Coset Intersection in Moderately Exponential Time

László Babai*

DRAFT: April 21, 2008 Slightly updated May 8, 2013

Abstract

We give an algorithm to find the intersection of two subcosets in the symmetric group S_n in time $\exp(O(\sqrt{n}\log n))$.

1 Introduction

This paper is a somewhat overdue detailed version of an algorithm, outlined in [Ba4, BKaL], for the result stated in the Abstract. As pointed out by Gene Luks, the original outlines omitted some significant details. In this paper we fill this gap. The timeliness of this detailed description comes from new applications of the result [BCo].

The Coset Intersection Problem asks, given two subgroups $G, H \leq S_n$ and two permutations $\sigma, \tau \in S_n$, to determine the intersection $G\sigma \cap H\tau$ (which itself is either empty or a coset of the subgroup $G \cap H$). As pointed out by Luks [Lu1], this problem is intimately related to the Graph Isomorphism Problem. Isomorphism of graphs on n vertices easily reduces to the coset intersection problem over a permutation domain of size $\binom{n}{2}$, and important special cases of the Graph Isomorphism Problem, such as the cases of graphs of bounded degree and of tournaments, reduce to special cases of the Coset Intersection Problem that are efficiently solvable.

The best known general time bound for the Graph Isomorphism Problem, $\exp\left(O(\sqrt{n\log n})\right)$ (where n is the number of vertices), due to Luks, is outlined in [BKaL]; it uses consequences of the Classification of Finite Simple Groups (CFSG) to great depth. A more elementary algorithm yields the slightly weaker bound $\exp(O(\sqrt{n}\log n))$. Both algorithms use Luks's

*University of Chicago. Email: laci@cs.uchicago.edu.

divide-and-conquer methods [Lu1] combined with a combinatorial trick by Zemlyachenko [ZKT]; the algorithm appears in [BL].

The algorithm for Coset Intersection, presented in this paper, uses a Graph Isomorphism subroutine; for our purposes, the $\exp(O(\sqrt{n}\log n))$ bound suffices.

We remark that at the cost of an additional combinatorial lemma, we can avoid using Graph Isomorphism routines, and in fact the procedure will yield a new Graph Isomorphism algorithm, also running in $\exp(\widetilde{O}(\sqrt{n}))$ time (where the tilde indicates a polylogarithmic factor).

Remark 1.1. Our analysis of the Coset Intersection Algorithm presented depends on a (simply stated) consequence of the CFSG. A completely elementary analysis yields the slightly weaker bound $\exp\big(O(\sqrt{n}\log^2 n)\big)$.

We mention a corollary to the main result.

Corollary 1.2. *Given two subgroups G and H of S_n and a permutation $\sigma \in S_n$, the centralizer of H is $G\sigma$ can be found in time* $\exp(O(\sqrt{n}\log n))$.

Indeed, it is trivial to find the centralizer K of H in S_n; and now we just compute $G\sigma \cap K$. □

A word of explanation of the term "moderately exponential" in the title: every instance x of an NP problem is by definition associated with a "witness space" $W(x)$. The size of $W(x)$ is typically exponential in $|x|$. Exhaustive search of $W(x)$ constitutes the "brute force solution" to the problem. Let $w_n = \max\{|W(x)| : |x| \le n$. We call an algorithm "moderately exponential" if on inputs of length n it takes time $O(w_n^{1-c})$ for some positive constant c. For the graph isomorphism problem, for graphs with k vertices, and for the coset intersection problem on domains of size k, the size of the witness space is essentially $k!$, so a moderately exponential algorithm would take time $\exp\big(O(k^{1-c})\big)$, so our algorithms qualify as moderately exponential for any $c < 1/2$.

Acknowledgment. I am grateful to Gene Luks for pointing out a gap in the original (1983) outline, and for nearly three decades of friendship and collaboration.

Much of our early work is still not published in detail (or not published at all). The conference paper [BKaL] was an amalgam of the outlines of three separate results by the three authors. Only Bill Kantor wrote up a detailed (and greatly improved) journal version of his part [Ka] in a timely manner. Having now paid this old debt myself, hopefully Gene Luks will also follow

with a detailed writeup of his $\exp\left(O(\sqrt{n \log n})\right)$ graph isomorphism test, still the best bound after a quarter century.

2 Group theoretic preliminaries

2.1 Groups: general concepts

For a general introduction to groups we refer to [Ro]. In this section we fix some terminology and notation.

Throughout this paper, the letters G and H will denote finite groups; the relation $H \subseteq G$ indicates that H is a *subgroup* of G. For $H \leq G$ and $\sigma \in G$, the set $H\sigma$ is a *right coset* of H in G. The *index* of H in G is the number of distinct right cosets of H in G and is denoted by $|G : H|$.

A *coset representative* of the coset $H\sigma$ is any $\sigma' \in H\sigma$. Note that in this case, $H\sigma' = H\sigma$. Note also that a right coset of H is also a left coset of another subgroup, namely $H\sigma = \sigma H_1$ where $H_1 = \sigma^{-1} H \sigma$.

2.2 Subcosets

By a *subcoset* of G we mean a subset of G which is either the empty set or a coset by a subgroup. We also use the notation $H\sigma$ for the empty set; in this case $H = \emptyset$ (which is not a subgroup), and σ is any element of G.

The family of subcosets of G is closed under intersection. Indeed, if $H_1, H_2 \leq G$ and $\sigma_1, \sigma_2 \in G$ then either $H_1\sigma_1 \cap H_2\sigma_2 = \emptyset$ or $H_1\sigma_1 \cap H_2\sigma_2$ is a right coset of $H_1 \cap H_2$.

2.3 Permutation groups: general concepts

For additional introductory material on permutation groups we refer the reader to [Wi, Se, Cam]. We especially recommend Gene Luks's lucid introductions [Lu3, Lu1], tailor-made for work with permutation group algorithms.

Let Ω be a nonempty set. The *symmetric group* acting on Ω consists of all permutations of Ω and is denoted $\mathrm{Sym}(\Omega)$. The *alternating group* $\mathrm{Alt}(\Omega)$ is the subgroup of $\mathrm{Sym}(\Omega)$ consisting of the even permutations; for $|\Omega| \geq 2$ it has index 2 in $\mathrm{Sym}(\Omega)$. For a positive integer t we use the notation $[t]$ to denote the set $\{1, \ldots, t\}$; and $S_t = \mathrm{Sym}([t])$ and $A_t = \mathrm{Alt}([t])$.

The *permutation groups* acting on Ω are the subgroups of $G \leq \mathrm{Sym}(\Omega)$. If $|\Omega| = t$ then we say that G has *degree* t.

The *action* of a group G on the set Ω is a homomorphism $\varphi : G \to \mathrm{Sym}(\Omega)$. This action is *faithful* if $\ker(\varphi) = \{1\}$; so faithful actions on Ω can be identified with permutation groups acting on Ω.

Let us fix an action $\varphi : G \to \mathrm{Sym}(\Omega)$. For $\sigma \in G$ and $x \in \Omega$, we denote by x^σ the image of x under $\phi(\sigma)$. For $\Delta \subseteq \Omega$ we write $\Delta^\sigma = \{x^\sigma : x \in \Delta\}$. For $L \subseteq G$ we write $x^L = \{x^\sigma : \sigma \in L\}$.

The set x^G is the *orbit* of x under G; the orbits partition Ω. We call $|x^G|$ the *length* of the orbit x^G. G is *transitive* if it has only one orbit, i. e., $x^G = \Omega$ for any (and therefore all) $x \in \Omega$.

A subset $\Delta \subseteq \Omega$ is *G-invariant* if $\Delta^\sigma = \Delta$ for all $\sigma \in G$. Such set is a union of some orbits of G.

For $x \in \Omega$, the *stabilizer* of x is the subgroup $G_x = \{\sigma \in G : x^\sigma = x\}$. It is immediate that $|x^G| = |G : G_x|$.

For $\Delta \subseteq \Omega$, we let $G_\Delta = \bigcap_{x\in\Delta} G_x$ denote the pointwise stabilizer of Δ in G. The *setwise stabilizer* of Δ is the subgroup $G_{\{\Delta\}} = \{\sigma \in G : \Delta^\sigma = \Delta\}$.

A *partition* $\mathfrak{B} = (\Omega_1, \ldots, \Omega_t)$ of Ω splits Ω into the union of disjoint nonempty subsets: $\Omega = \dot{\bigcup}_{i=1}^{t} \Omega_i$. The Ω_i are the *blocks* of the partition $\mathfrak{B}$. A partition is *G-invariant* if $(\forall \sigma \in G)(\forall i \le t)(\exists j \le t)(\Omega_i^\sigma = \Omega_j)$. Such a partition is also called a *system of imprimitivity* and its blocks are *blocks of imprimitivity*.

The action is *primitive* if it is transitive, $|\Omega| \ge 2$, and Ω has no nontrivial G-invariant partition.

We say that $\mathfrak{B}$ is a *minimal system of blocks* for G if $\mathfrak{B}$ is an invariant partition, $|\mathfrak{B}| \ge 2$, and there is no coarser invariant partition $\mathfrak{B}'$ with $|\mathfrak{B}'| \ge 2$. So an action is primitive exactly if it is transitive and the discrete partition is a minimal G-invariant partition.

Since permutation groups $G \le \mathrm{Sym}(\Omega)$ act (faithfully) on Ω, all concepts discussed apply in particular to permutation groups.

2.4 Bounds on the order of primitive groups

The following bounds on the order of primitive groups will be central to the timing of our algorithms.

Theorem 2.1. (CFSG) *If $G \le S_n$ is a primitive permutation group and $G \not\ge A_n$ then $|G| \le n^{\sqrt{n}}$.*

This bound rests on the Classification of Finite Simple Groups (CFSG) and is derived in [Cam]. A slightly weaker bound has a completely elementary combinatorial proof:

Theorem 2.2. (Elementary estimate) *If $G \leq S_n$ is a primitive permutation group and $G \not\geq A_n$ then $|G| \leq \exp\left(4\sqrt{n}\ln^2 n\right)$.*

This result is proved in [Ba2, Ba3]; the second part is greatly improved and simplified in [Py].

2.5 Wreath products

Definition 2.3. *Let $\Omega = \Delta \times T$. Let $H \leq \mathrm{Sym}(\Delta)$ and $G \leq \mathrm{Sym}(T)$. The* wreath product of H by G *is the subgroup $H \wr G \leq \mathrm{Sym}(\Omega)$ defined as the product $K\hat{T}$ where $K = H^T$ is the direct product of $|T|$ copies of H acting independently on each copy $\Delta \times \{i\}$ $(i \in T)$; and $\hat{T}$ is the natural action of T on Ω defined as $\hat{T} = \{\hat{\sigma} : \sigma \in T\}$ where $(x,i)^{\hat{\sigma}} = (x, i^\sigma)$ for $x \in \Delta$ and $i \in T$. So K is a normal subgroup of $H \wr G$ and $K \cap \hat{T} = \{1\}$; consequently $|H \wr G| = |H|^{|T|}|G|$.*

Informally, $H \wr G$ acts on the union of t copies of Δ by acting on each copy independently by H and then permuting the copies by G. The partition $\mathfrak{D} = \{\Delta \times \{i\} : i \in T\}$ is a system of imprimitivity for $H \wr G$. If G is primitive then $\mathfrak{D}$ is a minimal block system. (An intuitive example of the wreath product occurs as the automorphism group of graph X which has t isomorphic connected components. Let H be the automorphism group of a component. Then $\mathrm{Aut}(X) = H \wr S_t$.)

Let $\mathfrak{D} = \{\Delta_1, \ldots, \Delta_t\}$ be a partition of Ω into blocks of equal size $|\Delta_i| = d$. Then we identify Ω with $\Delta \times [t]$ such that $\Delta_i = \Delta \times \{i\}$. We note that $\mathrm{Sym}(\Delta) \wr S_t$ is the largest subgroup of $\mathrm{Sym}(\Omega)$ under which the partition $\mathfrak{D}$ is invariant.

2.6 Giant Action

Our algorithm will operate with a global parameter n, to be thought of as the initial number of vertices. All permutation groups in their action on vertices will have degree $\leq n$ throughout the algorithm, including its recursive calls.

We shall say that a permutation group G of degree $d \leq n$ is a *giant* if (a) $d \geq 7$; (b) $d > 2\sqrt{n}$; (c) G is either S_d or A_d.

2.7 Giant action on the blocks: the structure theorem

The following structure theorem for permutation groups with a giant action on the blocks is central to our applications. It first appeared as [Ba4, Lemma 21.15].

The theorem is a corollary to its "unabridged" version, Theorem 6.4. The abridged version is stated without proof in [BKaL]; detailed proof of the unabridged version appears in [BBT]. For completeness, we include the proof in the Appendix.

We use the notation $[k] = \{1, \ldots, k\}$. As before, Ω is a set of n elements. A_t denotes the alternating group of degree t, acting on $[t]$.

Theorem 2.4. (Giant Action Theorem - abridged) [Ba4, Lemma 21.15] *Let $G \leq \mathrm{Sym}(\Omega)$ be a permutation group and $\mathfrak{B} = \{\Omega_1, \ldots, \Omega_t\}$ a G-invariant partition of Ω. Assume that the induced G-action on $\mathfrak{B}$ is the alternating group A_t, $t \geq 7$. Assume further that $t \geq 2\sqrt{|\Omega|}$. Let G_t be the (setwise) stabilizer of Ω_t in G and let $H \leq \mathrm{Sym}(\Omega_t)$ be the restriction of G_t to Ω_t. Then there exist subgroups $N \lhd H$, $R \leq G$, and $M \lhd G$ such that $M \leq K$, with the following properties.*

(a) $K \cap R = \{1\}$ and $KR = G$ (so the action of R on $\mathfrak{B}$ is A_t)

(b) N is the restriction of M to Ω_t

(c) $K/M \cong H/N$

(d) one can identify the set $\overline{\Omega}_t = \Omega \setminus \Omega_t$ with the product set $\Omega_t \times [t-1]$ and the restriction of the group MR to $\overline{\Omega}_t$ with the wreath product $N \wr A_{t-1}$.

Moreover, the subgroups R, M, N, as well as the stated identifications can be constructed in polynomial time.

The proof yields a stronger version of (c).

(c') Let K_t and M_t denote the restriction of K and M to $\overline{\Omega}_t$, resp. Then, under the identification stated in (d), we have that K_t/M_t is the diagonal of $(H/N)^{t-1}$.

2.8 Subdirect products of alternating groups; linked actions

We say that the group G is a *subdirect product* of the groups G_1 and G_2 if $G \leq G_1 \times G_2$ and G projects *onto* each component.

If $G_1 \cong G_2$ then let $\varphi : G_1 \to G_2$ be an isomorphism. Let $\mathrm{diag}_\varphi(G_1, G_2)$ denote the group $\{(\sigma, \varphi(\sigma)) : \sigma \in G_1\}$.

The following facts are well known.

Fact 2.5. *If G_1 and G_2 are finite simple groups and G is a subdirect product of G_1 and G_2 then either $G = G_1 \times G_2$ or $G_1 \cong G_2$ and $G = \mathrm{diag}_\varphi(G_1, G_2)$ under some isomorphism $\varphi : G_1 \to G_2$.*

It is clear that if $t = |\Omega_1| = |\Omega_2|$ then any bijection between these two sets induces an isomorphism between $\mathrm{Alt}(\Omega_1)$ and $\mathrm{Alt}(\Omega_2)$. For $t \neq 6$, the converse also holds:

Fact 2.6. *If $|\Omega_1| = |\Omega_2| \neq 6$ then all isomorphisms between $\mathrm{Alt}(\Omega_1)$ and $\mathrm{Alt}(\Omega_2)$ are induced by bijections between Ω_1 and Ω_2.*

(This is equivalent to the better known fact that for $t \neq 6$, $\mathrm{Aut}(A_t) = S_t$.)

Suppose now that $G \leq \mathrm{Sym}(\Omega)$ where $\Omega = \Omega_1 \dot\cup \Omega_2$ and G *respects* the Ω_i, i. e., each Ω_i is G-invariant. We say that the two sets G_1 and G_2 are *linked* under the G-action if there is a bijection $\psi : \Omega_1 \to \Omega_2$ such that the partition $\{\{x, \psi(x)\} : x \in \Omega_1\}$ of Ω is G-invariant.

The following observation is immediate by combining the Facts stated.

Corollary 2.7. *Let $\Omega = \Omega_1 \dot\cup \Omega_2$. Assume $|\Omega_i| \geq 7$ and G is a subdirect product of $\mathrm{Alt}(\Omega_1)$ and $\mathrm{Alt}(\Omega_2)$. Then either $G = \mathrm{Alt}(\Omega_1) \times \mathrm{Alt}(\Omega_2)$ or the two parts are linked under the G-action and $G = \mathrm{diag}_\varphi(\mathrm{Alt}(\Omega_1), \mathrm{Alt}(\Omega_2))$ where φ is the isomorphism induced by the linking of the Ω_i.*

2.9 Bipartite graphs

Finally we shall need a simple lemma about automorphism groups of bipartite graphs. We use the notation $\mathfrak{X} = (\Omega_1, \Omega_2; E)$ for a bipartite graph with the two parts of the vertex set being Ω_1 and Ω_2 and the set of edges $E \subseteq \Omega_1 \times \Omega_2$. The automorphisms (self-isomorphisms) of $\mathfrak{X}$ respect the parts by definition; they form the group $\mathrm{Aut}(\mathfrak{X}) \leq \mathrm{Sym}(\Omega_1) \times \mathrm{Sym}(\Omega_2)$. The *density* of $\mathfrak{X}$ is $|E|/(|\Omega_1||\Omega_2|)$.

Lemma 2.8. *Let $\mathfrak{X} = (\Omega_1, \Omega_2; E)$ be a bipartite graph of density $\leq 1/2$ with $|\Omega_i| \geq 7$. Assume $\mathrm{Aut}(\mathfrak{X})$ projects onto the symmetric or the alternating group on each Ω_i. Then E is either empty or a perfect matching; in the latter case, the the matching links the actions of $\mathrm{Aut}(\mathfrak{X})$ on the two parts (diagonal action).*

Proof. Let $G = \mathrm{Aut}(\mathfrak{X}) \cap (\mathrm{Alt}(\Omega_1) \times \mathrm{Alt}(\Omega_2))$. It easily follows from the simplicity of the alternating groups that G is a subdirect product of $\mathrm{Alt}(\Omega_1)$ and $\mathrm{Alt}(\Omega_2)$. If it is the full direct product then G is transitive on $\Omega_1 \times \Omega_2$ and therefore E is either empty or all of $\Omega_1 \times \Omega_2$; the latter contrdicts the density condition.

Alternatively, $|\Omega_1| = |\Omega_2|$ and $G = \mathrm{diag}_\varphi(\mathrm{Alt}(\Omega_1), \mathrm{Alt}(\Omega_2))$, where φ arises from a bijection $\psi : \Omega_1 \to \Omega_2$ linking the two parts under G. Now the G-action on $\Omega_1 \times \Omega_2$ has two orbits: the diagonal $\{(x, \psi(x)) : x \in \Omega_1\}$ and

its complement. The former corresponds to a perfect matching; the density of the latter is $1 - 1/|\Omega_i| \geq 6/7$, contradicting the density assumption.

Assume now that the edges of $\mathfrak{X}$ form the perfect matching $\{(x, \psi(x)) : x \in \Omega_1\}$. Then obviously, $\mathrm{Aut}(\mathfrak{X})$ is the diagonal of $\mathrm{Sym}(\Omega_1) \times \mathrm{Sym}(\Omega_2)$ induced by ψ, i. e., the edges link the two actions. □

3 Polynomial-time algorithms for permutation groups

3.1 Fundamental algorithms

Gene Luks [Lu3] gives an excellent introduction to the polynomial-time theory of permutation group algorithms. For a more detailed introduction we refer to the monograph by Ákos Seress [Se].

Here we list some conventions and basic facts.

We say that a permutation group is "given," or "known" if a list of generators is given (known). To "find a subgroup" means to find generators for the subgroup.

An *action* $G \to \mathrm{Sym}(\Delta)$ of a known permutation group is given by listing the actions of the generators.

A *constructive membership test* for a known permutation group $G \leq \mathrm{Sym}(\Omega)$ requires, for any given permutation $\sigma \in \mathrm{Sym}(\Omega)$, either to certify that $\sigma \notin G$ or to represent σ as a straight line program from the given generators of G.

Basic efficient algorithms for permutation groups have been designed by Charles Sims in the 1960s [Si1, Si2]. These algorithms include constructive membersip test in, and the determination of, the order of known permutation groups. A variation on Sims's algorithm was found in 1980 and proven to run in polynomial time ($O(n^6)$) by [FHL]. An algorithm which more closely follows Sims's basic data structure was subsequently described by Knuth [Kn] and shown to run in $O(n^5)$.

These algorithms also control the number of generators, avoiding a potential blowup in repeated application. Specifically, for every known group, a sequence of generators is found which is nonredundant in the sense that none is generated by the preceding ones; such a sequence of elements in S_n has length $< 2n$ [Ba5].

Normal closures and consequently *kernels of actions* can also be found in polynomial time [Si1, Si2, FHL, Kn].

A final corollary which we shall use repeatedly: let $\varphi : G \to \mathrm{Sym}(\Omega)$ be an action and let $H\tau$ be a subcoset of $\mathrm{im}(\varphi)$. Then the preimage $\varphi^{-1}(H\tau)$ can be found in polynomial time.

3.2 Finding recognizable subgroups

In this section we adapt an idea from [Ba1]: recognizable subgroups of small index in a known group can be found efficiently. This permits us to reduce the question of coset intersection with a group L to the same question with respect to a supergroup K in which L has moderate index.

Following [Ba1], we call a subgroup H of a group G *recognizable* if H admits an *efficient membership test* for elements of G.

In this paper we take G to be $\mathrm{Sym}(\Omega)$ and we use the term "efficient" to mean "polynomial time," but the concept of recognizability can be adapted to other precise concepts of efficiency as well.

Prime examples of recognizable subgroups are automorphism groups of graphs, intersections of known groups, centralizers of known subgroups. It is not known whether any of these classes of recognizable subgroups can be found in polynomial time (in fact, the last two are equivalent in polynomial time and the first one reduces to them [Lu1]), yet testing membership in each is can easily be done in polynomial time.

Proposition 3.1. (Recognizable subgroups) *Let $G \leq \mathrm{Sym}(\Omega)$ be a known group and let $H \leq G$ be recognizable in polynomial time. Then generators for H and right coset representatives of H in G can be found in time polynomial in $|\Omega|$ and $|G : H|$.*

Proof. Following [Ba1], we create the chain $G_0 \geq G_1 \geq \cdots \geq G_m$ of subgroups where $G_0 = G$, $G_1 = H$, and the rest is the stabilizer chain of H, ending at $G_m = \{1\}$. Now we adapt Sims's algorithm (or its [FHL] variant) to this chain to obtain generators of all members of the chain as well as coset representatives for each G_i in G_{i-1}, within the time bound stated. The time bound follows from [Kn, FHL]. □

3.3 The coset intersection problem

For the terminology on subcosets, see Section 2.2.

We say that a nonempty subcoset $H\sigma$ of $\mathrm{Sym}(\Omega)$ is "known" if it is given by a list of generators of the subgroup H and a coset representative $\sigma' \in H\sigma$.

The Coset Intersection Problem takes two subcosets of $\mathrm{Sym}(\Omega)$ and asks to determine their intersection.

Noting that $G\sigma \cap H\tau = (G \cap H\tau\sigma^{-1})\sigma$, we may always assume $\sigma = 1$.

The following observation is an easy consequence of Proposition 3.1 about finding recognizable subgroups of moderate index.

Proposition 3.2. *Let $G\sigma$ and $H\tau$ be given nonempty subcosets of* $\mathrm{Sym}(\Omega)$. *Then $G\sigma \cap H\tau$ can be found in time polynomial in $|\Omega|$ and $|G : G \cap H|$.*

Proof. As above, we may assume $\sigma = 1$. Note that $G \cap H$ is a recognizable subgroup of G, so by Proposition 3.1, we can find generators for $G \cap H$ and a set R of coset representatives of $G \cap H$ in G within the time bound stated. Now proceed as follows:

31 **for** $\rho \in R$
32 **if** $\rho \in H\tau$ **then return** $(G \cap H)\rho$, **exit**
33 **end(for)**
34 **return** "empty"

It is clear that what this procedure returns is $G \cap H\tau$. □

3.4 Two reductions of the coset intersection problem

For $G, H \leq \mathrm{Sym}(\Omega)$, we use the notation $\mathrm{INT}(G, H)$ to denote the class of coset intersection problems where the input groups are G and H (and the coset representatives vary). So an *instance* of $\mathrm{INT}(G, H)$ is specified by the two coset representatives.

Let $G, H, K \leq \mathrm{Sym}(\Omega)$.

Our first reduction is based on one of Luks's divide-and-conquer tricks [Lu1].

Lemma 3.3. (STEPDOWN) *If $K \leq G$ and $|G : K| = k$ then an instance of* $\mathrm{INT}(G, H)$ *reduces to k instances* $\mathrm{INT}(K, H)$*; the reduction takes time polynomial in k and $|\Omega|$.*

Note that the value of k need not be known in advance and will be computed in the process.

Proof. Let $G = KR$ where R is a set of right coset representatives of K in G (so $|R| = k$). Observe that

$$G\sigma \cap H\tau = \bigcup_{\rho \in R} K\rho\sigma \cap H\tau. \tag{1}$$

If the right-hand side is not empty then it comes in the form $\bigcup_{\rho' \in R'} (K \cap H)\rho'$ for some set R' where $|R'| \leq |R|$. We need to combine this union into the form $(G \cap H)\xi$. Choose ξ to be any of the ρ'; then $G \cap H$ is generated by $K \cap H$ and the set $\{\rho'\xi^{-1} : \rho' \in R'\}$. □

Our second reduction is based on the recognizability of the intersection of known subgroups.

Lemma 3.4. (STEPUP) *If $G \le K$ and $|K : G| = k$ then an instance of* $\mathrm{INT}(G, H)$ *reduces to a single instance of* $\mathrm{INT}(K, H)$*; the reduction takes time, polynomial in k and $|\Omega|$.*

Again, the value of k need not be known in advance and will be computed in the process.

Proof. Assume we need to find $G\sigma \cap H\tau$; we may assume $\sigma = 1$. If $K \cap H\tau = \emptyset$ then $G \cap H\tau = \emptyset$. Otherwise, let $G_1 = K \cap H$ and $H_1 = G$. Let further $K \cap H\tau = G_1\xi$. With this notation, we need to reduce finding $G_1\xi \cap H_1$ to finding $G_1\xi$ ($H_1 = G$ is given). According to Proposition 3.2, one can do this in time, polynomial in $|\Omega|$ and $|G_1 : G_1 \cap H_1|$. Observe that $|G_1 : G_1 \cap H_1| = |(K \cap H) : (G \cap H)| \le |K : G| = k$. □

4 The color-isomorphism problem

4.1 Automorphisms vs. isomorphisms. Luks's window

Let $f : \Omega \to \Sigma$ be a coloring. Following Luks [Lu1], for $\Psi \subseteq \Omega$ and $L \subseteq \mathrm{Sym}(\Omega)$ we define the set $\mathcal{C}_\Psi(L)$ of *L-automorphisms of f with respect to the "window"* Ψ as

$$\mathcal{C}^f_\Psi(L) = \{\sigma \in L : (\forall x \in \Psi)(f(x) = f(x^\sigma)\}. \tag{2}$$

We write $\mathcal{C}^f(L) = \mathcal{C}^f_\Omega(L)$; this is the set of L-automorphisms of f.

As pointed out by Luks, if $G \le \mathrm{Sym}(\Omega)$ and Ψ is G-invariant then $\mathcal{C}_\Psi(G)$ is a subgroup of G, and for $\sigma \in \mathrm{Sym}(\Omega)$, $\mathcal{C}_\Psi(G\sigma)$ is either empty or a right coset of $\mathcal{C}_\Psi(G)$.

Let now f, f' be two colorings of Ω, $\Psi \subseteq \Omega$, and $L \subseteq \mathrm{Sym}(\Omega)$. We define the set of *$L$-isomorphisms* of f and f' with respect to Ψ as

$$\mathrm{ISO}_\Psi(L; f, f') = \{\sigma \in L : (\forall x \in \Psi)(f(x) = f'(x^\sigma)\}. \tag{3}$$

We omit the subscript Ψ if $\Psi = \Omega$.

For $\tau \in \mathrm{Sym}(\Omega)$, we define the coloring f^τ by $f^\tau(x) = f(x^{\tau^{-1}})$. We note that

$$\mathrm{ISO}_\Psi(L; f, (f')^\tau) = \mathrm{ISO}(L\tau^{-1}; f, f'). \tag{4}$$

In particular, if $G \le \mathrm{Sym}(\Omega)$ and $\tau \in \mathrm{Sym}(\Omega)$ then

$$\mathcal{C}^f_\Psi(G\tau) = \mathrm{ISO}_\Psi(G\tau; f, f) = \mathrm{ISO}_\Psi(G; f, f^{\tau^{-1}}), \tag{5}$$

so the study of automorphisms of strings within a subcoset is equivalent to the study of isomorphisms of strings within the corresponding subgroup.

First we prove an important special case of our main result.

Theorem 4.1. *Let f be a coloring of Ω. Let $G\sigma$ be a subcoset of* $\mathrm{Sym}(\Omega)$. *Then the subcoset $\mathcal{C}^f(G\sigma)$ of $G\sigma$-automorphisms of f can be computed in time* $\exp(O(\sqrt{n}\log n))$ *where* $n = |\Omega|$.

4.2 Reductions

Let $\Delta \subseteq \Omega$ and let f be the characteristic function of Δ. We define the (setwise) set-stabilizer

$$\mathcal{C}^{\Delta}(L) = \mathcal{C}^{f}(L). \tag{6}$$

Analogously we define the subcoset of set-isomorphisms by

$$\mathrm{ISO}(L, \Delta_1, \Delta_2) = \mathrm{ISO}(L, f_1, f_2) \tag{7}$$

where f_i is the characteristic function of Δ_i.

So the set-stabilizer and set-isomorphism problems are special cases of the color-automorphims and color-isomorphism problems.

The color-automorphism and isomorphism problems easily reduce to coset intersection: let $H\tau$ be the subcoset of isomorphisms of f_1 and f_2 with respect to $\mathrm{Sym}(\Omega)$ (this subcoset is esy to compute in polynomial time); then

$$\mathrm{ISO}(G\sigma, f_1, f_2) = G\sigma \cap H\tau. \tag{8}$$

Luks pointed out that conversely, coset intersection reduces to set stabilizer, an idea that will be the starting point of our set stabilizer algorithm. The reduction is given in Section 5.1.

4.3 The divide-and-conquer routines

Let $G \leq \mathrm{Sym}(\Omega)$, $\sigma \in \mathrm{Sym}(\Omega)$, and let $\Psi \subseteq \Omega$ be nonempty and G-invariant.

To compute $\mathcal{C}^f_{\Psi}(G\sigma)$, we follow Luks's divide-and-conquer strategy and apply our structure theorem in the case G is transitive with a giant on top. The divide-and-conquer strategy is based on the following observations (Luks): for any subsets $\Psi, \Psi_1, \Psi_2 \subseteq \Omega$ and any $L, L_1, L_2 \subseteq \mathrm{Sym}(\Omega)$ we have

(R1) $\mathcal{C}^f_{\Psi_1 \cup \Psi_2}(L) = \mathcal{C}^f_{\Psi_1}(\mathcal{C}^f_{\Psi_2}(L))$;

(R2) $\mathcal{C}^f_{\Psi}(L_1 \cup L_2) = \mathcal{C}^f_{\Psi}(L_1) \cup \mathcal{C}^f_{\Psi}(L_2)$.

First we describe three subroutines that will be used to call the main procedure recursively, executing Luks's divide-and-conquer strategy.

The first routine takes a partition of Ω to G-invariant parts and reduces Ω to those parts.

Procedure $\mathsf{PART\_INV}(\Omega, f_1, f_2, G, \sigma, \mathfrak{D})$

Input: a nonempty set Ω, colorings $f_i : \Omega \to \Sigma$ $(i = 1, 2)$, a group $G \leq \mathrm{Sym}(\Omega)$, a permutation $\sigma \in \mathrm{Sym}(\Omega)$, a partition $\mathfrak{D} = (\Delta_1, \ldots, \Delta_t)$ of Ω into $t \geq 2$ G-invariant subsets Ω_i.

Output: $\mathrm{ISO}(G\sigma; f_1, f_2)$

41 $H\tau := G\sigma$
42 **for** $i = 1$ **to** t
43 $\quad H\tau := \mathsf{ISO}(\Omega, f_1, f_2, H, \tau, \Delta_i)$
44 $\quad$ (: the subcoset $H\tau$ is reduced to $\mathrm{ISO}_{\Delta_i}(H\tau; f_1, f_2)$:)
45 **end(for)**
46 **return** $H\tau$

The next routine reduces the question of isomorphisms with respect to a group G to the same question with respect to a subgroup H of moderate index, implementing the idea of Lemma 3.3.

Procedure $\mathsf{STEPDOWN}(\Omega, f_1, f_2, G, H)$

Input: a nonempty set Ω, colorings $f_i : \Omega \to \Sigma$ $(i = 1, 2)$, two groups $H \leq G \leq \mathrm{Sym}(\Omega)$

Output: $\mathrm{ISO}(G; f_1, f_2)$

51 let R be a set of right coset representatives of H in G:
52 $\quad G = \dot{\bigcup}_{\sigma \in R} H\sigma$
53 **return** $\bigcup_{\sigma \in R} \mathsf{ISO}(\Omega, f_1, f_2, H, \sigma, \mathfrak{D})$
$\quad$ (: the value of $\mathfrak{D}$ is received from the main procedure :)

The third routine takes an invariant partition, reduces to the kernel of action on the blocks, and reduces Ω to the blocks.

Procedure $\mathsf{BLOCKS}(\Omega, f_1, f_2, G, \mathfrak{B})$

Input: a nonempty set Ω, colorings $f_i : \Omega \to \Sigma$ $(i = 1, 2)$, a group $G \leq \mathrm{Sym}(\Omega)$, a G-invariant partition $\mathfrak{B} = (\Omega_1, \ldots, \Omega_t)$ of Ω into $t \geq 2$ blocks Ω_i.

Output: $\mathrm{ISO}(G; f_1, f_2)$

61 let $\varphi : G \to \mathrm{Sym}(\mathfrak{B})$ be the G-action on the blocks
62 $K := \ker(\varphi), \quad \mathfrak{D} := \mathfrak{B}$
$\quad$ (: we redefined $\mathfrak{D}$ so this partition will be used in Step 4
$\quad$ of the main procedure :)
63 **return** $\mathsf{STEPDOWN}(\Omega, f_1, f_2, G, K)$

4.4 The case of wreath product with a giant

Assume $\Omega = \Delta \times [t]$ and $G = H \wr S_t$ where $H \leq \mathrm{Sym}(\Delta)$.

Procedure WREATH_SYM(Δ, t, f_1, f_2, H)

Input: a nonempty set Δ, integer $t \geq 1$, two colorings $f_i : \Delta \times [t] \to \Sigma$, a group $H \leq \mathrm{Sym}(\Delta)$.

Output: $\mathrm{ISO}(H \wr S_t; f_1, f_2)$.

71 **for** $j = 1, 2$
72 **for** $i = 1$ **to** t
73 $f_j^i := f_j|_{\Delta \times \{i\}}$ (: the "components of f_j" :)
74 **end(for)**
75 **end(for)**
76 **for** $j, j' = 1, 2$
77 **for** $i, i' = 1$ **to** t
78 $H(i, i', j, j') := \mathrm{ISO}(H; f_j^i, f_{j'}^{i'})$
(: computed either recursively or by brute force enumeration :)
79 **end(for)**
80 **end(for)**
81 piece together the $H(i, i', j, j')$ to obtain $\mathrm{ISO}(H \wr S_t; f_1, f_2)$
much like the way the isomorphisms of graphs are obtained
from the isomorphisms of their connected components
82 **return** $\mathrm{ISO}(H \wr S_t; f_1, f_2)$

Next we consider the situation when $G = H \wr A_t$. We implement the "STEPUP" idea (Lemma 3.4) to reduce to the case of the supergroup $H \wr S_t$.

Procedure WREATH_ALT(Δ, t, f_1, f_2, H)

Input: a nonempty set Δ, integer $t \geq 1$, two colorings $f_i : \Delta \times [t] \to \Sigma$, a group $H \leq \mathrm{Sym}(\Delta)$.

Output: $\mathrm{ISO}(H \wr A_t; f_1, f_2)$.

91 $G_1\sigma := \mathsf{WREATH\_SYM}(H \wr S_t; f_1, f_2)$
92 $H_1 := H \wr A_t$
93 use the procedure of Proposition 3.2 to
94 **return** $G_1\sigma \cap H_1$
95 (: Note: $|G_1 : G_1 \cap H_1| \leq |(H \wr S_t) : (H \wr A_t)| = 2$. :)

4.5 Finishing the giant

Procedure GIANTFINISH$(\Omega, f_1, f_2, G, \mathfrak{B})$

Input: a nonempty set Ω of size $|\Omega| \leq n$, colorings $f_i : \Omega \to \Sigma$ $(i = 1, 2)$, a group $G \leq \mathrm{Sym}(\Omega)$, a G-invariant partition $\mathfrak{B}$ of of Ω such that $t := |\mathfrak{B}| > \max\{6, 2\sqrt{n}\}$ and the image P of the G-action on $\mathfrak{B}$ is either $\mathrm{Sym}(\mathfrak{B})$ or $\mathrm{Alt}(\mathfrak{B})$.

Output: $\mathrm{ISO}(G; f_1, f_2)$.

(GF1) **if** $P = \mathrm{Sym}(\mathfrak{B})$ **then**
$G_1 := \varphi^{-1}(\mathrm{Alt}(\mathfrak{B}))$ (: so $|G : G_1| = 2$:)
return STEPDOWN$(\Omega, f_1, f_2, G, G_1)$, **exit**
(: henceforth $P = \mathrm{Alt}(\mathfrak{B})$:)

(GF2) let G_t be the setwise stabilizer of Ω_t in G
let H be the restriction of G_t to Ω_t
let K be the kernel of the G-cation on $\mathfrak{B}$
construct the subgroups $R \leq G$, $M \lhd G$ $(M \leq K)$ and $N \lhd H$ with the properties stated in the Giant Action Theorem (Theorem 2.4)
if $N \neq H$ **then**
return STEPDOWN$(\Omega, f_1, f_2, G, MR)$, **exit**
(: henceforth $N = H$ and therefore $M = K$ and $MR = G$:)

(GF3) **if** $|G| < |N|^t t!/2$ (: G is not the wreath product $N \wr A_t$:)
then $\mathfrak{D} := (\Omega \setminus \Omega_t, \Omega_t)$
(: we redefined $\mathfrak{D}$; this will force the algorithm to process, in Step 4, the block $\Omega \setminus \Omega_t$ first :)
STEPDOWN$(\Omega, f_1, f_2, G, G_t)$, **exit**
(: henceforth $|G| = |N|^t t!/2$ and therefore G is permutationally isomorphic to $N \wr A_t$:)

(GF4) identify Ω with $\Omega_t \times [t]$ and G with $H \wr A_t$ (Theorem 2.4)
return WREATH_ALT$(\Omega, f_1, f_2, G, \mathfrak{B})$, **exit**

4.6 The main color-isomorphism procedure

The number n will be a global constant (does not change during the execution of the algorithm) and will not be mentioned explicitly but it is part of the input to all procedures. It is the size of the initial domain Ω. The

domain Ω itself will vary; throughout the algorithm, we shall have $|\Omega| \leq n$. All variables are global, their values are passed down to the recursive calls even if this is not being made explicit. This is particularly important for the variable $\mathfrak{D}$, a partition of the current domain which controls some of the divide-and-conquer reductions.

A permutation group $P \leq \mathrm{Sym}(\mathfrak{B})$ is a *giant* if $|\mathfrak{B}| \geq \max\{7, 2\sqrt{n}\}$ and P is either $\mathrm{Sym}(\mathfrak{B})$ or $\mathrm{Alt}(\mathfrak{B})$.

Recall that by definition, $\mathrm{ISO}(G\sigma; f_1, f_2) = \mathrm{ISO}_\Omega(G\sigma; f_1, f_2)$.

In the comments, "henceforth [statement]" means "[statement] holds once we got past this step." However, recursive calls may move us back to earlier steps where [statement] may again fail to hold.

Procedure $\mathsf{ISO}(\Omega, f_1, f_2, G, \sigma, \Psi, \mathfrak{D})$
Note. If σ is the identity, we omit σ from the list of arguments. If, in addition, $\Psi = \Omega$, we omit Ψ as well.

Input: a nonempty set Ω of size $|\Omega| \leq n$, colorings $f_i : \Omega \to \Sigma$ $(i = 1, 2)$, a group $G \leq \mathrm{Sym}(\Omega)$, a permutation $\sigma \in \mathrm{Sym}(\Omega)$, a nonempty G-invariant subset $\Psi \subseteq \Omega$, a partition $\mathfrak{D}$ of Ψ into nonempty subsets

Output: $\mathrm{ISO}_\Psi(G\sigma, f_1, f_2)$

Note that the output does not depend on $\mathfrak{D}$; initially we may start with a completely arbitrary partition $\mathfrak{D}$.

1. **if** $|\Omega| \leq 2\sqrt{n}$ **then** use brute force enumeration to compute and **return** $\mathrm{ISO}_\Psi(G\sigma, f_1, f_2)$
   (: henceforth, $|\Omega| > 2\sqrt{n}$:)

2. **if** $\sigma \neq 1$ **then return** $\mathsf{ISO}(\Omega, f_1, f_2^{\sigma^{-1}}, G, \Psi, \mathfrak{D})$, **exit**
   (: henceforth, $\sigma = 1$:)

3. **if** $\Psi \neq \Omega$ **then** let $\varphi : G \to \mathrm{Sym}(\Psi)$ be the restriction homomorphism **return** $\varphi^{-1}(\mathsf{ISO}(\Psi, f_1|_\Psi, f_2|_\Psi, \varphi(G), \mathfrak{D}))$, **exit**
   (: henceforth, $\Psi = \Omega$ (we operate with "full screen") :)

4. let $(\Delta_1, \ldots, \Delta_s) := \mathfrak{D}$
   if $s \geq 2$ and each Δ_i is G-invariant **then**
   return $\mathsf{PART\_INV}(\Omega, f_1, f_2, G, 1, \mathfrak{D})$, **exit**
   (: we reduced Ω to the Δ_i in the given order; the order will be significant in Step (GF3); this is the only reason why we had to include $\mathfrak{D}$ in the input :)

5. **if** G is intransitive **then**
   let $\mathfrak{B} = (\Omega_1, \ldots, \Omega_t)$ be the partition of Ω into G-orbits
   return PART_INV$(\Omega, f_1, f_2, G, 1, \mathfrak{B})$, **exit**
   (: henceforth, G is transitive on Ω :)

6. let $\mathfrak{B} = (\Omega_1, \ldots, \Omega_t)$ of Ω be a minimal G-invariant partition
   let $\varphi : G \to \mathrm{Sym}(\mathfrak{B})$ be the induced G-action on $\mathfrak{B}$
   $P := \mathrm{im}(\varphi), \quad K = \ker(\varphi)$.
   (: $P \cong G/K$; and $P \leq \mathrm{Sym}(\mathfrak{B})$ is a primitive group :)

7. **if** P is not a giant **then**
   return BLOCKS$(\Omega, f_1, f_2, G, \mathfrak{B})$, **exit**
   (: if this step applies, it reduces $|\Omega|$ to $|\Omega|/t$:)
   (: henceforth P is a giant; in particular, $|\mathfrak{B}| > 2\sqrt{n}$:)

8. GIANTFINISH$(\Omega, f_1, f_2, G, \mathfrak{B})$, **exit**

4.7 Correctness of the algorithm

Steps 2 to 7 reduce the question to the case when G is transitive with a giant acting on the blocks of a minimal invariant partition. At that point we invoke the GIANTFINISH routine (Section 4.5).

We now consider the GIANTFINISH routine.

Step (GF1) reduces this to the case where the giant is alternating. Step (GF2) reduces to the case in which G_t acts as the wreath product $H \wr A_t$ on $\Omega \setminus \Omega_t$ according to Theorem 2.4.

Step (GF3) has a somewhat delicate interaction with Step 4 of the main procedure (Section 4.6). After stabilizing (setwise) the block Ω_t, we create the partition $\mathfrak{D} = (\Omega \setminus \Omega_t, \Omega_t)$, consisting of G-invariant parts (with respect to the reduced G). The STEPDOWN routine then sends us back to Step 4 where the part $\Omega \setminus \Omega_t$ will be processed first; this is critical.

We know from Theorem 2.4 (the Giant Action Theorem) that now the action on $\Omega \setminus \Omega_t$ is the wreath product $H \wr A_t$, so Step 4 takes us straight to Step (GF4), a leaf of the recursion tree. On the other branch, the window Ω_t, of size $< \sqrt{n}/2$, is processed; after the reduction to "full screen" in Step 3, we conclude with brute-force enumeration (Step 1, again a leaf).

4.8 Timing Analysis

Let $T(n)$ denote the maximum time our algorithm takes on instances under the given global constant n. "Polynomial time" will refer to an upper bound

that is polynomial in n.

Our recursion tree has two kinds of leaves: Step 1 (brute force enumeration when the domain is small enough) and Step (GF4), the conclusion when G has the wreath product structure with a large alternating group.

The cost of Step 1 is at most $\exp((\sqrt{n}+O(1))\ln n)$ each time it is executed.

Procedure WREATH_SYM, described in Section 4.4, is polynomial time. Consequently, the same holds for procedure WREATH_ALT, also described in Section 4.4, in view of the analysis of the STEPUP idea (Lemma 3.4) and the fact that the index k of the "STEPUP" used is 2. This means the cost of Step (GF4) of the algorithm is polynomial each time it is executed.

Next, we analyse the cost of each invocation of the GIANTFINISH procedure. The relatively straighforward structure of the subtree under such a node makes the cost easy to estimate.

Claim 4.2. *The cost of an invocation of* GIANTFINISH *is at most* $\exp((2\sqrt{n}+O(1))\ln n)$.

Proof. Step (GF1) reduces an instance of the problem to two if its instances, while achieving the progress indicated; in the subsinstances, $P = \mathrm{Alt}(\mathfrak{B})$. Step (GF2) further reduces each instance to at most $|H| \le |\Omega_t|! \le (n/t)! < (4n)^{\sqrt{n}}$ instances, achieving the progress stated; in the subinstances, $N = H$ and $G = M$. (GF3) reduces each instance to t instances, each of which will bottom out in two leaves: first, an application of the polynomial-time WREATH_ALT procedure, and second, the brute-force treatmant of the residual action on Ω_t. The cost of the latter is $\exp((\sqrt{n}+O(1))\ln n)$; this also includes the (polynomial) overhead with all reductions. So the total cost is $\exp((2\sqrt{n}+O(1))\ln n)$. □

Let us refer to an invocation of GIANTFINISH as a "quasi-leaf" in our recursion tree. Let us also refer to a call to Step 1 (brute force on small domains) as a "quasi-leaf" in case such a call is not part of a GIANTFINISH subtree.

Let $L(n,m)$ denote the maximum number of quasi-leaves in the execution of our main procedure, assuming $|\Omega| \le m$ (where $m \le n$).

The overhead of each recursive call is polynomial (including Steps 2, 3, 6).

It follows that the total cost of the algorithm is

$$T(n) \le \exp\left((2\sqrt{n}+O(1))\ln n\right) L(n,n). \tag{9}$$

We need to estimate $L(n,n)$.

If $m \leq 2\sqrt{n}$ then we are in a leaf (Step 1) so $L(n,m) = 1$. So we may assume $m > 2\sqrt{n}$ and so "polynomial in n" and "polynomial in m" are synonymous.

The recursive calls rely on the procedures described in Section 4.3. For the most part, this is Luks's divide-and-conquer, and the analysis follows the lines of [Lu1].

Procedure PART_INV is based on the partition of Ω into G-invariant subsets $\Delta_1, \ldots, \Delta_t$. The resulting recurrence is

$$L(n,m) \leq \sum_{i=1}^{t} L(n,m_i), \tag{10}$$

where $m_i \geq 1$ and $\sum_{i=1}^{t} m_i = m$. Since $L(n,1) = 1$, this recurrence is satisfied with equality by the function $L(n,m) = m$.

This takes care of the analysis of Steps 4 and 5, both of which implement PART_INV.

Procedure STEPDOWN (Section 4.3) takes us from the group G to the subgroup $H \leq G$; the number of quasi-leaves under G will be at most $|G:H|$ times the number of quasi-leaves under H.

This procedure is used in Procedure BLOCKS which takes us from a transitive group G to the kernel of its action P on t blocks of imprimitivity; and then Procedure PART_INV takes us separately to each block. If $|P| \leq t^{e(n)}$ for some function $e(n)$ then this gives us the recursive inequality

$$L(n,m) \leq t^{e(n)+1} L(n,m/t). \tag{11}$$

This recurrence is satisfied with equality by the function $L(n,m) = m^{e(n)+1}$.

In Step 7 we invoke procedure BLOCKS for non-giant primitive groups P; according to the bound of Theorem 2.1, we have $|P| < n^{\sqrt{n}} \leq t^{\sqrt{n}}$, so we can use the above analysis with $e(n) = \sqrt{n}$.

Let now $L^*(n,m)$ be a function that satisfies the following inequalities: the opposite of inequality (10):

$$L^*(n,m) \geq \sum_{i=1}^{t} L^*(n,m_i), \tag{12}$$

the opposite of inequality (11):

$$L^*(n,m) \geq t^{e(n)+1} L^*(n,m/t), \tag{13}$$

and the inequality $L^*(n,m) \geq 1$ for all m.

We observe that the function $L^*(n,m) = m^{e(n)+1}$ satisfies all the stated inequalities, where $e(n) = \sqrt{n}$. This proves the bound $L(n,m) \leq m^{\sqrt{n}+1}$ and consequently, in view of inequality (9), the overall time bound

$$T(n) \leq \exp\left((3\sqrt{n} + O(1)) \ln n\right). \tag{14}$$

We note that if we only use the elementary estimate of Theorem 2.2 on the order of non-giant primitive permutation groups then we obtain the slightly weaker bound

$$T(n) \leq \exp\left(O(\sqrt{n} \ln^2 n)\right). \tag{15}$$

5 Coset intersection

5.1 Reduction to stabilizer of transversal

We address the following problem, amenable to recursion. Let Ω_1, Ω_2 be sets, $\Psi_i \subseteq \Omega_i$, $\Psi = \Psi_1 \times \Psi_2$, and $\Omega = \Omega_1 \times \Omega_2$.

Let $L \subseteq \mathrm{Sym}(\Omega_1) \times \mathrm{Sym}(\Omega_2)$.

We use pr_i denote the projection operators: $\mathrm{pr}_i(\alpha_1, \alpha_2) = \alpha_i$ $(i = 1, 2)$. We use this notation both for the case $\alpha_i \in \Omega_i$ and for $\alpha_i \in \mathrm{Sym}(\Omega_i)$.

Let $\Delta \subseteq \Omega$ and let f be the characteristic function of Δ. We define the (setwise) stabilizer of Δ (with respect to the window Ψ) as the stabilizer of the characteriztic function of Δ:

$$\mathcal{C}_\Psi^\Delta(L) = \mathcal{C}_\Psi^f(L) \tag{16}$$

where the right-hand side is defined by equation (2). (This notation is consistent with the definition given in Section 4.2 where the window was assumed to be $\Psi = \Omega$.)

We say that Δ is a *partial transversal* of Ω if for every $x \in \Omega_i$ we have $|\mathrm{pr}_i^{-1}(x)| \leq 1$ $(i = 1, 2)$. Δ is a *(complete) transversal* of Ω if for every $x \in \Omega_i$ we have $|\mathrm{pr}_i^{-1}(x)| = 1$ $(i = 1, 2)$. Note that in the latter case, Δ establishes a bijection between Ω_1 and Ω_2; in particular, in this case, $|\Omega_1| = |\Omega_2|$. Note also that being a partial/complete transversal is invariant under $\mathrm{Sym}(\Omega_1) \times \mathrm{Sym}(\Omega_2)$.

Let $G \leq \mathrm{Sym}(\Omega_1) \times \mathrm{Sym}(\Omega_2)$ and $\sigma \in \mathrm{Sym}(\Omega_1) \times \mathrm{Sym}(\Omega_2)$. We shall consider the problem of finding $\mathcal{C}_\Psi^\Delta(G\sigma)$ under the following conditions:

(i) Ψ_i is $\mathrm{pr}_i(G)$-invariant $(i = 1, 2)$;

(ii) Δ is a partial transversal.

Without condition (ii), our problem would be to determine the automorphism group of a bipartite graph within $G\sigma$ (with respect to the window Ψ); under condition (ii), our bipartite graph is a matching.

As Luks points out, coset intersection is a special case of this problem. Indeed, let $\Omega = \Omega' \times \Omega'$; let $H_i \leq \mathrm{Sym}(\Omega')$ and $\sigma_i \in \mathrm{Sym}(\Omega')$ $(i = 1, 2)$. Then

$$H_1\sigma_1 \cap H_2\sigma_2 = \mathcal{C}^{\mathrm{diag}(\Omega)}(H_1\sigma_1 \times H_2\sigma_2) \tag{17}$$

where $\mathrm{diag}(\Omega) = \{(x, x) : x \in \Omega'\}$.

We note that if $K \leq \mathrm{Sym}(\Omega_1) \times \mathrm{Sym}(\Omega_2)$ and $H \leq \mathrm{pr}_i(K)$ then $\mathrm{pr}_i^{-1}(H) \leq K$ can be determined in polynomial time (preimage of subgroup of action, see Section 3.1). This fact will be used throughout.

5.2 The algorithm: reduction to giant actions

As before, we reduce the "stabilizer in a coset" problem to "isomorphism within a subgroup." We consider the problem of finding $\mathrm{ISO}(\Omega, \Delta_1, \Delta_2, G, \sigma, \Psi)$ where $\Omega = \Omega_1 \times \Omega_2$; the Δ_i are partial transversals of Ω; $G \leq \mathrm{Sym}(\Omega_1) \times \mathrm{Sym}(\Omega_2)$, and $\sigma \in \mathrm{Sym}(\Omega_1) \times \mathrm{Sym}(\Omega_2)$; and $\Psi = \Psi_1 \times \Psi_2$ is a "rectangular" window; we assume $\Psi_i \subseteq \Omega_i$ is invariant under $\mathrm{pr}_i(G)$.

The set $\mathrm{ISO}(\Omega, \Delta_1, \Delta_2, G, \sigma, \Psi)$ consists of those elements of the coset $G\sigma$ which transform Δ_1 into Δ_2.

1. If $|\Delta_1| \neq |\Delta_2|$ then return the empty set, exit.
   (: Henceforth, $|\Delta_1| = |\Delta_2|$. :)

2. If $\Delta_1 = \emptyset$ then return $G\sigma$, exit. (: Henceforth, $|\Delta_1| = |\Delta_2| \neq 0$. :)

3. If for $i = 1$ or 2, $|\Psi_i| = 1$, terminate in polynomial time: as pointed out by Luks [Lu1], in this case we are reduced to finding the stabilizer of a point in a subcoset of $\mathrm{Sym}(\Omega_{3-i})$, a task solvable in polynomial time by Sims's algorithm.

4. If for $i = 1$ or 2, $|\Omega_i| \leq 2\sqrt{n}$ then we perform brute force enumeration of each element of $\mathrm{pr}_i(G\sigma)$, reducing to $|G\sigma|$ instances of the preceding step.
   (: Henceforth, $|\Omega_i| > 2\sqrt{n}$:)

5. Just as in Steps 2 and 3 of the color-isomorphism algorithm of Section 4.6, reduce to the case $\sigma = 1$ and $\Psi = \Omega$.

6. Use Luks's first divide-and-conquer trick in each projection (Step 5 of the color-isomorphism algorithm of Section 4.6) to reduce to the case when both projections of G are transitive.

7. Use the color-isomorphism routine to reduce to the case when each Δ_i is a full transversal. Indeed, we have $|\Delta_1| = |\Delta_2| \neq 0$. So if Δ_i is not a full transversal then for $j = 1$ or 2, $\mathrm{pr}_j(\Delta_i)$ is a nonempty proper subset of Ω_j. Let

$$H\tau = \mathrm{ISO}(\mathrm{pr}_j(G), \mathrm{pr}_j(\Delta_1), \mathrm{pr}_j(\Delta_2)) \tag{18}$$

(computed via the algorithm of Section 4.6). Let further $G_1\rho = \mathrm{pr}_j^{-1}(H\tau)$ (the preimage of H in G under the projection pr_j). Now clearly

$$\mathrm{ISO}(G, \Delta_1, \Delta_2) = \mathrm{ISO}(G_1\rho, \Delta_1, \Delta_2), \tag{19}$$

and $\mathrm{pr}_j(G_1)$ is intransitive, so we recurse.

(: Henceforth $|\Delta_1| = |\Delta_2| = |\Omega_1| = |\Omega_2| > 2\sqrt{n}$ and each Δ_i defines a bijection between Ω_1 and Ω_2. :)

8. Let $\mathfrak{B}_j$ be a minimal G-invariant partition of Ω_j and let $P_j \leq \mathrm{Sym}(\mathfrak{B}_j)$ be image of the primitive G-action on $\mathfrak{B}$. Let K_j be the kernel of the G-action on $\mathfrak{B}_j$.

   If P_j is not a giant, use Luks's second divide-and-conquer trick (Step 7 of the color-isomorphism algorithm of Section 4.6) to reduce $\mathrm{pr}_j(G)$ to K_j.
   (: Henceforth, each P_j is a giant :)

   We discuss this case in the next section.

5.3 Giant action on the blocks: graph structure

At this point it will be convenient to switch to the language of bipartite graphs (see Section 2.9 for notation and terminology). We can equivalently describe our situation as follows: we have two bipartite graphs $\mathfrak{X}_i = (\Omega_1, \Omega_2; \Delta_i)$. The statement that Δ_i is a full transversal means that $\mathfrak{X}_i$ is just a perfect matching between Ω_1 and Ω_2.

Moreover, G acts on $\Omega_1 \dot\cup \Omega_2$, respecting each part. We need to find the subcoset $H\tau$ of G-isomorphisms between $\mathfrak{X}_1$ and $\mathfrak{X}_2$.

We also have the minimal G-invariant partition $\mathfrak{B}_i$ of Ω_i; and the G action on each $\mathfrak{B}_i$ is a giant.

Let us contract each block to a point. This way we obtain bipartite graphs $\mathfrak{Y}_1$ and $\mathfrak{Y}_2$ on the vertex set $(\mathfrak{B}_1, \mathfrak{B}_2)$. We note that the density of this bipartite graph is less than $1/2$ since the number of edges of $\mathfrak{Y}_i$ is at most the number of edges of $\mathfrak{X}_i$ which is $|\Delta_i| = |\Omega_i| < |\mathfrak{B}_i|^2/2$ (because P_i is a giant and therefore $|\mathfrak{B}_i| > 2\sqrt{n}$ while $|\Omega_i| \leq n$).

Let us now use a graph isomorphism routine to determine $\mathrm{ISO}(\mathfrak{Y}_1, \mathfrak{Y}_2)$ (isomorphisms respect the parts by definition). Let $H\tau$ be the subcoset of these isomorphisms. Let $\varphi : G \to \mathrm{Sym}(\mathfrak{B}_1) \times \mathrm{Sym}(\mathfrak{B}_2)$ be the G-action on the blocks; and let $G_2\rho = \varphi^{-1}(H\tau)$. Then $\mathrm{ISO}(G, \mathfrak{X}_1, \mathfrak{X}_2) = \mathrm{ISO}(G_2\rho, \mathfrak{X}_1, \mathfrak{X}_2)$. Therefore, if the G_2-action on either $\mathfrak{B}_i$ is not a giant, we recurse.

We may thus assume now that the G_2-action on each $\mathfrak{B}_i$ is a giant.

Let $G^* = \varphi(G)$ be the image of the G-action on $\mathfrak{B}_1 \dot\cup \mathfrak{B}_2$. It follows that $G^* \leq \mathrm{Aut}(\mathfrak{Y}_i)$ and G^* projects onto $\mathrm{Alt}(\mathfrak{B}_i)$ or $\mathrm{Sym}(\mathfrak{B}_i)$.

Therefore, by Lemma 2.8, $\mathfrak{Y}_i$ is a perfect matching between $\mathfrak{B}_1$ and $\mathfrak{B}_2$; the actions of G^* on $\mathfrak{B}_1$ and $\mathfrak{B}_2$ are linked; and the edges of $\mathfrak{Y}_i$ join precisely the linked pairs of vertices.

It follows that $\mathfrak{Y}_1 = \mathfrak{Y}_2$, both of them being the perfect matching corresponding to the diagonal action of G^*.

Let now $\mathfrak{B}_j = \{B_{j1}, \ldots, B_{jt}\}$ where the diagonal action matches B_{1i} to B_{2i}. We further consider this case next.

5.4 Giant action with matched blocks

The conclusion of the preceding section is that all edges of $\mathfrak{X}_j$ go between matched blocks B_{1i} to B_{2i}.

Let $C_i = B_{1i} \dot\cup B_{2i}$. Let us now mimic the GIANTFINISH routine applied to the G-invariant partition $\mathfrak{C} = \{C_1, \ldots, C_t\}$.

After steps (GF1) to (GF3), we may now assume that G is permutationally isomorphic to $H \wr A_t$ where H is the action of G_t on C_t.

Finally, to perform Step (GF4), we can mimic the WREATH_ALT procedure, using, as in Section 4.4, the STEPUP trick to reduce to the case when $G = H \wr S_t$. But now, $\mathrm{ISO}(G, \Delta_1, \Delta_2)$ can be found by copying the procedure WREATH_SYM (Section 4.4).

5.5 Analysis

The correctness of the algorithm follows the lines of Section 4.7.

The timing analysis follows the lines of Section 4.8 with the following modifications.

Let $L(n, m_1, m_2)$ denote the maximum number of quasi-leaves visited during the execution of the algorithm, where $|\Omega_i| \leq m_i$.

The new aspect of the analysis of the GIANTFINISH algorithm is the inclusion of a Graph Isomorphism routine. Given that Graph Isomorphism can be solved in $\exp(O(\sqrt{n}\log n))$, the cost of each invocation of the modified GIANTFINISH remains $\exp(O(\sqrt{n}\log n))$. Consequently, the overall running time bound will still be of the form

$$T(n) \leq \exp\left(O(\sqrt{n}\log n)\right) L(n, n, n). \tag{20}$$

We need to perform the divide-and-conquer recurrences both for m_1 and for m_2; the result will be an upper bound $L^*(n, m_1, m_2) \geq L(n, m_1, m_2)$ of the form

$$L^*(n, m_1, m_2) = (m_1 m_2)^{e(n)+1}. \tag{21}$$

Compared to the result of Section 4.8, this only changes the constant implied by the big-Oh notation in the final $\exp(O(\sqrt{n}\log n))$ bound.

6 Appendix: the Giant Action Theorem

In this section we give a full description and proof of the group theoretic results on which the algorithm rests. The results originate from [Ba4] where complete proofs of Lemma 6.2 and of the "abridged" version of the Giant Action Theorem 2.4 were given. The proof of the "unabridged" version (below) is a slight extension of the proof given in [Ba4] and was given in full in [BBT]. Lemma 6.2 and Theorem 2.4 were stated without proof in [BKaL, Sec. 10].

Let G be a group. $R \leq G$ is a *complement* to $K \leq G$ if $R \cap K = \{1\}$ and $RK = G$; if $K \triangleleft G$ then it follows that $R \cong G/K$. For $G \leq \mathrm{Sym}(\Omega)$ and a G-invariant partition $\mathfrak{B} = \{\Omega_1, \ldots, \Omega_t\}$ of Ω, we say that $\sigma \in G$ is *clean* if for every $i \leq t$, if $\Omega_i^\sigma = \Omega_i$ then σ fixes Ω_i pointwise. A *clean subgroup* consists of clean elements. Let K be the kernel of the G-action on $\mathfrak{B}$. We say that R is a *clean complement* to K if R is a complement to K and it is clean w. r. t. $\mathfrak{B}$.

We note that the term "clean" (element/subgroup/complement) is not standard and was introduced by the author for the purpose of the results in this section.

The observation below is straightforward from the definition of a clean complement.

Observation 6.1. *If R is a clean complement to K w. r. t. the system of imprimitivity $\mathfrak{B}$ then for every $\Omega_i \in \mathfrak{B}$, the restriction of the setwise stabilizer $G_{\{\Omega_i\}}$ to Ω_i is the same as the restriction of K to Ω_i.* □

The following lemma establishes the existence of a (clean) complement to the kernel of a giant action. It differs in spirit from the common types of complement theorems in group theory that typically establish the existence of a *normal* complement to a subgroup that is not normal; our case goes the other way around.

Lemma 6.2. [Ba4, Lemma 21.13] *Let $G \leq \mathrm{Sym}(\Omega)$ and let $\mathfrak{B} = \{\Omega_1, \ldots, \Omega_t\}$ be a G-invariant partition of Ω. Suppose the image of the G-action $G \to \mathrm{Sym}(\mathfrak{B})$ is $\mathrm{Alt}(\mathfrak{B})$; let K be the kernel of this action. Assume further that $t > 2\sqrt{n}$ where $n = |\Omega|$. Then K has a clean complement.*

Proof: For $\tau \in G$ let $\overline{\tau}$ denote the action of τ on $[t] = \{1, \ldots, t\}$ corresponding to the action of τ on $\mathfrak{B}$. Let $b = n/t$; so $|\Omega_i| = b$ for all i; and we have $t > 4b$. By Bertrand's postulate there is a prime p between $b+1$ and $2b+1$. Take $\pi \in G$ such that $\overline{\pi}$ is a p-cycle. As $b < p$, there is an m such that p does not divide m and π^m is clean. We may assume $m = 1$, and $\overline{\pi} = (1, 2, \ldots, p)$. Similarly, there exists a clean $\pi' \in G$ such that $\overline{\pi'} = (p, p+1, \ldots, 2p-1)$. Thus the commutator $\sigma := [\pi, \pi']$ is a clean permutation such that $\overline{\sigma}$ is a 3-cycle. Let σ_i be a conjugate of σ with $\overline{\sigma}_i = (i, i+1, t)$. (Note that conjugates of clean elements are clean.) It is easy to see that for t odd, the group generated by $\sigma_1, \sigma_3, \sigma_5, \ldots, \sigma_{t-2}$ is a clean complement to K. If t is even then let A denote the stabilizer of the block Ω_t in the group $\langle \sigma_1, \sigma_3 \rangle$. Clearly, $A \cong A_4$. Now $A, \sigma_4, \sigma_6, \ldots, \sigma_{t-2}$ generate a clean complement to K. □

Note that a clean complement does not necessarily exist if the G-action on $\mathfrak{B}$ is $\mathrm{Sym}(\mathfrak{B})$, even if t is as large as $n/2$, as the following example shows. Let $t = n/2$, $b = 2$ and $G \cong S_t$ and $\Omega = [t] \times (1, -1)$. Let $\Omega_i = \{(i, 1), (i, -1)\}$ $(1 \leq i \leq t)$. For each $\sigma \in S_t$ let $\sigma^* \in G$ be defined by such that $(i, j)^{\sigma^*} = (i^\sigma, j \,\mathrm{sgn}(\sigma))$ $(j = \pm 1 \; ; \; 1 \leq i \leq t)$. Note that K is the identity in this case and it does not have a clean complement, since each element of G which acts on $\mathfrak{B}$ as an odd permutation performs a transposition on the fixed blocks.

A slight extension of the proof of Lemma 6.2 yields the following generalization.

Lemma 6.3. *Let $G \leq \mathrm{Sym}(\Omega)$ be a permutation group. Let $\{T_1, \ldots, T_r\}$ be a partition of Ω into G-invariant subsets; and let $\mathfrak{B}_i$ be a G-invariant partition of T_i. Assume each block in $\mathfrak{B}_i$ has the same size b_i; so $b_i t_i = |T_i|$,*

where $t_i = |\mathfrak{B}_i|$ is the number blocks in T_i. Assume that the G-action on $\dot{\bigcup}_{i=1}^{r} \mathfrak{B}_i$ is $\mathrm{Alt}(\mathfrak{B}_1) \times \cdots \times \mathrm{Alt}(\mathfrak{B}_r)$; let K be the kernel of this action. Assume further that $\min_i t_i > 4 \max_i b_i$. Then K has a clean complement in G. □

Assume that $G \leq \mathrm{Sym}(\Omega)$ is a permutation group satisfying the conditions of Lemma 6.2. As K has a clean complement R, we can identify Ω with the set $\Delta \times [t]$, where $|\Delta| = |\Omega_i|$ for all i. We identify Ω_i with $\Delta \times \{i\}$. For $\sigma \in K$, let σ_i denote the restriction of σ to Ω_i, and for $\sigma \in \mathrm{Sym}(\Delta)$, let $\widetilde{\sigma}$ act on $\Omega = \Delta \times [t]$ by permuting the first components. For $S \leq \mathrm{Sym}(\Delta)$ and $i \leq t$, let $S_i = \{\widetilde{\sigma}_i : \sigma \in S\}$ and let $S^t = S_1 \times \cdots \times S_t$.

Let $H_1 \leq \mathrm{Sym}(\Omega_1)$ be the restriction of K to Ω_1, and let H be the corresponding subgroup of $\mathrm{Sym}(\Delta)$. Then $K \leq H_1 \times \ldots H_t$, where H_i is the copy of H acting on Ω_i; and K projects onto each H_i. Note also that H_i is the restriction of the setwise stabilizer of Ω_i in G to Ω_i.

The following theorem appears as [BBT, Thm. 3.4] except for item 6 which has been added at this time. It is the complete version of our Theorem 2.4 which appears as [Ba4, Lemma 21.15].

Theorem 6.4. (Giant Action Theorem - unabridged) *We use the notation of the previous paragraphs. Let $G \leq \mathrm{Sym}(\Omega)$ be a group satisfying the conditions of Lemma 6.2; in particular, the kernel K of the G-action on the blocks has a clean complement $R \cong A_t$. Moreover, H (the restriction of K to a block) has normal subgroups $N^* \leq N \leq H$, and G has normal subgroups M and M^*, $M^* \leq M \leq K$, such that*

1. *$M = N^t \cap K$ and $M^* = (N^*)^t$.*
2. *The restriction of M to $\Omega \backslash \Omega_i$ is $N^{t-1} = N_1 \times \cdots \times N_{i-1} \times N_{i+1} \times \cdots \times N_t$.*
3. *K/M is the diagonal of $(H/N)^t$; in particular, $K/M \cong H/N$.*
4. *N/N^* is abelian.*
5. *$M/M^* = \{(\sigma_1, \ldots, \sigma_t) : \sigma_i \in N/N^*, \prod_{i=1}^{t} \sigma_i = 1\}$.*
6. *$G/M = (K/M) \times (RM/M)$. In particular, $RM \lhd G$.*

Proof: Let $N := \{\sigma \in H : \exists \sigma_3, \ldots, \sigma_t \in H : (1, \sigma, \sigma_3, \ldots, \sigma_t) \in K\}$ and $N^* := \{\sigma \in H : (\sigma, 1, \ldots, 1) \in K\}$. Clearly $N^* \leq N \leq H$ and both N and N^* are normal in H. In order to prove item *1*, we need to verify that

(1a) $M \lhd G$

(1b) $M^* \leq K$

(1c) $M^* \triangleleft G$

It is evident that both M and $M^* \cap K$ are normalized by K as well as by R and are therefore normal in $G = KR$. We only need to prove item (1b); this follows using R-conjugates of the elements of K in the definition of N^*. $\square$

Claim 6.5. $\sigma \in N \iff (\exists(\sigma_1, \ldots, \sigma_t) \in K)(\exists i, j)(\sigma_i = \sigma, \sigma_j = 1)$.

Proof. Evident from the definition by conjugating by an appropriate element of R. $\square$

Claim 6.6. $\sigma \in N \iff (\sigma, \sigma^{-1}, 1, \ldots, 1) \in K$.

Proof. ($\Leftarrow$) Immediate from Claim 6.5.
($\Rightarrow$) Assume that $\widehat{\sigma} = (1, \sigma, \sigma_3, \ldots, \sigma_t) \in K$. First we show that some element of the form $(1, \sigma, 1, \sigma'_4, \ldots, \sigma'_t)$ also belongs to K. Indeed let $\tau \in R$ permute the blocks as the 3-cycle $(1, 2, 3)$ 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 R we get ϱ' which has $(1, \sigma, 1)$ in the first three positions. Now $\varrho'' = [\varrho', \tau^{-1}] = (\sigma, \sigma^{-1}, 1, \ldots, 1)$. $\square$
The Claim 6.6 clearly implies *2*, using the definition of M in item *1*.

Claim 6.7. $\sigma \in N \iff (\exists \sigma_2 \in N)(\exists \sigma_3, \ldots, \sigma_t \in H)((\sigma, \sigma_2, \ldots, \sigma_t) \in K$.

Proof. ($\Rightarrow$) Evident from Claim 6.5.
($\Leftarrow$) By Claim 6.6 we have $(\sigma_2, \sigma_2^{-1}, 1, \ldots, 1) \in K$. Taking the product of this and $(\sigma, \sigma_2, \ldots, \sigma_t) \in K$, we obtain that $(\sigma\sigma_2, 1, \sigma_3, \ldots, \sigma_t) \in K$, therefore, by definition, $\sigma\sigma_2 \in H$ and so $\sigma \in H$. $\square$

3: Let us consider the homomorphism $\varphi : K \to H/N$ defined by

$$\varphi(\sigma_1, \ldots, \sigma_t) = \sigma_2 N. \tag{22}$$

It suffices to show that $\ker(\varphi) = M$. Obviously $\ker(\varphi) \geq M$. On the other hand, let $(\sigma_1, \ldots, \sigma_t) \in \ker(\varphi)$. This implies that $\sigma_2 \in N$. It follows by Claim 6.7 that all the σ_i belong to N and therefore $(\sigma_1, \ldots, \sigma_t) \in M$.

4: It suffices to prove that $N^* \geq [N, N]$. Assume that $\sigma, \sigma' \in N$. This implies that the following elements belong to K: $(\sigma, \sigma^{-1}, 1, \ldots, 1)$,

$(\sigma', 1, \sigma'^{-1}, 1, \ldots, 1)$, $(\sigma^{-1}, \sigma, 1, \ldots, 1)$, and $(\sigma'^{-1}, 1, \sigma', 1, \ldots, 1)$. The product of these elements is $([\sigma, \sigma'], 1, \ldots, 1) \in K$, which means that $[\sigma, \sigma'] \in N^*$.

5: Suppose $(\sigma_1, \ldots, \sigma_t) \in M/M^*$. (The σ_i are cosets of N^*.) As $\sigma_i \in N/N^*$ for each $1 \leq i \leq t-1$, there is a $\pi(i) \in M/M^*$ with $(\pi(i))_i = {\sigma_i}^{-1}$, $(\pi(i))_t = \sigma_i$ and $(\pi(i))_j = 1$ otherwise. So we have that

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

but then $\prod_{i=1}^{t} \sigma_i = N^*$ by the definition of N^*, proving *5*.
We now prove item *6* through a series of claims.

Claim 6.8. *$[K, R] \leq M$.*

Proof. In the light of the identity $[u, xy] = [u, y][u^y, x^y]$ (valid in any group), we only need to prove that $[\sigma, \tau] \in M$ where $\sigma \in K$ and τ belongs to a set T of generators of R. Let T consist of the elements of R corresponding to 3-cycles in A_t. Let $i \leq t$ correspond to a fixed point of $\tau \in T$. Then $[\sigma, \tau] \in K$ has the indentity in the i-th position and therefore all components of $[\sigma, \tau]$ belong to N and so $[\sigma, \tau] \in M$. □

Claim 6.9. *$RM \triangleleft G$.*

Proof. Because of the identity $[a, b] = a^{-1}a^b$, a subgroup $U \leq G$ is normal exactly if $[U, G] \leq U$. Now $[RM, G]$ is generated by $[R, G]$ and $[M, G]$. We have $[M, G] \leq M$, and $[R, G] = [R, RK]$ is generated by $[R, R] \leq R$ and $[R, K] \leq M$, so $[RM, G] \leq RM$. □

Claim 6.10. *$K \cap RM = M$.*

Proof. It is easy to verify that for any three subgroups A, B, C of a group D, if $A \leq B$ and $B \cap C = 1$ then $B \cap CA = A$. (Here $AC = \{ac \mid a \in A, c \in C\}$ is not necessarily a subgroup.) Apply this in the roles $A = M$, $B = K$, $C = R$. □
It follows that $|K/M \cap RM/M| = 1$. Since K/M and RM/M are normal sugroups of G/M that generate G/M, we have $G/M = (K/M) \times (RM/M)$, completing the proof of item *6*. □

Now Theorem 2.4 (the "abridged" Giant Action Theorem) is immediate. Indeed, using the notation of Theorem 6.4, (a) and (b) follow by definition. (c) is stated as Item 3 in Theorem 6.4. (d) follows from Item 2 in Theorem 6.4 and the fact that R is a clean complement to K. □

References

[Ba1] L. Babai: Monte Carlo algorithms in graph isomorphism testing. Université de Montréal Tech. Rep. DMS 79-10, 1979 (pp. 42) `http://people.cs.uchicago.edu/~laci/lasvegas79.pdf`

[Ba2] L. Babai: On the order of uniprimitive permutation groups, *Annals of Math.* **113** (1981), 553–568.

[Ba3] L. Babai: On the order of doubly transitive permutation groups, *Inventiones Math.* **65** (1982), 473–484.

[Ba4] L. Babai: *Permutation Groups, Coherent Configurations and Graph Isomorphism.* D.Sc. Thesis (Hungarian), Hungarian Academy of Sciences, April 1983.

[Ba5] L. Babai: On the length of subgroup chains in the symmetric group. *Communications in Algebra* **14** (1986), 1729–1736.

[BBT] L. Babai, R. Beals, P. Takácsi-Nagy: Symmetry and complexity. In: *Proc. 24th STOC*, ACM 1992, pp. 438–449.

[BCaP] L. Babai, P. J. Cameron, P. P. Pálfy: On the orders of primitive groups with restricted nonabelian composition factors. *J. Algebra* **79** (1982), 161–168.

[BCo] L. Babai, P. Codenotti: Isomorphism of 3-graphs in moderately exponential time. In preparation.

[BKaL] L. Babai, W. M. Kantor, E. M. Luks: Computational complexity and the classification of finite simple groups. In: *Proc. 24th FOCS*, IEEE Computer Society Press, 1983, pp. 162–171.

[BKlL] L. Babai, P. Klingsberg, E. M. Luks: Distributive lattice isomorphims is in polynomial time. 1983. (unpublished)

[BL] L. Babai, E. M. Luks: Canonical labeling of graphs. In: *Proc. 15th STOC*, ACM 1983, pp. 171–183.

[Cam] P. J. Cameron: Finite permutation groups and finite simple groups, *Bull. London Math Soc.* **13** (1981), 1–22.

[FHL] M. L. Furst, J. Hopcroft, E. M. Luks: Polynomial-time algorithms for permutation groups. In: *21st FOCS*, IEEE Computer Soc. 1980, pp. 36-41.

[Ka] W. M. KANTOR: Sylow's theorem in polynomial time. *J. Computer and System Sci.* **30** (1985), 359–394.

[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: Computing the composition factors of a permutation group in polynomial time. *Combinatorica* **7** (1987), 87–99.

[Lu3] E. M. LUKS: Permutation groups and polynomial-time computation. In *Groups and Computation*, DIMACS series in Discrete Mathematics and Theoretical Computer Science **11** (1993), 139–175.

[Lu4] E. M. LUKS: Hypergraph isomorphism and structural equivalence of Boolean functions. In: *Proc. 31st STOC*, ACM Press, 1999, pp. 652–658.

[Py] L. PYBER: The orders of doubly transitive groups, elementary estimates. *J. Comb. Theory, Ser. A* **62** (1993), 361–366.

[Ro] J. J. ROTMAN: *The Theory of Groups, An Introduction.* Allyn and Bacon, Boston, 1973.

[Se] Á. SERESS: *Permutation Group Algorithms.* Cambridge University Press, 2003.

[Si1] C. C. SIMS: Computation with Permutation Groups. In: *Proc. 2nd Symp. Symb. Algeb. Manip.* (S. R. Petrick,ed.), ACM, New York, 1971, pp. 23–28.

[Si2] C. C. SIMS: Some group theoretic algorithms. In: *Lecture Notes in Math.* Vol. 697, Springer, 1978, pp. 108-124.

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

[ZKT] V. N. ZEMLYACHENKO, N. KORNIENKO, R. I. TYSHKEVICH: The Graph Isomorphism Problem (in Russian). *The Theory of Computation I*, Notes Sci. Sem. LOMI 118, Leningrad 1982.