isomorphism testing.

In the context of isomorphism of sets under group action, uniform versions of our results are obtained.

1 Introduction

Let $\Omega = \{1, \ldots, n\}$ and let $\mathrm{Sym}(\Omega)$ denote the symmetric group acting on Ω, i.e. the group of all permutations of Ω. Any permutation $\sigma \in \mathrm{Sym}(\Omega)$ naturally induces an action on $\{0,1\}^n$: for $x = (x_1, \ldots, x_n) \in \{0,1\}^n$, let $x^\sigma = (x_{1'}, \ldots, x_{n'})$, where $i' = i^{\sigma^{-1}}$. We also obtain an action on the set of all Boolean functions $f : \{0,1\}^n \to \{0,1\}$ by setting $f^\sigma(x) = f(x^\sigma)$. The *automorphism group* $\mathrm{Aut}(f)$ of a Boolean function f consists of all σ such that $f^\sigma = f$.

The *symmetric* Boolean functions f are defined by the property $\mathrm{Aut}(f) = \mathrm{Sym}(\Omega)$. It is well known that these functions belong to a low non-uniform complexity class: non-uniform TC^0. (For a more detailed study of symmetric functions, see [FKPS].)

This simple observation suggests the possibility of deeper links between the automorphism group and the complexity of a Boolean function. A systematic study of this connection was initiated by P. Clote and E. Kranakis [CK].

Let us define the *symmetry class* $\mathcal{F}(G)$ as the set of Boolean functions f such that $G \leq \mathrm{Aut}(f)$ (the functions *invariant* under G). Let $\mathcal{G}$ denote a sequence $G_n \leq \mathrm{Sym}(\Omega_n)$ of groups ($|\Omega_n| = n$). We say that the language $L \subseteq \{0,1\}^*$ belongs to the symmetry class $\mathcal{F}(\mathcal{G})$ if the indicator function of $L \cap \{0,1\}^n$ belongs to $\mathcal{F}(G_n)$ for every n.

Observing that groups having polynomially bounded index in the symmetric group S_n have extremely simple structure, Clote and Kranakis [CK] prove that for such sequences $\mathcal{G}$, $\mathcal{F}(\mathcal{G}) \subset TC^0$ (nonuniform). They also point out that the really relevant parameter of the

groups G_n should be their number of orbits in their induced action on $\{0,1\}^n$.

For $G \leq \mathrm{Sym}(\Omega)$, let $s(G)$ denote the number of *string-orbits* of G, i.e. the number of orbits of the induced G-action on $\{0,1\}^n$.

Clote and Kranakis state the following beautiful conjecture [CK, Conj. 33, p. 588] (Conj. 8, p. 65 in the conference paper). This conjecture has now become a corollary to our main result.

Theorem 1.1 *Let $\mathcal{G}$ be a sequence of permutation groups with polynomially bounded number of string-orbits (i.e. $s(G_n) = n^{O(1)}$). Then the symmetry class $\mathcal{F}(\mathcal{G})$ is in non-uniform NC.*

Indeed we prove more generally that the complexity of the symmetry class $\mathcal{F}(G)$ is essentially determined by the number $s(G)$ of string-orbits.

Theorem 1.2 *Let $G \leq \mathrm{Sym}(\Omega)$. Then (a) all functions in the symmetry class $\mathcal{F}(G)$ can be computed by circuits of size, polynomial in $s(G)$, and depth, polynomial in $\log(s(G))$. (b) There exist functions in $\mathcal{F}(G)$ which cannot be computed by circuits of size $\leq s(G)/(2\log s(G))$.*

The proof of part (b) is straightforward counting (cf. [Sha], [Lup]). It shows, however, that, curiously, unless such a trivial reason prevents the given symmetry class to be recognized by circuits of a given size, they will immediately be recognizable with this size *in a highly parallel manner.*

In the case of transitive groups, we have an even stronger result.

Theorem 1.3 *Let $\mathcal{G} = \{G_n\}$ be a sequence of transitive permutation groups with polynomially bounded number of string-orbits (i.e. $s(G_n) = n^{O(1)}$). Then the symmetry class $\mathcal{F}(\mathcal{G})$ is in non-uniform TC^0.*

These questions are closely related to the following generalization of the Graph Isomorphism Problem (Luks [Lu1]):

String-isomorphism under G-action. Let $G \leq \mathrm{Sym}(\Omega)$ ($\Omega = \{1,\ldots,n\}$) and $x, y \in \{0,1\}^n$. Decide whether or not x and y are G-equivalent (i.e. belong to the same G-orbit). If so, find the coset of $\mathrm{Aut}(x)$ consisting of those $\sigma \in G$ for which $x^\sigma = y$.

Luks has shown that this problem (assuming G is given by a list of generators) is solvable in polynomial time for certain classes of groups, but in general the problem is at least as hard as graph isomorphism [Lu1]. On the other hand, at this moment it seems hopeless to parallelize Luks's solutions even in the case of 2-groups; the latter would be equivalent to NC-solvability of isomorphism for trivalent graphs, a major open problem in the area [LM].

Our main auxiliary result pertains to this problem and gives an upper bound on its *uniform* complexity.

Theorem 1.4 *Given a permutation group G by a list of generators, one can solve the string-isomorphism problem under G-action by uniform circuits of size polynomial in $s(G)$ and depth polynomial in $\log(s(G))$.*

The proof of this result requires the NC-algorithm for basic permutation group manipulation [BLS] and therefore it is not elementary (recall that the [BLS] algorithm depends on consequences of the classification of finite simple groups). This difficulty can be avoided if we are interested in the inherently nonuniform problems posed by our main theorem; in this case we can ascertain that only a small number of groups are encountered throughout the algorithm, and we may assume (nonuniformly) that everything we need about these groups is wired into our circuits in advance.

Even this elementary proof, however, requires nontrivial estimates from asymptotic group theory [Ba2,3], [Py], [BaP].

2 Preliminaries

For $s = s(n)$ define $NC(s)$ to be the class of languages computable by nonuniform boolean circuits of size $s^{O(1)}$ and depth $(\log s)^{O(1)}$. $NC^1(s)$ is defined analogously (depth $= O(\log s)$). With a common abuse of terminology, we will use $NC(s)$ and $NC^1(s)$ to refer to function classes as well. We will construct $NC(s(G))$ circuits to compute the class $\mathcal{F}(G)$ of functions admitting G as a subgroup of their automorphism group.

For $u \in \Omega$ the stabilizer G_u is defined as $G_u = \{\sigma \in G \mid u^\sigma = u\}$. If $B \subseteq \Omega$ we denote by $G_{\{B\}} = \{\sigma \in G \mid B^\sigma = B\}$, the *setwise* stabilizer of B. We use G_B to denote the *pointwise* stabilizer: $G_B = \bigcap_{u\in B} G_u$. If $G = G_{\{B\}}$ we say that B is G-stable. For $B \subseteq \Omega$, $\sigma \in \mathrm{Sym}(\Omega)$, if $B^\sigma = B$ we use σ^B to denote the restriction of σ to B. If B is G-stable, we set $G^B = \{\sigma^B \mid \sigma \in G\}$. If G is the full symmetric or the alternating group then we call G a *giant*. $\Delta \subset \Omega$ is a *block of imprimitivity* if $\forall \sigma \in G$, either $\Delta^\sigma = \Delta$ or $\Delta^\sigma \cap \Delta = \emptyset$. G is *primitive* if the only blocks of imprimitivity are Ω and the singletons. For $g_1, g_2 \in G$ $[g_1, g_2] = g_1 g_2 g_1^{-1} g_2^{-1}$ denotes the *commutator* of g_1 and g_2. For $M, N \leq G$ $[M, N] = \langle [m, n] \mid m \in M, n \in N\rangle$ will stand for the *mutual commutator* of M and N.

A *structure forest* F for a permutation group $G \leq \mathrm{Sym}(\Omega)$ is a forest on which G acts as automorphisms such that the leaves form the permutation domain Ω

and the roots correspond to orbits. Each vertex v can be identified with a block of imprimitivity $B(v)$ of the G-action on Ω, consisting of all leaves below v. $\mathcal{B}(v)$ will denote the set of blocks corresponding to the children of the node v, i.e. $\mathcal{B}(v) = \{B(u) \mid u \text{ is a child of } v\}$. We refer to $|B(v)|$ as the *size* of the node v. $L(v) \leq \mathrm{Sym}(B(v))$ and $H(v) \leq \mathrm{Sym}(\mathcal{B}(v))$ shall denote the action of G_v on $B(v)$ and $\mathcal{B}(v)$ respectively. A node v is *primitive* if $H(v)$ is primitive. A node v is called a *giant node* if $H(v)$ is a giant. We call a structure forest F *primitive* if all of its nodes are primitive. If the group is transitive then the structure forest is a *structure tree.* Here we deviate somewhat from the standard terminology, because our *primitive structure forest(tree)* is what is usually called a *structure forest(tree).* If G is a group with t orbits, its structure forest consists of t trees $T_1, \ldots, T_t$. $\mathcal{L}_i$ denotes the set of nodes on level i; $\mathcal{L}_0$ consists of the roots. We call $k_{i,j} = |\mathcal{B}(v)|$ (number of children) the *degree* of $\mathcal{L}_i$ in T_j. The subscript j will be omitted if G is transitive. K_i will stand for the pointwise stabilizer of level i. Clearly $K_i \lhd G$ since K_i is the kernel of the G-action on $\mathcal{L}_i$. The quotient group G/K_i acts faithfully on $\mathcal{L}_i$.

We call a group $G \hookrightarrow Q = H_1 \times \ldots \times H_r$ a *subdirect product* of the groups H_i if for every i, $\pi_i(G) = H_i$, where $\pi_i : Q \rightarrow H_i$ is the i^{th} projection. Our analysis will depend on the following (folklore) lemma (cf. Scott[Sc], Luks[Lu3]).

Lemma 2.1 *Let $G \hookrightarrow Q = H_1 \times \ldots \times H_r$ be a subdirect product, where each H_i is a simple group. Then after some rearrangement of the factors, we may write*

$$Q = (H_1 \times \ldots \times H_{i_1}) \times (H_{i_1+1} \times \ldots \times H_{i_2}) \times \ldots \times (H_{i_{r-1}+1} \times \ldots \times H_{i_r})$$

such that $H_{i_j+1} \simeq H_{i_j+2} \simeq \ldots \simeq H_{i_{j+1}}$ for all $0 \leq j \leq r-1$ (where $i_0 = 0$), and

$$G = \mathrm{diag}(H_1 \times \ldots \times H_{i_1}) \times \mathrm{diag}(H_{i_1+1} \times \ldots \times H_{i_2}) \times \ldots \times \mathrm{diag}(H_{i_{r-1}+1} \times \ldots \times H_{i_r}),$$

(i.e after appropriate identifications, G consists of the elements of the form $(\alpha, \ldots, \alpha)(\beta, \ldots, \beta)\ldots(\kappa, \ldots, \kappa)$). □

Let L and H be permutation groups on sets A and B, respectively. We define the *wreath product* of L by H, denoted $L \wr H$, as the group of all permutations $\pi \in \mathrm{Sym}(A \times B)$ such that for $a \in A, b \in B$, we have $(a,b)^\pi = (a^{\sigma_b}, b^\tau)$, where for each $b \in B$, σ_b is an arbitrary element of L, and $\tau \in H$. For different b's the permutations σ_b are chosen independently, so $|L \wr H| = |L|^{|B|}|H|$. For $b \in B$ let $A_b = A \times \{b\} \subseteq A \times B$. The A_b's form a system of imprimitivity for $L \wr H$. Let T be a structure tree for $G = L \wr H$ acting on $\Omega = A \times B$ such that for some i the nodes in $\mathcal{L}_i$ correspond to the A_b's. T has the following properties:

1. For $v \in \mathcal{L}_i, L(v) \simeq L$.
2. $G/K_i \simeq H$.
3. For $v \in \mathcal{L}_i, K_i \simeq L(v)^{|\mathcal{L}_i|}$.

Conversely, for transitive G the third condition implies that G is the wreath product of $L(v)$ by G/K_i. In any case $K_i \leq L(v)^{|\mathcal{L}_i|}$ and $G \leq L(v) \wr (G/K_i)$.

We quote a classical result of Hölder (cf. Sec.5. 3. [KM]).

Theorem 2.2 (Hölder) *Provided that $n \neq 2, 6$, $Aut(S_n) \simeq S_n$.* □

We represent the coset $G\sigma$ by the element σ together with a set of generators for G. Let a string $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ be fixed. For $a, b \in \Omega$ let $a \sim b$ denote that $x_a = x_b$. We use the following handy notation introduced by Luks: For $B \subseteq \Omega$ and $K \subseteq \mathrm{Sym}(\Omega)$ define $C_B(K)$ to be $\{\sigma \in K \mid b \sim b^\sigma \text{ for all } b \in B\}$. We quote three basic observations, made by Luks.

Proposition 2.3

(a) $C_{B_1 \cup B_2}(K) = C_{B_1}(C_{B_2}(K))$

(b) $C_B(K_1 \cup K_2) = C_B(K_1) \cup C_B(K_2)$

(c) *If $G \leq \mathrm{Sym}(\Omega)$ and $B \subseteq \Omega$ is G-stable, then $C_B(G)$ is a subgroup of G, and for $\sigma \in \mathrm{Sym}(\Omega)$, $C_B(G\sigma)$ is either empty or a right coset of $C_B(G)$.*

The *string stabilizer problem* is the problem of determining the coset $C_\Omega(G\sigma)$ for some $G \leq \mathrm{Sym}(\Omega), \sigma \in \mathrm{Sym}(\Omega), x \in \{0,1\}^n$. For the purposes of recursive algorithms, we consider the more general problem of determining $C_B(G\sigma)$, where B is a G-stable set. Formally,
Input: $G \leq \mathrm{Sym}(\Omega), \sigma \in \mathrm{Sym}(\Omega), x \in \{0,1\}^n$, and B a G-stable subset of Ω.
Output: $C_B(G\sigma)$.

The *nonuniform string-stabilizer problem* is the problem of determining $C_B(G\sigma)$ by nonuniform circuits with input $\sigma \in \mathrm{Sym}(\Omega)$. G, B, and x are wired into the circuit.

As shown by Luks, the string-stabilizer problem is equivalent to the "string-isomorphism problem" stated in the Introduction. Indeed,

Proposition 2.4 (Luks)
The string-isomorphism problem is TC^0-equivalent to the string-stabilizer problem.

Proof: Let $x, y \in \{0,1\}^n$, and $\sigma \in \mathrm{Sym}(\Omega)$ such that $y^\sigma = x$ (if such a σ exists it can be found in TC^0). Then the set $\{\tau \in G \mid x^\tau = y\}$ is exactly $(C_B(G\sigma))\sigma^{-1}$. Note that if x, G and B are fixed, σ depends only on y. Conversely, let $\sigma \in \mathrm{Sym}(\Omega)$, $x \in \{0,1\}^n$. Set $y = x^{\sigma^{-1}}$, and read the above equality backward. □

Proposition 2.5 *An $NC(s(G))$ solution of the string-stabilizer problem implies Theorem 1.2.*

Proof: For $f \in \mathcal{F}(G)$, computing f reduces in (nonuniform) $TC^0(s(G))$ to the string-isomorphism problem, as the set of strings on which f is 1 is a union of G-orbits. The circuit recognizes these orbits in parallel, and computes the OR of the result. To recognize a single orbit, we wire an orbit representative x into the circuit, which on input y computes a $\sigma \in \mathrm{Sym}(\Omega)$ with $y^\sigma = x$, and outputs 1 if $C_\Omega(G\sigma)$ is nonempty. □

The rest of the paper concerns the string stabilizer problem.

3 Structure of permutation groups with large alternating groups acting on their blocks

Let G be a group. $H \leq G$ is a *complement* to $K \leq G$ if $H \cap K = 1$ and $HK = G$. For $G \leq \mathrm{Sym}(\Omega)$ and a system of imprimitivity $\mathcal{B}$ of G we say that $\sigma \in G$ is *clean* if for every $B \in \mathcal{B}$, if $B^\sigma = B$ then σ acts trivially on B. A *clean subgroup* consists of clean elements. Let K be the kernel of the G-action on $\mathcal{B}$. We say that H is a *clean complement* to K if H is a complement to K and it is clean w.r.t. $\mathcal{B}$. Clearly $H \simeq G/K$. The observation below is straightforward from the definition of a clean complement.

Observation 3.1 *If H is a clean complement to K, w.r.t. the system of imprimitivity $\mathcal{B}$ then for every $B \in \mathcal{B}$ we have $G^B_{\{B\}} = K^B$.* □

The following result is proved in [Ba3] (cf. [BKL, Sec. 10]).

Lemma 3.2 *Let $G \leq \mathrm{Sym}(\Omega)$ have a depth two structure tree T such that $H(\mathrm{root}) = A_{k_0}$ and $k_0 \geq 4k_1$. Then K_1 has a clean complement.*

Proof: Let $\mathcal{L}_1 = \{v_1, \ldots, v_{k_0}\}$ be level 1 of T. For $\tau \in G$ let $\overline{\tau}$ denote the action of τ on $\mathcal{L}_1$. By Bertrand's postulate there is a prime p between k_1 and $k_0/2$. Take $\pi \in G$ such that $\overline{\pi}$ is a p-cycle. As $k_1 < p$ there is an m such that p does not divide m and π^m is clean. W.l.o.g. $m = 1$, and π permutes $v_1, \ldots, v_p$ cyclically and fixes v_i and each of its children for $i = p+1, \ldots, k_0$. Similarly, there exists a clean $\pi' \in G$ such that $\overline{\pi'} = (v_p, \ldots, v_{2p-1})$. Thus $\sigma := [\pi, \pi']$ is a clean 3-cycle. Let σ_i be a conjugate of σ with $\overline{\sigma}_i = (v_i, v_{i+1}, v_{k_0})$. (Note that conjugates of clean elements are clean.) It is easy to see that for k_0 odd, the group generated by $\sigma_1, \sigma_3, \sigma_5, \ldots, \sigma_{k_0-2}$ is a clean complement to K_1. If k_0 is even then take $A = \langle \sigma_1, \sigma_3 \rangle_{v_{k_0}} \simeq A_4$. Now $A, \sigma_4, \sigma_6, \ldots, \sigma_{k_0-2}$ generate a clean complement to K_1. □

Note that a clean complement does not necessarily exist if $H(\mathrm{root}) = S_{k_0}$, even if k_0 is as large as $|\Omega|/2$, as the following example shows. Let $k_1 = 2$ and $G \simeq S_{k_0}$ and let $v_i(1), v_i(-1)$ denote the two children of node v_i on level 1 ($1 \leq i \leq k_0$). For each $\sigma \in S_{k_0}$ let $\sigma^* \in G$ be such that $\sigma^*(v_i) = v_{\sigma(i)}$ and $\sigma^*(v(j)) = v_{\sigma(i)}(\mathrm{sgn}(\sigma)j)$ ($j = 1, 2$; $1 \leq i \leq k_0$). Note that K_1 is the identity in this case and it does not have a clean complement, since each element of G which acts on $\mathcal{L}_1$ as an odd permutation performs a transposition on the fixed blocks.

A slight extension of the proof of Lemma 3.2 yields the following generalization.

Lemma 3.3 *Let $G \leq \mathrm{Sym}(\Omega)$ be a group with t orbits. Assume G has a structure forest consisting of depth two trees $T_1, \ldots, T_t$ such that*

1. $\min_j k_{0,j} \geq 4 \max_j k_{1,j}$.
2. $G/K_1 = A_{k_{0,1}} \times \ldots \times A_{k_{0,t}}$, *where $A_{k_{0,j}}$ acts on the $k_{0,j}$ children of the root of T_j.*

Then K_1 has a clean complement H. □

Assume that $G \leq \mathrm{Sym}(\Omega)$ is a transitive permutation group satisfying the conditions of Lemma 3.2. As K_1 has a clean complement we can identify Ω with the set $\mathcal{B}(\mathrm{root}) \times B$, where $|B| = |B(w)|$ for each node $w \in \mathcal{B}(\mathrm{root})$. We identify $B(w)$ with $\{w\} \times B$ such that for each $b \in B$, $\mathcal{B}(\mathrm{root}) \times \{b\}$ is an orbit of H. For $\sigma \in K_1$, let $\sigma_w := \sigma^{B(w)}$, and for $\sigma \in \mathrm{Sym}(B)$, let $\widetilde{\sigma}$ act on $\Omega = \mathcal{B}(\mathrm{root}) \times B$ by permuting the second components. For $S \leq \mathrm{Sym}(B)$, let $S(w) = \{\widetilde{\sigma}_w \mid \sigma \in S\}$. Note that our groups $L(w)$ arise as $L(w) = L^w$ for some $L \leq \mathrm{Sym}(B)$. The following theorem appears in [Ba3].

Theorem 3.4 *Let $G \leq \mathrm{Sym}(\Omega)$ be a transitive group satisfying the conditions of Lemma 3.2. Let $L = L(w)$ for some $w \in \mathcal{B}(\mathrm{root})$. Then L has normal subgroups $M_0 \leq M_1$, and G has normal subgroups N_1 and N_0, $N_0 \leq N_1 \leq K_1$, such that*

1. $N_1 = ({M_1}^{w_1} \times \ldots \times {M_1}^{w_{k_0}}) \cap K_1$ *and* $N_0 = {M_0}^{w_1} \times \ldots \times {M_0}^{w_{k_0}}$.
2. ${N_1}^{\Omega \setminus \mathcal{B}(w_i)} = {M_1}^{w_1} \times \ldots {M_1}^{w_{i-1}} \times {M_1}^{w_{i+1}} \times \ldots \times {M_1}^{w_{k_0}}$.
3. $L/M_1 \simeq K_1/N_1$.
4. M_1/M_0 *is abelian.*
5. $(N_1/N_0) = \{(\sigma_1, \ldots, \sigma_{k_0}) \mid \sigma_i \in M_1/M_0, \prod_{i=1}^{k_0} \sigma_i = 1\}$.

Proof: *2*: Let $M_1 := \{\sigma \in L \mid \exists \sigma_3, \ldots, \sigma_{k_0} \in L : (1, \sigma, \sigma_3, \ldots, \sigma_{k_0}) \in K_1\}$. Clearly $M_1 \lhd L$.

Claim 3.5 $\sigma \in M_1 \iff (\sigma, \sigma^{-1}, 1, \ldots, 1) \in K_1$

Proof of Claim: ($\Leftarrow$) Conjugating $(\sigma, \sigma^{-1}, 1, \ldots, 1)$ by a proper element of H we get $(1, \sigma, \sigma^{-1}, 1, \ldots, 1) \in K_1$, so by the definition of M_1, $\sigma \in M_1$.
($\Rightarrow$) Assume that $\widehat{\sigma} = (1, \sigma, \sigma_3, \ldots, \sigma_{k_0}) \in K_1$. First we show that some element of the form $(1, \sigma, 1, \sigma'_4, \ldots, \sigma'_{k_0})$ also belongs to K_1. Indeed let $\tau = (v_1, v_2, v_3)$ and set

$$\varrho = [\tau, \widehat{\sigma}] = \tau^{-1}\widehat{\sigma}^{-1}\tau\widehat{\sigma} = (\sigma_3^{-1}, \sigma, \sigma^{-1}\sigma_3, 1, \ldots, 1).$$

Conjugating ϱ by an appropriate element of H we get ϱ' which has $(1, \sigma, 1)$ in the first three positions. Now $\varrho'' = [\varrho', \tau^{-1}] = (\sigma, \sigma^{-1}, 1, \ldots, 1)$. □
The Claim clearly implies *2*, using the definition of N_1 in *1*.

3: Let us consider the homomorphism $\varphi : K_1 \to L/N_1$ defined by:

$$\varphi(\sigma_1, \ldots, \sigma_{k_0}) = \sigma_2 M_1$$

It suffices to show that $\mathrm{Ker}(\varphi) = N_1$. Obviously $\mathrm{Ker}(\varphi) \geq N_1$. On the other hand suppose that $\pi \in \mathrm{Ker}(\varphi)$. This implies that $\pi_2 \in M_1$. We are going to prove that $\pi_1, \pi_3, \ldots, \pi_{k_0}$ are also in M_1. By the Claim there is a $\pi' \in K_1$ with $\pi' = (\pi_2, \pi_2^{-1}, 1, \ldots, 1)$. So $\pi\pi' = (\pi_1\pi_2, 1, \pi_3, \ldots, \pi_{k_0})$ so we infer that $\pi_1, \pi_3, \ldots, \pi_{k_0} \in M_1$.
Let $M_0 := \{\sigma \in L \mid (\sigma, 1, \ldots, 1) \in K_1\}$.

4: It is enough to prove that $M_0 \geq [M_1, M_1]$. Assume that $\sigma, \sigma' \in M_1$. This implies that $(\sigma, \sigma^{-1}, 1, \ldots, 1)$, $(\sigma', \sigma'^{-1}, 1, \ldots, 1)$ and the inverses of their conjugates $(\sigma^{-1}, 1, \sigma, 1, \ldots, 1)$, $(\sigma'^{-1}, 1, \sigma', 1, \ldots, 1)$ are in K_1. So the product of these elements is $([\sigma, \sigma'], 1, \ldots, 1) \in K_1$, which means that $[\sigma, \sigma'] \in M_0$.

5: Suppose $(\sigma_1, \ldots, \sigma_{k_0}) \in N_1/N_0$. (The σ_i are cosets of M_0.) As $\sigma_i \in M_1/M_0$ for each $1 \leq i \leq k_0 - 1$ there is a $\pi(i) \in N_1/N_0$ with $(\pi(i))_i = \sigma_i^{-1}$, $(\pi(i))_{k_0} = \sigma_i$ and $(\pi(i))_j = 1$ otherwise. So we have that

$$(\sigma_1, \ldots, \sigma_{k_0}) \prod_{i=1}^{k_0-1} \pi(i) = (1, \ldots, 1, \prod_{i=1}^{k_0} \sigma_i).$$

but then $\prod_{i=1}^{k_0} \sigma_i = M_0$ by the definition of M_0, proving *5*. □

4 Asymptotic estimates for permutation groups

In this section we review some inequalities on the number of string-orbits of a permutation group and bounds on the order of primitive permutation groups. The following is immediate:

Fact 4.1 *If $H \leq G$ then $s(G) \leq s(H) \leq s(G)|G : H|$.*

Let G be a transitive group acting on a set Ω, $|\Omega| = n$. Let $\{\Omega_1, \ldots, \Omega_k\}$ denote a system of imprimitivity of G with block size b ($1 \leq b \leq n$; $bk = n$). If L_i denotes the permutation group of degree b induced on Ω_i by the setwise stabilizer of Ω_i in G then the groups $L_i \simeq L$ are isomorphic transitive permutation groups.

Proposition 4.2 $\min(s(L) - 1, k) \leq \log s(G)$.

Proof: $G \leq L \wr S_k$, which has $\binom{k+s(L)-1}{k}$ string-orbits, so by Fact 4.1 $s(G) \geq \binom{k+s(L)-1}{k}$. This implies the Proposition. □
Since $b \leq s(L) - 1$, we obtain in particular:

Corollary 4.3 $\min(b, k) \leq \log s(G)$. □

Proposition 4.4 *Let Δ be a block of imprimitivity of G and $L = G^{\Delta}_{\{\Delta\}}$ (the permutation group induced by the setwise stabilizer of Δ on Δ). Then $s(L) \leq s(G)$.*

Proof: Since Δ is a block of G, for every nonempty $X, Y \subseteq \Delta$, if $X^\tau = Y$ for some $\tau \in G$ then $\tau \in G_{\{\Delta\}}$ and therefore $X^{\tau'} = Y$ for $\tau' = \tau^{\Delta} \in L$. So two subsets of Δ are in the same G-orbit if and only if they are in the same L-orbit. □

By using the following elementary estimate [Ba 2,3] we are able to avoid the use of the Classification of Finite Simple Groups.

Theorem 4.5 *Let G be a primitive permutation group of degree n not containing A_n. Then*

$$|G| < \exp(4\sqrt{n}\log^2 n).$$□

Let $c_2 \geq 8$ be a constant such that for all $x \geq c_2$, $2^{x/2} \geq \exp(4\sqrt{x}\log^2 x)$ and $x \log x < x^2/8$.

Lemma 4.6 *Let $G \leq \mathrm{Sym}(\Omega)$ and T be a structure forest for G. Assume that for a node $w \in T$ with $k = |B(w)|$, $c_2 \leq k \leq s(L(w)) \leq \log s(G)$ and $k! \geq s(G)$. Then $A_k \leq L(w)$.*

Proof: Suppose that $A_k \not\leq L(w)$. If $L(w)$ is primitive then by Theorem 4.5 $|L(w)| < \exp(4\sqrt{k}\log^2 k)$. On the other hand

$$k \log k \geq \log s(G) \geq s(L(w)) \geq \frac{2^k}{|L(w)|} \geq 2^{k/2},$$

a contradiction (assuming $k \geq c_2$). If $L(w)$ is imprimitive then $L(w) \leq S_m \wr S_l$ for some $m, l > 1$ with $ml = k$. So

$$\begin{aligned} k \log k &\geq \log s(G) \geq s(L(w)) \\ &\geq s(S_m \wr S_l) = \binom{m+l}{l} \geq \frac{(ml)^2}{8}, \end{aligned}$$

which is impossible for $ml = k \geq c_2$. □

The following result serves as the basis of the structure to be derived in Section 5. We sketch the proof in the Appendix.

Theorem 4.7 (BaP) *Let $G \leq \mathrm{Sym}(\Omega)$, $|\Omega| = n$, and F a primitive structure forest for G. Then for any $t \geq 1$, if F has no giant node of degree $> t$ then*

$$s(G) \geq 2^{n/(c_1 t)}$$

for some absolute constant c_1.□

5 Transitive Case: Structure Theory

Let $G \leq \mathrm{Sym}(\Omega)$ be transitive and T a primitive structure tree for G. By Theorem 4.7 T has a level, which we will call the *explosion level*, where for all nodes u, the group $H(u)$ contains an alternating group of degree at least $n/(c_1 \log s(G))$. Let us contract the levels below and above the explosion level to one level each (keeping the root separate). We obtain a depth three structure tree with $k_0 k_2 \leq c_1 \log s(G)$. So we conclude:

Theorem 5.1 *For every transitive $G \leq \mathrm{Sym}(\Omega)$ there exists a depth three structure tree T, such that $k_0 k_2 \leq c_1 \log s(G)$ and the nodes on level 1 are giants.* □

Definition: For w a node in T, we denote by $\psi_w : G_w \to \mathrm{Sym}(\mathcal{B}(w))$ the homomorphism obtained by restriction to $\mathcal{B}(w)$ i.e. $\psi_w(\sigma) = \sigma^{\mathcal{B}(w)}$.

Now we can state a refinement of Theorem 5.1.

Theorem 5.2 *Let $G \leq \mathrm{Sym}(\Omega)$ be transitive. If there exists a depth three structure tree with giants of degree $k_1 \geq \max(1 + k_0, 7)$ on level 1 then there exists another depth three structure tree T^* with $k_0^* \leq k_0$, $k_1^* = k_1$ such that the action of K_1^* on level 2 contains a full direct product of k_0^* copies of A_{k_1}.*

Proof: For $S \leq G$ set $\widetilde{S} = S/(S \cap K_2)$ (note $S \cap K_2 \lhd S$ as $K_2 \lhd G$). For each node u on level 1 we have that $A_{k_1} \leq \psi_u(\widetilde{G}_u)$. We need the following

Claim 5.3 *Let G be as above and u a node on level 1. Then $\psi_u(\widetilde{K}_1)$ still contains A_{k_1}.*

Proof of Claim: As $\psi_u(\widetilde{K}_1) \lhd \psi_u(\widetilde{G}_u)$ we have $A_{k_1} \cap \psi_u(\widetilde{K}_1) \lhd A_{k_1}$. For $k_1 \geq 5$, A_{k_1} is simple, so $A_{k_1} \cap \psi_u(\widetilde{K}_1)$ is either A_{k_1} or the identity. We show that the second case is impossible. Indeed $\psi_u(\widetilde{G}_u)/\psi_u(\widetilde{K}_1)$ is a quotient group of $\widetilde{G}_u/\widetilde{K}_1 \leq \mathrm{Sym}(k_0 - 1)$. On the other hand, $A_{k_1} \cap \psi_u(\widetilde{K}_1) = 1$ would imply $|\psi_u(\widetilde{G}_u) : \psi_u(\widetilde{K}_1)| \geq |A_{k_1}| = k_1!$, a contradiction because $k_0 < k_1$. □

We have that $\widetilde{K}_1 = K_1/K_2$ is a subdirect power of A_{k_1}. By Theorem 2.1, K_1/K_2 is a direct power of some $\mathrm{diag}(A_{k_1} \times \ldots \times A_{k_1})$. If the diagonal constituents are single A_{k_1}'s, we are done. If not, we regroup the nodes on levels 1 and 2. Let $D_1, \ldots, D_{k_0^*}$ be the partition of the nodes at level 1 corresponding to the diagonal constituents. Each D_i will be a new level 1 node. For nodes $u, v \in D_i$, the diagonal linking of $H(u)$ and $H(v)$ gives us a unique bijection $u_j \mapsto v_j$ for $1 \leq j \leq k_1$ between the children $u_1, \ldots, u_{k_1}$ of u and the children $v_1, \ldots, v_{k_1}$ of v, as by Hölder's theorem [KM p.43] the automorphism group of S_k or A_k is S_k for $k > 6$. We regroup the level 2 nodes as follows. For $1 \leq i \leq k_0^*$ and $1 \leq j \leq k_1$, let $B_{ij} = \bigcup_{u \in D_i} B(u_j)$.

Claim 5.4 *The B_{ij} form a system of imprimitivity.*

From the claim it follows that the regrouped tree has the desired properties.

Proof of Claim: It suffices to show that for any τ in G if $u^\tau = v$ and $(u')^\tau = v'$ for $u, v, u', v' \in D_1$ then $u_1^\tau = v_1$ implies $(u_1')^\tau = v_1'$. Assume the contrary. So $(u_1')^\tau = v_j'$ for some $j \neq 1$. Take a $\kappa \in K$ which fixes u_1 but moves u_j. The action of $\tau\kappa\tau^{-1}$ on $\mathcal{B}(u)$ and $\mathcal{B}(u')$ is different, so $(\tau\kappa\tau^{-1})^{\mathcal{B}(u) \cup \mathcal{B}(u')}$ is not a diagonal element but $\tau\kappa\tau^{-1}$ is in K_1, a contradiction. □

As a consequence of Theorem 5.2 we know that for every permutation group G there exists a depth three structure tree T either with $k_0, k_1, k_2 \leq c_2 \log s(G)$ ($c_2 \geq 4$) or with the following properties:

- $k_0 k_2 \leq c_1 \log s(G)$.
- $k_1 \geq 4 \log s(G)$.
- For every node u on level 1, $H(u)$ is a giant.
- K_1/K_2 contains $(A_{k_1})^{k_0}$.

Definitions: The *socle* of a finite group G denoted by $\mathrm{Soc}(G)$ is the subgroup generated by all minimal normal subgroups of G.

Theorem 5.5 *Let $G \leq \mathrm{Sym}(\Omega)$ and T a structure tree for G as above. If $k_2! \geq \max(s(G), 7!)$ and $k_1 > \max(4 \log s(G), 52)$ then $\mathrm{Soc}(K_2) = (A_{k_2})^{k_1 k_0}$.*

We need a calculation.

Proposition 5.6 *Let b, k, m be positive integers. If $b! \geq m \geq 2^b$, $b > 4$, $k \geq \max(b, 52)$ and*

$$m \geq \frac{\binom{2^b + k - 1}{k}}{b!}$$

then $k < 3 \log m$.

Proof: We consider three cases:

1. $2^b \geq 2k^2$. Then

$$m \geq \frac{\binom{2^b+k-1}{k}}{b!} \geq \left(\frac{2^b}{k}\right)^k \frac{1}{b!} \geq \frac{(2k)^k}{b^b} \geq \left(\frac{2k}{b}\right)^k \geq 2^k,$$

so in this case $k \leq \log m$.

2. $2k^2 > 2^b > 2k$. Then

$$m \geq \frac{\binom{2^b+k-1}{k}}{b!} \geq \left(\frac{2^b}{k}\right)^k \frac{1}{b!} \geq \frac{2^k}{b^b} > 2^{k/3},$$

assuming $k \geq 52$. Therefore $k < 3 \log m$.

3. $2k \geq 2^b$. Then

$$m \geq \frac{\binom{2^b+k-1}{2^b-1}}{b!} \geq \frac{(1+\frac{k}{2^b})^{2^b-1}}{b!} \geq \frac{(1.5)^{2^b-1}}{b!} \geq b!,$$

for $b \geq 5$, a contradiction. So under our conditions only the first two cases are possible. □

Proof of Theorem 5.5: Let w be a node on level 2. By Proposition 4.2, $s(L(w)) \leq \log s(G)$. From our assumption $k_2! = (|B(w)|)! \geq s(G)$, using Lemma 4.6 we can conclude that $A_{k_2} \leq L(w)$. Following an argument similar to that in the proof of Proposition 5.3 it can be shown that $\psi_w((K_1)_w)$ still contains a giant. Let φ_2 denote the restriction homomorphism $K_1 \rightarrow \mathrm{Sym}(\mathcal{L}_2)$. As $K_1/K_2 \geq (A_{k_1})^{k_0}$ we can set $\widehat{K}_1 := \varphi_2^{-1}((A_{k_1})^{k_0})$ and $\widehat{K}_2 = \widehat{K}_1 \cap K_2$. Obviously $\widehat{K}_1 \lhd K_1$.

Claim 5.7 *For $w \in \mathcal{L}_2$, $\psi_w(K_2) \geq \psi_w(\widehat{K}_2) \geq A_{k_2}$.*

Proof of Claim: $K_1/\widehat{K}_1$ is an elementary abelian 2-group as for all $\tau \in K_1/\widehat{K}_1$ and for all v on level 1, $(\tau^2)^{\mathcal{B}(v)}$ is an even permutation of $\mathcal{B}(v)$. Moreover $\psi_w((K_1)_w)/\psi_w((\widehat{K}_1)_w)$ is a quotient group of a subgroup of $K_1/\widehat{K}_1$, so it is also an elementary abelian 2-group. So $\psi_w((\widehat{K}_1)_w) \geq A_{k_1}$. Since $(1/4)k_1 > \log s(G) \geq s(L(w)) \geq k_2$, by Lemma 3.3 $\widehat{K}_2$ has a clean complement in $\widehat{K}_1$. Using Observation 3.1 we conclude that $\psi_w((\widehat{K}_1)_w) = \psi_w(\widehat{K}_2) \geq A_{k_2}$ for each w on level 2. □

Note that $S := \mathrm{Soc}(K_2)$ is a subdirect power of A_{k_2}. Then by Lemma 2.1, S is a direct power of some $\mathrm{diag}(A_{k_2} \times \ldots \times A_{k_2})$. But for nodes on level 2, their corresponding A_{k_2}'s belonging to the same diagonal constituent is a G-invariant relation. The only nontrivial system of imprimitivity on level 2 is the set $\{B(u) \mid u \text{ is on level } 1\}$. So either $S = (A_{k_2})^{k_1 k_0}$ or $S \simeq (A_{k_2})^{k_0}$ is a direct product of k_0 subgroups of the form $\mathrm{diag}(A_{k_2} \times \ldots \times A_{k_2})$, by Theorem 2.2. We show that the latter case cannot happen. If it did then for any v on the first level we would have that, $A_{k_1} \times A_{k_2} \leq L(v) \leq S_{k_1} \times S_{k_2}$. So

$$(s(G))^2 \geq s(L(v)) = \frac{\binom{2^{k_2}+k_1-1}{k_1}}{k_2!}.$$

However this inequality is impossible by Proposition 5.6 with $m = s(G)$, $k = k_1$, and $b = k_2$. □

Corollary 5.8 *Let G satisfy the conditions of Theorem 5.5. Then*

$$(A_{k_2} \wr A_{k_1})^{k_0} \leq K_2 \leq (S_{k_2} \wr S_{k_1})^{k_0}.$$ □

6 Luks's Halving Algorithm

We describe here an algorithm due to Luks for computing $C_B(G\sigma)$, where B is G stable. Let $X_1, \ldots, X_k$ be a system of imprimitivity for G. These blocks will be fixed at the start of the algorithm: the algorithm makes recusive calls to itself, and B will always be a union of some of the blocks. We assume that we have a subroutine for computing $C_{X_i}(G\sigma)$, which runs in parallel in time ℓ with w processors.

Suppose $B = \bigcup_{i \in I} X_i$ If $|I| = 1$, we use the subroutine for $C_{X_i}(G\sigma)$. Otherwise we *halve* the set I. Let $I = I_1 \cup I_2$ with $|I_1| = \lfloor |I|/2 \rfloor$ and $|I_2| = \lceil |I|/2 \rceil$. Let $B_1 = \bigcup_{i \in I_1} X_i$ and let $B_2 = \bigcup_{i \in I_2} X_i$. Let $G^* = G_{\{B_1\}}$ and $\{\tau_1, \ldots, \tau_r\}$ be coset representatives for G^* in G. Use the Halving Algorithm recursively in parallel to find $C_{B_1}(C_{B_2}(G^*\tau_j\sigma))$ for all $j \in \{1, \ldots, r\}$.

In the nonuniform model, we can suppose that for each of a system of string-orbit representatives, the groups $C_B(G)$ encountered, and the coset representatives τ_i are known in advance. Then to determine the coset $C_B(G\sigma)$ it suffices to find a single i for which $C_{B_1}(C_{B_2}(G^*\tau_i\sigma))$ is nonempty.

In the uniform model, G^* can be found in NC using the algorithms of [BLS]. The τ_i can be found in $NC(2^{|I|})$ by considering the action of G on the power set of I. After calculating the $C_{B_1}(C_{B_2}(G^*\tau_i\sigma))$, we use the following Lemma to paste them together to make $C_B(G\sigma)$.

Lemma 6.1 *Suppose $G^* \leq \mathrm{Sym}(\Omega)$, $\tau_1, \ldots, \tau_r$ are given, and $\bigcup_i G^*\tau_i$ is known to be a single coset of some unknown group G. Then we can find this coset in $NC(r + |\Omega|)$.*

Proof: The coset we want is $G\tau_1$, so it suffices to find generators for G. We have $G\tau_1 = \bigcup_i G^*\tau_i$, so $G = \bigcup_i G^*\tau_i\tau_1^{-1}$. Therefore G is generated by the generators of G^* together with the quotients $\tau_i\tau_1^{-1}$. We can reduce the number of generators to $O(|\Omega|)$ in NC using the algorithms of [BLS]. □

The time and processor bound for the subroutine to compute $C_{X_i}(G\sigma)$, together with an easy induction on $|I|$, give us the following Lemma.

Lemma 6.2 *The halving algorithm runs in time polynomial in* $(k\ell + \log|\Omega|)$ *with polynomial in* $(4^k w + |\Omega|)$ *many processors.*□

The halving algorithm can be used as the subroutine to compute $C_{X_i}(G\sigma)$ by picking a new block system, such as the one below $X_1, \ldots X_k$ in the structure tree. If the structure tree can be collapsed to a bounded number of levels of degree $O(\log s(G))$, then the halving algorithm can be used at every level of the structure tree in $NC(s(G))$. In other cases, we will need a separate algorithm for the lower part of the structure tree.

7 Algorithm: the transitive case

This section represents the "hard part" of our algorithm. We begin with some simple observations:

Observation 7.1 *If* $H \leq G$ *and* $|G:H|$ *is polynomial in* $s(G)$ *then the string-stabilizer problem for* G *reduces in* $NC^1(s(G))$ *to the string-stabilizer problem for* H. *This reduction can be done uniformly if a system of coset representatives for* $|G:H|$ *can be calculated.*

Observation 7.2 $C_B(G\sigma)$ *can be found in* $NC(|G^B| + |\Omega|)$.

Indeed, we enumerate G^B in parallel and determine which elements of $(G\sigma)^B$ stabilize the coloring of B.

Observation 7.3 *If* $B_1, \ldots, B_k$ *is a system of imprimitivity for* G, *with* $L = G^{B_1}_{\{B_1\}}$ *such that* G *is the wreath product of* L *by* A_k, *then* $C_B(G\sigma)$ *can be found in* $NC(k|L| + |\Omega|)$. *If in addition* $L = A_{|B_1|}$, *then* $C_B(G\sigma)$ *is essentially a counting problem, which can be solved in* TC^0.

The above two observations require a comment. Once we find a subgroup of G whose action on B is the same as $C_B(G)$, to find $C_B(G)$ we need to add generators for G_B (the pointwise stabilizer of B in G) to our generating set. This is trivial in the nonuniform model, as G_B can be calculated in advance. In the uniform model, we need the algorithms of [BLS] to construct G_B in NC.

Theorem 7.4 *Let* $G \leq \mathrm{Sym}(\Omega)$ *be transitive. Then (a) all functions in the symmetry class* $\mathcal{F}(G)$ *can be computed by* $NC(s(G))$ *circuits. (b) There exist functions in* $\mathcal{F}(G)$ *which cannot be computed by circuits of size less than* $s(G)/(2\log s(G))$.

Proof: Part (b) follows from straightforward counting (cf [Sha],[Lup]). Part (a) follows immediately from

Theorem 7.5 *For* $G \leq \mathrm{Sym}(\Omega)$ *transitive,* $C_\Omega(G\sigma)$ *can be calculated uniformly in* $NC(s(G))$.

Proof of Theorem 7.5: Assume that $s(G)$ is known. First we construct a depth three structure tree T for G, such that $k_0 k_2 \leq c_1 \log s(G)$ and the nodes at level 1 are giants, as in Theorem 5.1. If $k_1 \geq \max(1 + k_0, 7)$ then modify T so that the action of K_1 on level 2 contains a full direct product of k_0 copies of A_{k_1}, as in Theorem 5.2. We consider three cases:

1. $k_1 \leq c_2 \log s(G)$. In this case we use the Halving Algorithm at each level of the structure tree.

2. $k_2! > s(G)$. In this case k_0 is $O(\log\log s(G))$, so $|G : K_1| \leq s(G)$ and it suffices to solve the string-stabilizer problem for K_1. By Corollary 5.8 in this case $(A_{k_2} \wr A_{k_1})^{k_0} \leq K_1 \leq (S_{k_2} \wr S_{k_1})^{k_0}$, so the string-stabilizer problem is in TC^0 by Observation 7.3.

3. $k_2! \leq s(G)$. In this case we use the halving algorithm on the top level of the tree. When get down to a block $B(u)$ with u at level 1, we know by Theorem 3.4 that the group $L(u)$ acting on $B(u)$ has a subgroup H which acts as a clean A_{k_1} on the nodes at level 2. Also, by Theorem 3.4, the subgroup $K_2{}^{B(u)}$ which fixes the nodes at level 2 has a subgroup N_1 which acts as a full direct product inside $k_1 - 1$ of the blocks on level 2. Since

$$\begin{aligned} |L(u) : HN_1| &\leq 2|K : N_1| = 2|L : M_1| \\ &\leq 2|L| \leq 2k_2! \leq 2s(G) \end{aligned}$$

it suffices to solve the set stabilizer problem for HN_1. Let v be a child of u on level 2. $|HN_1 : (HN_1)_v| = k_1 \leq s(G)$, so it suffices to solve the set stabilizer problem for $G^* = (HN_1)_v$. Let $B_1 = B(v)$ and $B_2 = B(u) \setminus B(v)$. Then $(G^*)^{B_2}$ is a full wreath product of M_1 by A_{k_1}, and $|M_1| \leq k_2! \leq s(G)$, so $C_{B_2}(G^*\sigma)$ can be found in $NC(s(G))$ by Observation 7.3. Since $|B_1|! \leq s(G)$, $C_{B_1}(C_{B_2}(G^*\sigma))$ can also be found in $NC(s(G))$ by Observation 7.2.

In the uniform model, $s(G)$ is not known. We can, however, calculate a lower bound s for $s(G)$ such that assuming $s(G) = s$ does not contradict all three of the above cases. The algorithm will then run in $NC(s)$. The calculation of s is discussed in Section 10.□

8 Reduction of the intransitive case

The intransitive case can be reduced to the transitive case using a variant of another algorithm of E. M. Luks [Lu1] as follows: To find $C_\Omega(G\sigma)$, let Δ be the largest orbit of G on Ω. Use the algorithm for the transitive case to find $C_\Delta(G\sigma)$, and recursively find

$C_{\Omega\setminus\Delta}(C_\Delta(G\sigma))$. Note that in the nonuniform model, we already know the groups $C_\Delta(G)$ and $C_{\Omega\setminus\Delta}(C_\Delta(G))$ in advance.

Lemma 8.1 $s((C_\Delta(G))^{\Omega\setminus\Delta}) \leq s(G)$.

Proof: By reordering, assume that Δ is an initial segment of Ω. Let x' denote $(x_1, \ldots, x_{|\Delta|})$. Let $y_1, y_2 \in \{0,1\}^{\Omega\setminus\Delta}$. Then y_1 and y_2 are in different orbits of $C_\Delta(G)$ if and only if $x'y_1$ and $x'y_2$ are in different orbits of G, as for $\sigma \in G$, $(x'y_1)^\sigma = x'y_2$ implies $\sigma \in C_\Delta(G)$.□

Lemma 8.2 $|\Delta| \geq |\Omega|/\log s(G)$.

Proof: G can have at most $\log s(G)$ orbits, so the largest, Δ, must contain at least a $(1/\log s(G))$ fraction of the points in Ω. □

Lemma 8.3 *The number of recursive calls is* $O((\log n)(\log s(G)))$.

Proof: By Lemma 8.1, $s(G)$ does not increase during the recursive calls, and by Lemma 8.2, $|\Omega|$ is decreased by a $(1 - 1/\log s(G))$ factor with each recursive call.□

We now have proven the following

Theorem 8.4 *Let $G \leq \mathrm{Sym}(\Omega)$. Then (a) all functions in the symmetry class $\mathcal{F}(G)$ can be computed by $NC(s(G))$ circuits. (b) There exist functions in $\mathcal{F}(G)$ which cannot be computed by circuits of size $\leq s(G)/(2\log s(G))$.*

In fact, we have

Theorem 8.5 *For all groups $G \leq \mathrm{Sym}(\Omega)$, $C_\Omega(G\sigma)$ can be calculated uniformly in $NC(s(G))$.*

This clearly implies Theorem 1.4.

9 The Algorithms

Below we give pseudocode for the main string stabilizer algorithm.

STRINGSTAB$[G, \sigma, B, x]$
Input: $G \leq \mathrm{Sym}(\Omega)$, $\sigma \in \mathrm{Sym}(\Omega)$, B a G-stable subset of Ω, $x \in \{0,1\}^{|\Omega|}$.
Output: $C_B(G\sigma)$
/* This algorithm makes $O((\log s(G))(\log|\Omega|))$ recursive calls to itself. */

Step 1: *Let $B_1 \subseteq B$ be the largest orbit of G on B, and let $B_2 = B \setminus B_1$.*

Step 2: *Find* $G^1\sigma_1 =$ **TRANS-STAB**$[G, \sigma, B_1]$.

Step 3: *Recursively find* $G^2\sigma_2 =$ **STRINGSTAB**$[G^1, \sigma_1, B_2, x]$.

Step 4: *Output* $G^2\sigma_2$.

The following algorithm deals with the transitive case.

TRANS-STAB$[G, \sigma, B]$
Input: $G \leq \mathrm{Sym}(\Omega)$,$\sigma \in \mathrm{Sym}(\Omega)$, $B \subseteq \Omega$ an orbit of G.
Output: $C_B(G\sigma)$
/* $s(G)$ is assumed to be known. */

Step 1: *Construct a primitive structure tree T' for G^B.*

Step 2: *Construct a depth 3 structure tree T by collapsing the levels of T' above and below the largest giant.*
/* By theorem 4.7 we will have $k_0 k_2 \leq c_1 \log s(G)$. */

Step 3: **if** $k_1 \geq \max(1 + k_0, 7)$ **then** *modify T so that the action of K_1 on level 2 contains the full direct product ${A_{k_1}}^{k_0}$.*
/* This is done by regrouping the nodes on levels 1 and 2 as in Theorem 5.2. */

Step 4: **if** $k_1 < c_2 \log s(G)$ **then** *output* **HALVE**$[G, \sigma, B, T, 1,$ **true**$]$ *and* **halt**.
/* In this case all three levels of T have degree less than $c_2 \log s(G)$, so the Halving Algorithm used all the way to the bottom level of T runs in $NC(s(G))$. */

Step 5: **if** $k_2! > s(G)$ **then** *Calculate $\tau_1, \ldots, \tau_r$ coset representatives for K_1 in G. Output $\bigcup_i C_B(K_1\tau_i\sigma)$, and* **halt**.
/* In this case $r = |G : K_1| < s(G)$. K_1 contains $(A_{k_2} \wr A_{k_1})^{k_0}$ by Theorem 5.5. so the string stabilizer problem for K_1 can be solved in NC by Observation 7.3. */

Step 6: *Output* **HALVE**$[G, \sigma, B, T, 1,$ **false**$]$.
/* In this case we use the Halving Algorithm on the top level of T, and use the subroutine **BOTTOM** on the bottom two levels. In this case $k_2! \leq s(G)$. */

Next we describe the repeated halving procedure as applied in Steps 4 and 6 above. The parameter ℓ describes the current level in the structure tree. The parameter *bot* is **true** if the halving algorithm is to be used at all levels of the tree. Otherwise, the procedure **BOTTOM** is used for the bottom two levels of the tree.

HALVE$[G, \sigma, B, T, \ell, bot]$
Input: $G \leq \mathrm{Sym}(\Omega)$, $\sigma \in \mathrm{Sym}(\Omega)$, B a G-stable subset of Ω, T a structure tree for a supergroup of G, and ℓ a level of T such that $B = \bigcup_{i\in I} B(u_i)$ for some nodes u_i at level ℓ of T, and *bot* a boolean flag indicating whether

the halving algorithm is to be used recursively all the way to the leaves of T.
Output: $C_B(G\sigma)$
/* We use a subroutine **BOTTOM** for the bottom two levels of T unless the flag *bot* is set. */

Step 1: **if** $|I| = 1$ **then**
if *bot* **then**
if $\ell = 3$ **then** *Output* $C_B(G\sigma)$
else HALVE$[G, \sigma, B, T, \ell + 1, bot]$
else BOTTOM$[G, \sigma, B]$.
/* In the case $|I| = 1$, *bot* = **true**, and $\ell = 3$, the output $C_B(G\sigma)$ is either $G\sigma$ or $\emptyset$ depending on whether $b \sim b^\sigma$ where $B = \{b\}$. */

Step 2: *Select* $\{I_1, I_2\}$ *a partition of* I *with* $|I_1| = \lfloor \frac{|I|}{2} \rfloor$. *Let* $B_1 = \bigcup_{i \in I_1} B(u_i)$ *and* $B_2 = B \setminus B_1$.
/* This partition may be fixed in advance. */

Step 3: *Calculate* $G^* = G_{\{B_1\}}$, *and* $\{\tau_1, \ldots, \tau_r\}$ *coset representatives for* $G : G^*$.

Step 4: *For each* $1 \leq j \leq r$ *calculate* $G^1\sigma_j =$ **HALVE**$[G^*, \tau_j\sigma, B_1, T, \ell, bot]$.

Step 5: *Output* $\bigcup_{j \leq r}$**HALVE**$[G^1, \sigma_j, B_2, T, \ell, bot]$.

Finally, we describe the routine used in Step 6 of **TRANS-STAB** below the top level. (On the top level we use halving.)

BOTTOM$[G, \sigma, B]$
Input: $G \leq \mathrm{Sym}(\Omega)$, $\sigma \in \mathrm{Sym}(\Omega)$, B a G-stable subset of Ω.
Output: $C_B(G\sigma)$
/* G^B is assumed to have a depth 2 structure tree T with an alternating group H acting on level 1 as a clean complement to K_1. This algorithm runs in $NC(k_0(k_1!)+|\Omega|)$, and is used on the bottom two levels of the structure tree in the case $k_2! \leq s(G)$. */

Step 1: *Construct* N_1 *as in Theorem 3.4.*
/* $N_1 \leq K_1$ acts as a full direct product on any $k_1 - 1$ of the blocks at level 1, and $|K_1 : N_1| \leq k_1!$. */

Step 2: *Select* u *a node at level* 1. *Let* $B_2 = B(u)$ *and* $B_1 = B \setminus B_2$.

Step 3: *Calculate* $\tau_1, \ldots, \tau_r$ *a system of coset representatives for* $(HN_1)_u$ *in* HK_1.

Step 4: *Find, for each* $1 \leq i \leq r$, $G^*\sigma_i = C_{B_1}((HN_1)_u\tau_i\sigma)$.
/* This can be done in $NC(k_0(k_1!) + |\Omega|)$ by Observation 7.3 as $(HN_1)_u$ acts as a full wreath product on B_1. */

Step 5: *Find and output* $\bigcup_i C_{B_2}(G^*\sigma_i)$.
/* This can be done in $NC(k_0(k_1!) + |\Omega|)$ by Observation 7.2. */

10 Uniform vs. nonuniform

To prove our main results (Theorems 1.1–1.3), we need the nonuniform version of Theorem 1.4 only. Below we indicate how to avoid the use of the NC algorithm [BLS] in this case.

Proposition 10.1 *G and x determine a set of $O(n)$ groups which are the only groups encountered by the algorithm.*

Proof: G and x completely determine the groups encountered during the algorithm, as $C_B(G\sigma)$ is empty or a coset of $C_B(G)$.

If we are using the Halving Algorithm down to level ℓ of the structure tree, then the set of halvings corresponds to a binary tree with the leaves being the nodes at level ℓ of the structure tree. The Halving Algorithm can be thought of as performing a depth first traversal of this *halving tree*, each node of which is associated with a single group as follows: Associate the input group G with the root. The group G^* which fixes the current halving is the input to the left child of the root. (The phase of the algorithm corresponding to the left child works simultaneously with all of the cosets of G^* in G.) The output group of the left child is the input group of the right child (which again corresponds to a phase of the algorithm which works in parallel on many cosets of this group). Therefore, while the number of cosets encountered is large, the number of subgroups is linear in the number of points at level ℓ of the structure tree. From this, it follows that the total number of subgroups encountered by the string-stabilizer algorithm is linear in $|\Omega|$.□

Recall that in the nonuniform model, G and x are wired into the circuit. Therefore any operations which need to be performed on G, such as finding coset representatives and generators for a subgroup and constructing clean complements, can be done in advance. Thus we avoid using the [BLS] algorithm.

Next we indicate hitherto ignored details, required in the *uniform* model, of the proof of Theorem 1.4.

In the uniform model, generators for G are part of the the input to the algorithm, so we need to perform permutation group manipulations in NC, using the algorithms of [BLS]. To construct the structure tree, we recognize giants and detect diagonal linking. In the Halving Algorithm, we construct pointwise stabilizers of sets, and we reduce large generating sets. In case 3 of Theorem 7.5, we construct clean complements.

Also, $s(G)$ is not known. Only the algorithm for the transitive case makes reference to $s(G)$. We do not need $s(G)$ to construct the depth three structure tree, as we simply construct a primitive structure tree and collapse the levels above and below the largest giant. Once this is done, we know by Theorem 4.7 that $s(G) \geq 2^{k_2 k_0/c_1}$. In addition, if $(A_{k_2} \wr A_{k_1})^{k_0} \not\leq K_1$, then by Corollary 5.8 we know $s(G) \geq \min(k_2!, 2^{k_1/4})$. So we can calculate a lower bound s for $s(G)$ as follows:

1. Initialize s to $2^{k_2 k_0/c_1}$.
2. If $(A_{k_2} \wr A_{k_1})^{k_0} \not\leq K_1$ then let $s = \max(s, \min(k_2!, 2^{k_1/4}))$.

We then run the algorithm as if $s(G) = s$. By construction of s, one of the cases of Theorem 7.5 will hold. The algorithm will run in $NC(s)$, which is acceptable as $s(G) \geq s$.

11 Transitive Case : TC^0 result

In this section we prove a stronger result in the nonuniform model in the case that G is transitive and $s(G)$ is polynomial in n.

Theorem 11.1 *Let $\mathcal{G} = \{G_n\}$ be a sequence of transitive permutation groups with polynomially bounded number of string-orbits (i.e. $s(G_n) = n^{O(1)}$). Then the symmetry class $\mathcal{F}(\mathcal{G})$ is in non-uniform TC^0.*

This is an easy consequence of the following:

Theorem 11.2 *If $\mathcal{G}$ is a family of transitive permutation groups G_n acting on an n element set Ω_n, and $s(G_n) \leq n^c$ for some positive constant c, then for all $x \in \{0,1\}^n$ there is a TC^0 circuit computing (on input σ) $C_\Omega(G\sigma)$.*

Note that in the nonuniform model, finding a single element of $C_\Omega(G\sigma)$ is equivalent to calculating $C_\Omega(G\sigma)$, since $C_\Omega(G)$ is known in advance. We need the the following generalization of Observation 7.1:

Observation 11.3 *If $H \leq G$ and $s(H)$ is polynomial in n then the string-stabilizer problem for G reduces to the string-stabilizer problem for H in (nonuniform) AC^1.*

In the case when $n^c \geq s(G)$ the following stronger version of Proposition 4.2 holds:

Proposition 11.4 *If $s(G) = n^c, c \geq 0$ then $\min(s(L)-1, k) \leq 2c$.*

Proof: By Proposition 4.2 $k \leq c \log n$. By Fact 4.1, $s(G) \geq s(L \wr S_k) = \binom{k+s(L)-1}{k}$. As this estimate is symmetric in k and $s(L)-1$, we may assume that $k = \min(s(L), k)$. For n sufficiently large, $n/(c \log n)^2 \geq \sqrt{n}$. Therefore,

$$\begin{aligned} n^c &\geq s(G) \geq \binom{k+s(L)-1}{k} \geq \binom{s(L)}{k} \\ &\geq \binom{n/k}{k} \geq \left(\frac{n}{k^2}\right)^k \geq n^{k/2}. \end{aligned}$$

So $k \leq 2c$. □

Corollary 11.5 *If $s(G) = n^c$ then k_0, $k_2 \leq 2c$.*

Proof: Proposition 11.4 and Theorem 5.1. □

Proof of theorem 11.2: By Theorem 5.2 there is a depth three structure tree T for G such that the action of K_1 on level 2 contains ${A_{k_1}}^{k_0}$. Let $\varphi : K_1 \to {S_{k_1}}^{k_0}$ be the action of K_1 on level 2. Let $G^* = \varphi^{-1}({A_{k_1}}^{k_0})$, and let F be the structure forest for G^* consisting of the bottom two levels of T. Applying Lemma 3.3 to G^*, we obtain a clean complement H to K_1^*. Note that $s(H) = \binom{k_1+2^{k_2}-1}{2^{k_2}-1}^{k_0}$ is polynomial in n, as k_2 and k_0 are bounded by the constant $2c$. By Observation 11.3 it suffices to show that the string-stabilizer problem for H is in TC^0. This follows from Observation 7.3.□

Acknowledgements. The first author is grateful to Peter Clote for calling the Clote-Kranakis conjecture to his attention. He owes particular gratitude to Gene Luks who taught him how to "divide and conquer" permutation groups.

References

[Atk] M.D. Atkinson: An algorithm for finding the blocks of a permutation group, *Math. Comp.* **29** (1975), pp. 911–913.

[Ba1] L. Babai: On the order of uniprimitive permutation groups, *Annals of Math.* **113** (1981), 553–568.

[Ba2] L. Babai: On the order of doubly transitive permutation groups, *Inventiones Math.* **65** (1982), 473–484.

[Ba3] L. Babai: *Permutation Groups, Coherent Configurations, and Graph Isomorphism*, D. Sc. Thesis (in Hungarian), Budapest 1984.

[BaP] L. Babai, L. Pyber: Permutation groups with exponentially many orbits on the power set, J Comb. Theory, Series A, to appear.

[BKL] L. Babai, W. M. Kantor, E. M. Luks: Computational complexity and the classification of finite simple groups, *in: Proc. 24th FOCS*, 1983, pp. 162–171.

[BLS] L. Babai, E. M. Luks, Á. Seress, Permutation groups in *NC*, *in: Proc. 19th ACM STOC*, 1987, pp. 409-420.

[CK] P. Clote, E. Kranakis: Boolean functions, invariance groups, and parallel complexity, *SIAM J. Comp.* **20** (1991), 553–590. (Preliminary version: *Proc. 4th IEEE Symp. Stucture in Complexity Theory*, 1989, pp. 55–66.)

[FKPS] R. Fagin, M. Klawe, N. Pippenger, L. Stockmeyer: Bounded depth, polynomial size circuits for symmetric functions, *Theoret. Comp. Sci.* **36** (1985), 239–250.

[Ha] M. Hall, Jr.: *The Theory of Groups*, Macmillan, New York 1959.

[Kn] D. E. Knuth: Efficient representation of perm groups, *Combinatorica* **11** (1991), pp. 57–68.

[Lu1] E. M. Luks: Isomorphism of graphs of bounded valence can be tested in polynomial time, *J. Comp. Sys. Sci.* **25** (1982), 42–65.

[Lu2] E. M. Luks: Parallel Algorithms for Permutation Groups and Graph Isomorphism, *in: Proc. 27th FOCS*, 1986, pp. 292–302.

[Lup] O. B. Lupanov: On a method of circuit synthesis, *Izvestia VUZ Radiofizika* **1** (1958), 120–140.

[LM] E. M. Luks, P. McKenzie: Fast parallel computation with permutation groups, *Proc. 27th IEEE FOCS* (1985), 505–514.

[MC] P. McKenzie, S. A. Cook: The parallel complexity of the abelian permutation group membership problem, *in: Proc. 24th FOCS*, 1983, pp. 154–161.

[Py] L. Pyber: The orders of doubly transitive groups, elementary estimates, to appear

[Sha] C. E. Shannon: The synthesis of two-terminal switching circuits, *Bell Syst. Tech. J.* **28** (1949), 59–98.

[Sim] C. C. Sims, Some group theoretic algorithms, *in: Lecture Notes in Math.* **697** (1978), pp. 108-124.

[We] I. Wegener, *The Complexity of Boolean Functions*, Wiley–Teubner 1987, Chap. 4.

[Wi] H. Wielandt: *Finite Permutation Groups*, Acad. Press, New York 1964.

Appendix

Proof of the Babai-Pyber Theorem

We now sketch a proof of Theorem 4.7[BaP].

Proof: As $s(G) \geq \prod_i s(G^{\Delta_i})$ where $\{\Delta_1, \ldots, \Delta_m\}$ are the orbits of G, it suffices to prove the theorem for transitive groups G. Also, we may restrict ourselves to sufficiently large t. Assume that t is large enough that for all $x \geq 2$, $t^{x-1} \geq \exp(4\sqrt{x}\log^2 x)$. Set $c_3 = 4c_2$.

Let $s_t(n)$ denote the minimum of $s(G)$ for transitive groups G of degree n with a primitive structure tree T with no giant nodes of degree $> t$. For $1 \leq n \leq c_2 t$, we have $s_t(n) \geq n+1 \geq 2 \geq 2^{c_2/c_3} \geq 2^{n/(c_3 t)}$. To complete the proof, it is enough to prove the following claim:

Claim: *For* $n \geq c_2 t$, $s_t(n) \geq t2^{n/(c_3 t)}$.

Proof: By induction on n. Let G be a transitive group of degree $n \geq c_2 t$, and T a primitive structure tree for G with no giant nodes of degree larger than t. Suppose the claim is true for smaller values of n. Collapse the levels of T below level 1 to a single level. Let $H = H(\text{root})$, and let $L = L(u)$ for some $u \in \mathcal{L}_1$. We consider three cases:

1. $k_1 \geq c_2 t$. We have $|H| \leq k_0!$, and for $k_0 > t$ we have
$$|H| \leq \exp(4\sqrt{k_0}\log^2 k_0)$$
so in any case $|H| \leq t^{k_0-1}$. By induction, $s(L) \geq t2^{k_1/(c_3 t)}$, so
$$\begin{aligned} s(G) &\geq s(K_1)/|H| \geq s(L^{k_0})/|H| = s(L)^{k_0}/|H| \\ &\geq (t2^{k_1/(c_3 t)})^{k_0}/t^{k_0-1} = t2^{n/(c_3 t)} \end{aligned}$$
as desired.

2. $k_1 < c_2 t \leq k_0$. By Theorem 4.5 and choice of c_2 we have $|H| \leq 2^{k_0/2}$. Also,
$$t \leq 2^t = 2^{c_2 t/c_2} \leq 2^{k_0/8}$$
and
$$2^{n/(c_3 t)} = 2^{k_0 k_1/(c_3 t)} < 2^{k_0 c_2 t/(c_3 t)} = 2^{k_0/4}$$
Combining, we obtain
$$s(G) \geq s(L)^{k_0}/|H| \geq 2^{k_0}/|H| \geq 2^{k_0/2} \geq t2^{n/(c_3 t)}$$
as desired.

3. $k_1, k_0 < c_2 t$. In this case the estimate $s(G) \geq \binom{k_0+k_1}{k_0}$ suffices. This estimate is symmetric in k_0 and k_1, so we may assume that $k_0 \leq k_1$. Since $k_0 k_1 = n \geq c_2 t$, we must have $k_0 \geq 2$. Therefore,
$$\begin{aligned} s(G) &\geq \binom{k_0+k_1}{k_0} = \left(\frac{k_1+k_0}{k_0}\right)\cdots\left(\frac{k_1+1}{1}\right) \\ &> 2^{k_0-3}k_1^2 \geq 2^{k_0-3}k_1 k_0 \geq 2^{k_0-3}c_2 t \geq t2^{k_0} \\ &\geq t2^{(c_2/c_3)k_0} = t2^{(c_2 t)k_0/(c_3 t)} \geq t2^{n/(c_3 t)} \end{aligned}$$
proving the Claim and the Theorem.□