Graph Isomorphism in Quasipolynomial Time

[Extended Abstract][*]

László Babai
University of Chicago, USA
laci@cs.uchicago.edu

ABSTRACT

We show that the Graph Isomorphism (GI) problem and the more general problems of String Isomorphism (SI) and Coset Intersection (CI) can be solved in quasipolynomial $(\exp((\log n)^{O(1)}))$ time. The best previous bound for GI was $\exp(O(\sqrt{n \log n}))$, where n is the number of vertices (Luks, 1983); for the other two problems, the bound was similar, $\exp(\widetilde{O}(\sqrt{n}))$, where n is the size of the permutation domain (Babai, 1983).

Following the approach of Luks's seminal 1980/82 paper, the problem we actually address is SI. This problem takes two strings of length n and a permutation group G of degree n (the "ambient group") as input (G is given by a list of generators) and asks whether or not one of the strings can be transformed into the other by some element of G.

Luks's divide-and-conquer algorithm for SI proceeds by recursion on the ambient group. We build on Luks's framework and attack the obstructions to efficient Luks recurrence via an interplay between local and global symmetry. We construct group theoretic "local certificates" to certify the presence or absence of local symmetry, aggregate the negative certificates to canonical k-ary relations where $k = O(\log n)$, and employ combinatorial canonical partitioning techniques to split the k-ary relational structure for efficient divide-and-conquer. We show that in a well–defined sense, Johnson graphs are the only obstructions to effective canonical partitioning. The central element of the algorithm is the "local certificates" routine which is based on a new group theoretic result, the "Unaffected stabilizers lemma," that allows us to construct global automorphisms out of local information.

Keywords

Algorithms, complexity of computation, graphs, graph isomorphism, group theory, divide and conquer

Categories and Subject Descriptors

F.2.2 [**Nonnumerical algorithms**]: Computations on discrete structures—*Graph Isomorphism, Permutation groups*; G.2.2 [**Graph Theory**]: Graph algorithms—*Automorphism groups, Group theory, Coherent configurations*

1. INTRODUCTION

1.1 Notation, Fraktur, Group Theory Reference

We shall use a number of Fraktur characters. In particular, $\mathfrak{x}$, $\mathfrak{y}$, $\mathfrak{z}$ will denote strings, $\mathfrak{S}_n$ the symmetric group of degree n, $\mathfrak{S}(\Omega)$ the symmetric group acting on the set Ω (the group of all permutations of Ω), $\mathfrak{A}_n$ the alternating group of degree n, and $\mathfrak{A}(\Omega)$ the alternating group acting on the set Ω (the group of even permutations of Ω).

Here is a list of the Roman equivalents of the Fraktur characters we use:
$\mathfrak{x}$ – x, $\mathfrak{y}$ – y, $\mathfrak{z}$ – z, $\mathfrak{A}$ – A, $\mathfrak{S}$ – S, $\mathfrak{X}$ – X, $\mathfrak{Y}$ – Y

A *permutation group* acting on the set Ω is a subgroup $G \leq \mathfrak{S}(\Omega)$. (The "$\leq$" sign stands for "subgroup.") The *degree* of G is $|\Omega|$, the size of the *permutation domain* Ω.

For $x \in \Omega$ and $\sigma \in G$ we write x^σ for the σ-image of x. The set $x^G = \{x^\sigma \mid \sigma \in G\}$ is the G-*orbit* of x. We say that G is *transitive* if $x^G = \Omega$ for some (and therefore any) $x \in \Omega$. We note that the orbits of a normal subgroup of a transitive group have equal length. A transitive group $G \leq \mathfrak{S}(\Omega)$ is *primitive* if $|\Omega| \geq 2$ and there is no nontrivial G-invariant equivalence relation on Ω.

The *stabilizer* G_x of $x \in \Omega$ is the subgroup $G_x = \{\sigma \in G \mid x^\sigma = x\}$ consisting of those elements of G fixing x. The *pointwise stabilizer* of a subset $\Delta \subseteq \Omega$ is the subgroup $G_{(\Delta)} = \bigcap\{G_x \mid x \in \Delta\}$. An *action* of G on a set Δ is a homomorphism $G \to \mathfrak{S}(\Delta)$.

For basic group theory we refer to the introductory chapters of Rotman [Ro]. Our standard reference for permutation groups is Dixon and Mortimer [DiM]. For the polynomial-time algorithmic theory of permutation groups we refer to Seress [Se]. Permutation groups are given by a list of generators. For a permutation group $G \leq \mathfrak{S}(\Omega)$, the following

[*] Full paper available at arXiv:1512.03547. Research supported in part by NSF Grants CCF-7443327 (2014-current), CCF-1017781 (2010-2014), and CCF-0830370 (2008–2010). Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the author and do not necessarily reflect the views of the National Science Foundation (NSF).

STOC'16, June 19–21, 2016, Cambridge, MA, USA
© 2016 ACM. 978-1-4503-4132-5/16/06...$15.00
http://dx.doi.org/10.1145/2897518.2897542

tasks can be completed in polynomial time: testing membership in G; computing $|G|$, the order of G; computing the pointwise stabilizer $G_{(\Delta)}$ for $\Delta \subseteq \Omega$; finding the kernel of an action $G \to \mathfrak{S}(\Delta)$.

1.2 Results: The String Isomorphism Problem

Let G be a group of permutations of the set $[n] = \{1, \dots, n\}$ and let $\mathfrak{x}, \mathfrak{y}$ be strings of length n over a finite alphabet. We say that $\sigma \in \mathfrak{S}_n$ is a G-isomorphism between $\mathfrak{x}$ and $\mathfrak{y}$ if $\sigma \in G$ and $\mathfrak{x}^\sigma = \mathfrak{y}$. The strings $\mathfrak{x}$ and $\mathfrak{y}$ are *G-isomorphic*, denoted $\mathfrak{x} \cong_G \mathfrak{y}$, if such a σ exists. The *String Isomorphism (SI) problem* asks, given G, $\mathfrak{x}$, and $\mathfrak{y}$, does $\mathfrak{x} \cong_G \mathfrak{y}$ hold? (The group G is given by a list of generators.)

A function $f(n)$ is *quasipolynomially bounded* (or "quasipolynomial" for short) if there exist constants c, C such that $f(n) \le \exp(C(\log n)^c)$ for all sufficiently large n.

We prove the following result.

Theorem 1. *The String Isomorphism problem can be solved in quasipolynomial time.*

The Graph Isomorphism (GI) Problem asks to decide whether or not two given graphs are isomorphic. The Coset Intersection (CI) problem asks, given two subcosets of the symmetric group, do they have a nonempty intersection.

Corollary 2. *The Graph Isomorphism problem and the Coset Intersection problem can be solved in quasipolynomial time.*

The SI and CI problems were introduced by Luks [Lu82] (cf. [Lu93]) who also pointed out that these problems are polynomial-time equivalent (under Karp reductions) and GI easily reduces to either. For instance, GI for graphs with v vertices is identical, under obvious encoding, with SI for binary strings of length $n = \binom{v}{2}$ with respect to the induced action of the symmetric group of degree v on the set of n unordered pairs.

The previous best bound for each of these three problems was $\exp(\widetilde{O}(n^{1/2}))$ (the tilde hides polylogarithmic factors[1]), where for GI, n is the number of vertices, for the two other problems, n is the size of the permutation domain. For GI, this bound was obtained in 1983 by combining Luks's group-theoretic algorithm [Lu82] with a combinatorial partitioning lemma by Zemlyachenko (see [ZKT, BaL, BaKL]). For SI and CI, additional group-theoretic observations were used ([Ba83], cf. [BaKL]). No improvement over either of these results was found in the intervening decades.

The actual result we obtain is slightly stronger than stated above; only the length of the largest orbit of G matters.

Theorem 3. *The SI problem can be solved in time, polynomial in n (the length of the strings) and quasipolynomial in $n_0(G)$, the length of the largest orbit of the ambient group.*

The first class of graphs for which GI was studied using group theory was that of vertex-colored graphs (isomorphisms preserve color by definition) with small color multiplicities [Ba79a] (1979).

Corollary 4. *The GI problem for vertex-colored graphs can be solved in time, polynomial in n (the number of vertices) and quasipolynomial in the largest color multiplicity.*

1.3 Unaffected Stabilizers Lemma

In this section we describe the group theoretic lemma that is the key mathematical tool of the algorithm.

We refer to $\mathfrak{S}(\Omega)$ and $\mathfrak{A}(\Omega)$ as the *giants* among the permutation groups acting on Ω.

For a group G and a set Γ we say that the action $\varphi : G \to \mathfrak{S}(\Gamma)$ is a *giant representation* of G (or a *giant homomorphism*) if G^φ (the image of G under φ) is a giant, i. e., $G^\varphi \ge \mathfrak{A}(\Omega)$.

We define the central new concept of this paper.

Definition 5 (Affected). *Let Ω and Γ be sets, $G \le \mathfrak{S}(\Omega)$, and let $\varphi : G \to \mathfrak{S}(\Gamma)$ be a giant representation. We say that $x \in \Omega$ is* affected *by φ if the φ-image of the stabilizer G_x is not a giant, i. e., $(G_x)^\varphi \not\ge \mathfrak{A}(\Gamma)$.*

We note that if $x \in \Omega$ is affected then every element of the orbit x^G is affected. So we can speak of *affected orbits.*

The affected/unaffected dichotomy underlies the core "local certificates" routine and is the central divide-and-conquer tool of the algorithm through a group theoretic lemma we state next.

Theorem 6. *Let $G \le \mathfrak{S}(\Omega)$ be a permutation group and let n_0 denote the length of the largest orbit of G. Let $\varphi : G \to \mathfrak{S}_k$ be a giant representation, i. e., $G^\varphi \ge \mathfrak{A}_k$.*

Let $U \subseteq \Omega$ denote the set of elements of Ω not affected by φ. Then the following hold.

(a) (Unaffected Stabilizers Lemma) *Assume $k > \max\{8, 2 + \log_2 n_0\}$. Then φ restricted to $G_{(U)}$, the pointwise stabilizer of U, is still a giant representation, i. e., $(G_{(U)})^\varphi \ge \mathfrak{A}_k$. In particular, $U \ne \Omega$ (at least one element is affected).*

(b) (Affected Orbit Lemma) *Assume $k \ge 5$. If Δ is an affected G-orbit, i. e., $\Delta \cap U = \emptyset$, then* $\ker(\varphi)$ *is not transitive on Δ; in fact, each orbit of* $\ker(\varphi)$ *in Δ has length $\le |\Delta|/k$.*

Part (b) is an easy exercise. The proof of part (a) builds on the (elementary) O'Nan–Scott–Aschbacher characterization of primitive permutation groups ([Sco, AsS], cf. [DiM, Thm. 4.1A]) and depends on the classification of Finite Simple Groups (CFSG) through "Schreier's Hypothesis" (a consequence of CFSG) that asserts that the outer automorphism group of every finite simple group is solvable (but see comments in Section 5.1).

Note that part (a) is somewhat counter-intuitive: it asserts that if the stabilizer of each vertex $x \in U$ maps onto A_k or S_k then even the intersection of these stabilizers maps onto A_k or S_k.

The condition $k > \max\{8, 2 + \log_2 n_0\}$ in part (a) is tight. In fact, there are infinitely many examples with $k = 2 + \log_2 n$ which have *no affected points.*

We omit the proof of Theorem 6 from this Extended Abstract but state a lemma that constitutes the first part of the proof of the Unaffected Stabilizers Lemma.

A permutation group $G \le \mathfrak{S}(\Omega)$ is *primitive* if it is transitive and admits no nontrivial G-invariant partition of Ω.

Lemma 7. *Under the assumptions of the Unaffected Stabilizers Lemma, if G is primitive then φ is injective (and therefore $G \cong \mathfrak{A}_k$ or $G \cong \mathfrak{S}_k$).*

[1] Accounting for those logs, the best bound for GI for more than three decades was $\exp(O(\sqrt{n \log n}))$, established by Luks in 1983, outlined in [BaKL].

The algorithmic role of the Unaffected Stabilizers Lemma will be to allow us to construct global automorphisms out of local information. The discovery of this connection (Sep. 14, 2015) was the turning point of this long-running project. In the rest of this paper we shall try to indicate how to connect the dots.

2. STRATEGY

2.1 Quasipolynomial Complexity Analysis, Multiplicative Cost

Throughout this paper, we use the term "efficient" to mean "quasipolynomial time" or leading to a recurrence that resolves to quasipolynomial time.

The analysis will be guided by the observation that if $f(x) \geq 1$ and $q(x) \geq 1$ are monotone increasing functions and

$$f(n) \leq q(n) f(9n/10) \qquad (1)$$

for all sufficiently large n then $f(n) \leq q(n)^{O(\log n)}$. In particular, if $q(x)$ is quasipolynomially bounded then so is $f(x)$. We apply this recurrence with $f(n)$ being the worst-case cost (number of group operations) on instances of size $\leq n$ and $q(n)$ is a bound on the branching factor in the algorithm to which we refer as the "multiplicative cost" in reference to Eq. (1). So our goal will be to achieve a *significant reduction* in the problem size, say $n \leftarrow 9n/10$, at a quasipolynomial multiplicative cost. There is also an additive cost to the reduction, but this will typically be absorbed by the multiplicative cost.

Both in the group-theoretic and in the combinatorial arguments, we shall actually use double recursion. In addition to the input domain Ω of size $n = |\Omega|$, we shall also build an auxiliary set Γ and track its size $m = |\Gamma| \leq n$. Most of the action will occur on Γ. Significant progress will be deemed to have occurred if we significantly reduce Γ, say $m \leftarrow 9m/10$, while not increasing n. When m drops below a threshold $\ell(n)$ that is polylogarithmic in n, we perform brute force enumeration over the symmetric group $\mathfrak{S}(\Gamma)$ at a multiplicative cost of $\ell(n)!$. This eliminates the current Γ and significantly reduces n. Subsequently a new Γ, of size $m \leq n$, is introduced, and the game starts over. If $q_1(x)$ is the multiplicative cost of significantly reducing Γ then the overall cost estimate becomes $f(n) \leq q_1(x)^{\log^2 n}$, quasipolynomial as long as $q_1(n)$ is quasipolynomial.

2.2 Obstruction to Efficient Luks Recurrence

2.2.1 Strings: notation

We shall denote the set of positions by Ω; with this notation, a *string* is a map $\mathfrak{x} : \Omega \to \Sigma$ where Σ is a finite alphabet. The ambient group $G \leq \mathfrak{S}(\Omega)$ acts on the set of positions and has an induced action $\mathfrak{x} \mapsto \mathfrak{x}^\sigma$ $(\sigma \in G)$ on the set of strings, defined by $\mathfrak{x}^\sigma(x) = \mathfrak{x}(x^{\sigma^{-1}})$ $(x \in \Omega)$. Rather than just solving the decision problem $\mathfrak{x} \overset{?}{\cong}_G \mathfrak{y}$, we wish to compute the set $\mathrm{Iso}_G(\mathfrak{x}, \mathfrak{y}) = \{\sigma \in G \mid \mathfrak{x}^\sigma = \mathfrak{y}\}$ of G-isomorphisms between the strings $\mathfrak{x}$ and $\mathfrak{y}$. Following Luks, we do this by recursion on G.

2.2.2 Systems of imprimitivity

Let $G \leq \mathfrak{S}(\Omega)$ be a transitive permutation group. If $\Omega = \Omega_1 \cup \cdots \cup \Omega_k$ is a G-invariant partition, the set $\{\Omega_1, \ldots, \Omega_k\}$ is called a *system of imprimitivity* for G and the Ω_i the *blocks of imprimitivity.* The trivial systems of imprimitivity are the discrete partition $(k = n)$ and the unit partition $(k = 1)$. If there is no nontrivial system of imprimitivity, the group G is said to be *primitive.* In the imprimitive case we have a G-action on the set of blocks: $G \to \mathfrak{S}_k$. If K is the kernel of this action then K is intransitive: the blocks Δ_i are K-invariant so they are further partitioned by the K-orbits.

2.2.3 Luks's framework

Following Luks [Lu82], we attack the SI problem by recursion on the ambient group G. If G is intransitive, we process it orbit by orbit. In particular, we may always assume that G is transitive.

Next we consider a minimal system of imprimitivity (the blocks are maximal; if G was primitive, this means the discrete partition) and reduce the G-isomorphism problem to $|G : K|$ instances of the K-isomorphism problem where K is the kernel of the G-action on the set of blocks. While orbit-by-orbit processing of K-isomorphism provides for efficient recurrence since the K-orbits are of size $\leq n/2$, the multiplicative cost incurred is the index $|G : K|$. This becomes prohibitive if the group G/K which acts as a primitive group on the set of blocks is large (not quasipolynomially bounded) compared to the number of blocks.

2.2.4 The obstruction

In 1981, Cameron's described the structure of the large primitive permutation groups [Cam]. From this description we can infer the situation when a naive implementation of Luks's method ceases to be efficient.

It turns out that the obstruction to efficient Luks recurrence is a giant representation of the ambient group, $\varphi : G \to \mathfrak{S}(\Gamma)$, for some set Γ of size $m := |\Gamma|$ greater than a polylogarithmic threshold (depending on our target exponent); under naive implementation of Luks's algorithm, m will go in the exponent of the running time. This characterization of the obstruction works for any threshold $m \geq 1 + \log_2 n$.

The set Γ and the map φ are constructed in polynomial time. Our contribution is in breaking the $\mathfrak{A}(\Gamma)$ symmetry.

2.3 Further Conceptual Preparation

In this section we discuss additional concepts before we can turn to an outline of the algorithm.

2.3.1 Relational structures, symmetry defect

We introduce one more concept that plays a key role in our divide-and-conquer process.

Let $G \leq \mathfrak{S}(\Omega)$ where $|\Omega| = n$. The *symmetry defect* of G is the smallest value t such that G has an orbit Δ with $|\Delta| = n - t$ such that $\mathfrak{A}(\Delta) \leq G$. (The group $\mathfrak{A}(\Delta) \leq G$ fixes all elements of $\Omega \setminus \Delta$.)

By a "structure" on the set Ω we mean any kind of algebraic or combinatorial structure with underlying set (vertex set) Ω (graph, colored digraph, hypergraph, group, relational structure, etc.). A *k-ary relational structure* on Ω is a pair $\mathfrak{X} = (\Omega, \mathcal{R})$ where $\mathcal{R} = (R_1, \ldots, R_r)$ is a list of k-ary relations, i. e., $R_i \subseteq \Omega^k$.

By the *symmetry defect* of a structure $\mathfrak{X}$ we mean the symmetry defect of $\mathrm{Aut}(\mathfrak{X})$.

If the symmetry defect is t then the *relative symmetry* *defect* is t/n where $n = |\Omega|$. We shall usually omit the term "relative:" if we give a number between 0 and 1 as the "symmetry defect," we refer to the relative symmetry defect.

The symmetry defect of an explicit structure $\mathfrak{X}$ can be determined in polynomial time by checking, for each 3-cycle $\sigma \in \mathfrak{S}(\Omega)$, whether or not $\sigma \in \mathrm{Aut}(\mathfrak{X})$.

2.3.2 Large subgroups of the symmetric group

A theorem of Jordan [Jor1] and Liebeck [Lie83] characterizes the subgroups of small index in $\mathfrak{S}_n$.

THEOREM 8 (JORDAN–LIEBECK). *Let $r < n/2$. If $G \leq \mathfrak{S}_n$ has index $< \binom{n}{r}$ in $\mathfrak{S}_n$ or in $\mathfrak{A}_n$ then the symmetry defect of G is less than r.*

Using the binary entropy value $H(1/4) = 0.811278\ldots$ it follows that if $n!/|G| \leq 1.75^n$ then for sufficiently large n the symmetry defect of H is at most $1/4$.

2.3.3 Canonical assignments

Suppose with each input string $\mathfrak{z} : \Omega \to \Sigma$ we associate a structure $\mathfrak{X}_\mathfrak{z}$ (on a different set; typically the underlying set of $\mathfrak{X}_\mathfrak{z}$ will be the set Γ from the Luks obstruction).

We say that the assignment $\mathfrak{z} \mapsto \mathfrak{X}(\mathfrak{z})$ is *canonical* if all G-isomorphisms $\mathfrak{z}_1 \to \mathfrak{z}_2$ induce isomorphisms $\mathfrak{X}_{\mathfrak{z}_1} \to \mathfrak{X}_{\mathfrak{z}_2}$ for $\mathfrak{z}_i \in \{\mathfrak{x}, \mathfrak{y}\}$. In other words, the assignment $\mathfrak{z} \mapsto \mathfrak{X}_\mathfrak{z}$ defines a *functor* from the category of G-isomorphisms of strings to the category of isomorphisms of the structures under consideration. The category we need to consider is very small, it has only two objects, $\mathfrak{x}$ and $\mathfrak{y}$.

2.3.4 Johnson graphs

The Johnson graph $J(v,t)$ has $n = \binom{v}{t}$ vertices labeled by the t-subsets $T \subseteq [v]$. The t-subsets T_1, T_2 are adjacent if $|T_1 \setminus T_2| = 1$.

Johnson graphs play a sepcial role in this theory because of their high resilience to canonical partitioning (see Prop. 11).

2.4 Outline of the Algorithm

2.4.1 Overview

We follow Luks's algorithm until we hit an obstruction; G is transitive and we have a giant representation $\varphi : G \to \mathfrak{S}(\Gamma)$. We want to find information about the φ-image of the unknown group $\mathrm{Aut}_G(\mathfrak{x})$.

Ideally we would like to significantly reduce the group G (recursion) by reducing its action on Γ. Specifically, our goal is to find a subgroup $\overline{H} \leq \mathfrak{S}(\Gamma)$ (the "encasing group") such that

(a) $\overline{H}$ has index $> c^m$ in $\mathfrak{S}(\Gamma)$ for some constant $c > 1$ (where $m = |\Gamma|$); and

(b) we guarantee $(\mathrm{Aut}_G(\mathfrak{x}))^\varphi \leq \overline{H}$. This will allow us to reduce the G-isomorphism problem to H-isomorphism, where $H = \varphi^{-1}(\overline{H})$, significant progress.

Such an $\overline{H}$ may not exist; $(\mathrm{Aut}_G(\mathfrak{x}))^\varphi$ may have small index in $\mathfrak{S}(\Gamma)$. We shall see (Sec. 3.2) that this is an easy case: by lifting the 3-cycles in $\mathfrak{S}(\Gamma)$, we can decide whether or not $\mathrm{Aut}_G(\mathfrak{x}))^\varphi$ has symmetry defect say $\leq 1/4$ by efficient Luks reduction, and if the answer is positive, we shall have found $\mathrm{Iso}_G(\mathfrak{x}, \mathfrak{y})$ via this reduction.

In the remaining cases we pursue the goal stated above. As an **intermediate goal** we want to find a **canonical k-ary relational structure** $\mathfrak{X}(\mathfrak{x})$ on Γ with $k = O(\log n)$ and with symmetry defect $\geq 1/4$. This will implicitly define our $\overline{H} = \mathrm{Aut}(\mathfrak{X}(\mathfrak{x}))$. Subsequently we shall use $\mathfrak{X}(\mathfrak{x})$ to make such a group explicit.

The intermediate goal is achieved by the LocalCertificates routine. Subsequently, our combinatorial partitioning algorithms will produce the desired encasing of $\mathrm{Aut}_G(\mathfrak{x})^\varphi$. These algorithms attempt to find a good canonical coloring of Γ (no color class greater than $(3/4)m$) or a canonical equipartition of a large color class. Failing these, the *Design Lemma* canonically reduces the k-ary relational structure to binary (a regular graph, which automatically has symmetry defect $\geq 1/2$). The second, the Split-or-Johnson routine will produce a canonically embedded *Johnson graph.* While Johson graphs are highly resilient to canonical partitioning, once found, they provide for dramatic reduction of Γ and thereby are excellent recursive tools.

2.4.2 Individualization, multiplicative cost

All these reductions involve quasipolynomial multiplicative cost through "individualization" of (assignment of unique colors to) various objects. These objects are elements of Γ in the case of the Design Lemma but they are more abstract objects for Split-or-Johnson. Note that individualizing an object from a canonical set of s objects incurs a multiplicative cost of s (we need to repeat the process with all choices).

Due to these individualizations, we shall actually not achieve goal (b). What we shall achieve is that the (pointwise) stabilizer in $\mathrm{Aut}_G(\mathfrak{x})$ of the set S of objects individualized maps into $\overline{H}$, and the index of this stabilizer in $\mathrm{Aut}_G(\mathfrak{x})$ is quasipolynomially bounded. All steps being effective, this will suffice for efficient recurrence.

2.4.3 Local certificates

We explain the idea at the heart of the algorithm. Our strategy is an interplay between local and global symmetry, formalized through a technique we call *"local certificates."* Locality in our context refers to logarithmic-size subdomains of Γ to which we refer as the *test sets.* Fix a number $k = |T| > 2 + \log_2 n$ but not much greater (say $k = O(\log n)$) and make sure $m > 10k$ (otherwise we apply brute force to Γ). The test sets will be the subsets $T \subset \Gamma$ of size $|T| = k$. Given a test set T, we shall test whether or not $\mathrm{Aut}_G(\mathfrak{x})$ acts on T (via φ) as a giant; we call the positive outcome "local symmetry" and say that T is "full" in this case. We shall certify both the presence and the absence of local symmetry. A certificate of non-fullness will be an "encasing subgroup" $M(T) \leq \mathfrak{S}(T)$ that is guaranteed to include the action on T of the group $\mathrm{Aut}_{G_T}(\mathfrak{x})$ where G_T is the setwise stabilizer of T in G. Note that this is "local information." On the other hand, our "fullness certificate" necessarily has to be "global," consisting of a subgroup $K(T)$ of the unknown group $\mathrm{Aut}_{G_T}(\mathfrak{x})$ that projects onto $\mathfrak{A}(T)$ or $\mathfrak{S}(T)$ (via φ). Both types of certificates are built in an iterative process ("growing the beard") that relies on the affected/unaffected dichotomy. Let $\psi_T : G_T \to \mathfrak{S}(T)$ be the restriction of the domain φ to G_T and then its range to $\mathfrak{S}(T)$. This is a giant representation. The first layer W of the "beard" will consist of the points in Ω affected by ψ_T. Then we reduce G_T to the automorphism group $A(W)$ of the string $\mathfrak{x}^W$, the restriction of $\mathfrak{x}$ to W. This computation is feasible via efficent Luks reduction by the "Affected orbit lemma" (part (b) of Thm. 6). We replace G_T by $A(W)$ and repeat. We stop when one of the following occurs: (a) ψ_T no longer maps $A(W)$ on a giant; or (b) all points of Ω outside W are unaffected by the updated $A(W)$ (the "beard stops growing"). In case (a), the ψ_T-image of $A(W)$ is our non-fullness certificate. In case (b), we cannot use $A(W)$ as a fullness certificate since $A(W) \not\leq \mathrm{Aut}_G(\mathfrak{x})$ ($A(W)$ only respects the partial input $\mathfrak{x}^W$). Here comes the Unaffected Stabilizer lemma to rescue: let $K(T) = (A(W))_{(U)}$ where $U = \Omega \backslash W$ is the set of unaffected points. Then $K(T) \leq \mathrm{Aut}_G(\mathfrak{x})$ (since the input positions not respected by $A(W)$ are now fixed and therefore respected); and $K(T)$ still maps onto $\mathfrak{S}(T)$ or $\mathfrak{A}(T)$.

A detailed explanation and pseudocode of this procedure will be given in Sec. 4.1.

2.4.4 Aggregation

Let F be the group generated by the fullness certificates and let $S \subseteq \Gamma$ be the support of F^{φ}, i. e., the set of points of Γ not fixed by F^{φ}. If $0.1m \leq |S| \leq 0.9m$ then we have a good canonical partition of Γ (namely, S and $\Gamma \backslash S$, significant progress. If $|S| \geq 0.9m$ then we can use the permutation group structure of F^{φ} (orbits, blocks of imprimitivity) via some 19th century group theory to create a good partition of Γ.

Finally if $|S| < 0.1m$ then we can handle S via efficient Luks recurrence and need to worry only about $\Gamma \setminus S$; so we may assume $S = \emptyset$, i. e., none of the test sets is full, so we have a non-giant group $M(T)$ on every test set T.

In fact we shall not only study test sets T but also compare pairs of test sets T, T', and we also compare test sets for input $\mathfrak{x}$ with test sets for input $\mathfrak{y}$. We take this $\binom{m}{k}^2$ pieces of local information as input and combine them to a canonical k-ary relation with symmetry defect $\geq m - k + 1 > 0.9m$.

We then apply the Design Lemma and subsequently the Split-or-Johnson routine to this k-ary relation to significantly reduce the ambient group at quasipolynomial multiplicative cost.

2.4.5 Combinatorial partitioning I: Design Lemma

Let us consider a coloring $f : \Omega \to$ colors. We say that color j is *dominant* if the color class $\Delta = f^{-1}(j)$ of points of color j has size $|\Delta| > (3/4)|\Omega|$.

We say that a coloring of Ω is *good* if there is no dominant color.

An *equipartition* of a set is a partition to equal parts. The trivial equipartitions are the discrete partition and the unit partition.

A *good equipartition* of a colored set Ω is a nontrivial equipartition of a dominant color class.

Canonicity of an assigment of a structure $u(\mathfrak{X})$ to a class of studctures $\mathfrak{X}$ means, as before, that isomorphisms $\mathfrak{X} \to \mathfrak{Y}$ induce isomorphisms of the associated structures, $u(\mathfrak{X}) \to u(\mathfrak{Y})$), i. e., we have a functor from the category of isomorphisms of the structures $\mathfrak{X}$ to the category of isomorphisms of the associated structures.

A good canonical equipartition implicitly assumes that the set $\Delta \subseteq \Omega$ that is being equipartitioned is canonical.

A good canonically embedded graph is a canonically assigned graph whose vertex set is a dominant color class in a canonical coloring.

THEOREM 9 (DESIGN LEMMA). *Let $\mathfrak{X} = (\Omega, \mathcal{R})$ be a k-ary relational structure with $n = |\Omega|$ vertices, $2 \leq k \leq n/2$, and symmetry defect $\geq 1/4$. Then in time $n^{O(k)}$ we can find a sequence S of at most $k-1$ vertices such that after individualizing each element of S we can either find*

(a) a good canonical coloring of Ω, or

(b) a good canonical equipartition of Ω, or

(c) a good canonically embedded nontrivial regular graph.

Here canonicity is relative to the arbitrary choice of the sequence S.

Outcomes (a) and (b) allow for efficient Luks reduction. Case (c) requires further processing (Split-or-Johnson).

The proof of this lemma employs the d-dimensional Weisfeiler–Leman canonical refinement (d-WL), a natural way to canonically refine a coloring of Ω^d, with $d = O(k)$. The classical $d = 2$ case was introduced by Weisfeiler and Leman in 1968 [WeL, We]. The general case was defined by Rudi Mathon and this author [Ba79b] and independently by Immerman and Lander [ImL]. The celebrated Cai–Furer–Immerman paper [CaiFI] establishes that this very general combinatorial refinement method alone cannot discriminate between non-isomorphic pairs of graphs unless $d = \Omega(v)$ where v is the number of vertices. However, our application shows that d-WL becomes a powerful tool for $d = O(\log v)$ when combined with the group theory method. As far as I know, this paper is the first to derive analyzable gain from employing the d-dimensional WL method for unbounded values of d (or any value $d > 4$).

The proof of the Design Lemma rests, among other things, on Fisher's inequality that a block design (BIBD) has at least as many blocks as it has points.

2.4.6 Combinatorial partitioning II: Split-or-Johnson

THEOREM 10 (SPLIT-OR-JOHNSON). *Given a nontrivial regular graph $X = (V, E)$ with $n = |V|$ vertices, at quasipolynomial multiplicative cost we can find either*

(a) a good canonical coloring of V, or

(b) a good canonical equipartition of V, or

(c) a good canonically embedded nontrivial Johnson graph.

Here canonicity is relative to the arbitrary choices made that resulted in the multiplicative cost. The trivial Johnson graphs are the complete graphs $J(k, 1)$.

Outcomes (a) and (b) again allow for efficient Luks reduction. Outcome (c) provides even greater efficiency. Assume the canonically embedded Johnson graph is $J(m', t)$; so $m \geq \binom{m'}{t} \geq \binom{m'}{2}$ and therefore $m' < 1 + \sqrt{2m}$. We can therefore replace Γ by a set of size $m' = O(\sqrt{m})$, a dramatic reduction of the problem size.

We point out that item (c) is inevitable: Johnson graphs do not permit canonical partitioning of the type given in (a) and (b).

PROPOSITION 11 (RESILIENCE OF JOHNSON GRAPHS). *The multiplicative cost of a good canonical coloring or a good canonical equipartition of the Johnson graph $J(m, t)$ is $\geq (4t)^{m/(4t)}$.*

The proof rests on Theorem 8 which implies that if we do not invest such excessive multiplicative cost then our Johnson graph is simply reduced to a slighly smaller Johnson graph.

Note that $t = 2$ is an interesting case, largely responsible for the lack of progress over the $\exp(\widetilde{O}(\sqrt{v}))$ bound for a long time.

3. TECHNICAL PRELIMINARIES

After these semi-formal explanations, we turn to more rigorous treatment of some aspects of the algorithm.

3.1 Luks's Divide-and-Conquer

We briefly review Luks's framework. The goal is to find the set $\mathrm{Iso}_G(\mathfrak{x}, \mathfrak{y})$ of G-isomorphisms $\mathfrak{x} \to \mathfrak{y}$.

For the purposes of recursion, we make two generalizations. First, we also consider K-isomorphisms, where $K \subseteq G$ is not necessarily a group; and second, we observe only part of the input, "visible" through a "window" $\Delta \subseteq \Omega$.

Definition 12 (Window isomorphism). *Let $\Delta \subseteq \Omega$ and $K \subseteq \mathfrak{S}(\Omega)$. We set*

$$\mathrm{Iso}_K^{\Delta}(\mathfrak{x}, \mathfrak{y}) = \{\sigma \in K \mid (\forall u \in \Delta)(\mathfrak{x}(u) = \mathfrak{y}(u^{\sigma}))\}. \quad (2)$$

We apply this concept only to subcosets $K = G\tau$ ($G \leq \mathfrak{S}(\Omega)$ and $\tau \in \mathfrak{S}(\Omega)$) and G-invariant subsets Δ (i. e., $(\forall \sigma \in G)(\Delta^{\sigma} = \Delta)$). We note that in this case, $\mathrm{Iso}_{G\tau}^{\Delta}(\mathfrak{x}, \mathfrak{y})$ is either empty or a coset of the window-automorphism group $\mathrm{Aut}_G^{\Delta}(\mathfrak{x}) := \mathrm{Iso}_G^{\Delta}(\mathfrak{x}, \mathfrak{x})$ and will be represented by generators of this group plus one window-isomorphism.

The following *shift identity* is evident: if $K \subseteq \mathfrak{S}(\Omega)$ and $\Delta \subseteq \Omega$ then

$$\mathrm{Iso}_{K\tau}^{\Delta}(\mathfrak{x}, \mathfrak{y}) = \mathrm{Iso}_K^{\Delta}(\mathfrak{x}, \mathfrak{y}^{\tau^{-1}})\tau. \quad (3)$$

In particular, finding $G\tau$-isomorphisms is equivalent to finding G-isomorphisms. The extension to cosets is nevertheless helpful, as we shall see below ("descent").

We now state one of the principal recurrences in Luks's theory[2]; we call it the **"Chain Rule"**: window by window processing. Let $\Delta = \Delta_1 \cup \dots \cup \Delta_k$.

Procedure ChainRule

```
initialize: $L \leftarrow K$
for $i = 1$ to $k$
    $L \leftarrow \mathrm{Iso}_L^{\Delta_i}(\mathfrak{x}, \mathfrak{y})$
return $L$
```

It is clear that this procedure returns $\mathrm{Iso}_K^{\Delta}(\mathfrak{x}, \mathfrak{y})$. For the case of interest ($K = G\tau$ is a coset and the Δ_i are G-invariant) this provides for an extremely efficient recurrence: let $n_i = |\Delta_i|$; then we obtain the recurrence $f(n) \leq \sum f(n_i)$ for the cost of processing window size n (ignoring the overhead), which resolves to $f(n) = O(n)$ (assuming, as we always may, that the Δ_i are disjoint).

By the Chain Rule, we may always assume that G is transitive (so the window is $\Delta = \Omega$).

Luks's other recurrence, which we call **"descent,"** states that if $K = \bigcup_{i=1}^{\ell} K_i$ then $\mathrm{Iso}_K^{\Delta}(\mathfrak{x}, \mathfrak{y}) = \bigcup_{i=1}^{\ell} \mathrm{Iso}_{K_i}^{\Delta}(\mathfrak{x}, \mathfrak{y})$. This is used in the context that $K = G$ is a group, $H \leq G$ is a subgroup, and the K_i are the (right) cosets of H in G. It follows (in view of the shift identity (3)) that G-isomorphism reduces to $|G : H|$ instances of H-isomorphism (with respect to the same window).

When applying descent in our context, the multiplicative cost $|G : H|$ must be quasipolynomially bounded and must be offset by a significant reduction of the problem size.

3.2 Lifting

We are dealing with the Luks obstruction; so G is transitive and we have a giant representation $\varphi : G \to \mathfrak{S}(\Gamma)$. We want to find information about the φ-image of $\mathrm{Aut}_G(\mathfrak{x})$. Let $\overline{\sigma} \in \mathfrak{S}(\Gamma)$ and $H \leq G$. A *lifting* of $\overline{\sigma}$ to H is a permutation $\sigma \in G$ such that $\sigma^{\varphi} = \overline{\sigma}$. The set of liftings of $\overline{\sigma}$ to H is the set $\varphi^{-1}(\overline{\sigma}) \cap H$. This set is either empty or a subcoset of H.

Observation 13. *Let $\overline{\sigma} \in \mathfrak{S}(\Gamma)$; assume σ is a lifting of $\overline{\sigma}$ to G. Let further $K = \ker(\varphi)$. Then the set of liftings of $\overline{\sigma}$ to $\mathrm{Aut}_G(\mathfrak{x})$ is the set $\mathrm{Aut}_{K\sigma}(\mathfrak{x})$.*

Proof. The lifting of $\overline{\sigma}$ to G is $K\sigma$ so we need the set $K\sigma \cap \mathrm{Aut}_G(\mathfrak{x}) = \mathrm{Aut}_{K\sigma}(\mathfrak{x})$. □

We note that σ, if exists, can be found in polynomial time. So (by the shift equation) finding the coset of liftings of σ to $\mathrm{Aut}_G(\mathfrak{x})$ reduces to a single instance of K-isomorphism.

Now every K-orbit has size $\leq n/m$ where $m = |\Gamma|$.

Corollary 14. *One can decide, by efficient Luks reduction, whether or not $\mathrm{Aut}_G(\mathfrak{x})^{\varphi}$ is a giant in $\mathfrak{S}(\Gamma)$, and if it is, construct $\mathrm{Aut}_G(\mathfrak{x})$ and $\mathrm{Iso}_G(\mathfrak{x}, \mathfrak{y})$, invoking $O(m)$ instances of K-isomorphism, where $K = \ker(\varphi)$. The cost of this computation can be estimated by the recurrence $f(n) \leq n^4 f(n/m)$ which resolves to $f(n) = n^{O(\log n)}$.*

Proof. The set of 3-cycles in Γ generates $\mathfrak{A}(\Gamma)$, and together with a transposition they generate $\mathfrak{S}(\Gamma)$. So $(\mathrm{Aut}_G(\mathfrak{x})^{\varphi}$ is a giant iff all 3-cycles in $\mathfrak{S}(\Gamma)$ lift to $\mathrm{Aut}_G(\mathfrak{x})$; if they do, these liftings together with the lifting of a transposition (which may be empty) generate $\mathrm{Aut}_G(\mathfrak{x})$. Moreover, if $\mathrm{Aut}_G(\mathfrak{x})^{\varphi}$ is a giant then $\mathfrak{x} \cong_G \mathfrak{y}$ iff $\mathfrak{x} \cong_K \mathfrak{y}$ or $\mathfrak{x} \cong_{K\tau} \mathfrak{y}$ where τ is a lifting of a transposition in $\mathfrak{S}(\Gamma)$ to G.

As to the complexity, first we note that $K \triangleleft G$ and all orbits of a normal subgroup of a transitive group have equal length. Let k be the length of the orbits of K. Then $k \leq n/m$ by the Affected Orbit Lemma (part (b) of Thm. 6) (in fact, $k \leq n/2$ would suffice for us). Therefore, combining the procedure described with the Chain Rule applied to the orbits of K we obtain the recurrence $f(n) \leq m^3(n/k)f(k) \leq< n^4 f(n/2)$ which resolves to $f(n) = n^{O(\log n)}$. (Here m^3 is an upper bound on the number of 3-cycles to be lifted. In fact, lifting a connected set of 3-cycles would suffice.) □

So we are happy if all 3-cycles in $\mathfrak{S}(\Gamma)$ lift to $\mathrm{Aut}_G(\mathfrak{x})$. In fact, we are happy if many of them lift: let $H \leq \mathfrak{S}(\Gamma)$ be the group generated by those 3-cycles that lift and let ℓ be the length of the largest orbit of H. If $\ell > m/2$, we already get an efficient reduction, by splitting Γ, dealing with the smaller part by recursion and dealing with the large orbit as above.

So the only case we need to worry about is when $\ell \leq m/2$. In fact, by recursion we may now assume none of the 3-cycles lifts. In this case the non-fullness certificates will aggregate to a canonical k-ary relation on Γ, as indicated in the last two paragraphs of Section 2.4.4.

4. LOCAL CERTIFICATES

In this section we describe the construction and aggregation of local certificates.

4.1 Local Certificates Algorithm

In this subsection we give a detailed description of the construction of local certificates, the core algorithm of the paper, which was informally explained in Sec. 2.4.3.

The situation we consider is as follows.

The input is a transitive permutation group $G \leq \mathfrak{S}(\Omega)$, a giant representation $\varphi : G \to \mathfrak{A}(\Gamma)$, and two strings $\mathfrak{x}, \mathfrak{y} : \Omega \to \Sigma$.

[2] We note that the idea of the Chain Rule already appears in a key role in [Ba79a]

Notation: $n = |\Omega|$, $m = |\Gamma|$. We fix a quantity $k > 2+\log_2 n$ (but not much greater) and assume $m \geq 10k$. Subsets $T \subset \Gamma$ of size $|T| = k$ will be referred to as "test sets." The letter T will be used for test sets throughout this section.

If $L \leq G$ then L also acts on Γ via φ so for a test set T we can speak of the setwise stabilizer of T in L; we write L_T for this subgroup.

We say that T is L-invariant if $L_T = L$. We write $\psi_T : G_T \to \mathfrak{S}(T)$ for the map that restricts the the domain of φ to G_T and the range to $\mathfrak{S}(T)$. The group G_T can be computed in polynomial time as $G_T = \varphi^{-1}(\mathfrak{S}(\Gamma)_T)$.

We note that $\psi_T : G_T \to \mathfrak{S}(T)$ is an epimorphism (since $|T| \leq m-2$).

Definition 15 (Full set). *Let T be a test set. We say that T is* full *with respect to the input string $\mathfrak{x}$ if $\mathrm{Aut}_G(\mathfrak{x})_T^{\psi_T} \geq \mathfrak{A}(T)$, i. e., the G-automorphisms of $\mathfrak{x}$ induce a giant on T.*

We consider the problem of deciding whether or not a given test set is full and compute useful certificates of either outcome. We show that this question can efficiently (in time $k!\,\mathrm{poly}(n)$) be reduced to the String Isomorphism problem on inputs of size $\leq n/k$.

Certificate of non-fullness. We certify non-fullness of the test set T by computing a permutation group $M(T) \leq \mathfrak{S}(T)$ such that (i) $M(T) \ngeq \mathfrak{A}(T)$ and (ii) $M(T) \geq \mathrm{Aut}_G(\mathfrak{x})_T^{\psi_T}$ ($M(T)$ is guaranteed to contain the projection of the G-automorphism group of $\mathfrak{x}$).

Such an "encasing group" $M(T)$ can be thought of as a constructive refutation of fullness.

Certificate of fullness. We certify fullness of the test set T by computing a permutation group $K(T) \leq \mathfrak{S}(\Omega)$ such that (i) $K(T) \leq \mathrm{Aut}_G(\mathfrak{x})$ and (ii) T is $K(T)$-invariant and $K(T)^{\psi_T} \geq \mathfrak{A}(T)$.

Note that $K(T)$ represents an easily (poly-time) verifiable proof of fullness of T.

Our ability to find $K(T)$, the certificate of fullness, may be surprising because it means that from a local start (that may take only a small segment of $\mathfrak{x}$ into account), we have to build up global automorphisms (automorphisms of the full string $\mathfrak{x}$). Our ability to do so critically depends on the "Unaffected Stabilizers Lemma" (Thm. 6).

Theorem 16 (Local certificates). *Let $T \subseteq \Gamma$ where $|T| = k$ be a test set. Assume $\max\{8, 2+\log_2 n\} < k \leq m/10$. By making $\leq k!n^2$ calls to String Isomorphism problems on domains of size $\leq n/k$ and performing $k!\,\mathrm{poly}(n)$ computation we can decide whether or not T is full and*

(a) if T is full, find a certificate $K(T) \leq \mathrm{Aut}_G(\mathfrak{x})$ of fullness

(b) if T is not full, find a certificate $M(T) \leq \mathfrak{S}(T)$ of non-fullness.

Definition 17 (Affected). *Let $G \leq \mathfrak{S}(\Omega)$ be a permutation group and and $\varphi : G \to \mathfrak{S}(\Gamma)$ a homomorphism. Consistently with previous usage, for a subgroup $H \leq G$ we say that $x \in \Omega$ is* affected by *(H, φ) if $H_x^{\varphi} \ngeq \mathfrak{A}(\Gamma)$. Let $\mathrm{Aff}(H, \varphi)$ denote the set of elements affected by (H, φ), i. e.,*

$$\mathrm{Aff}(H,\varphi) = \{x \in \Omega \mid H_x^{\varphi} \ngeq \mathfrak{A}(\Gamma)\}. \qquad (4)$$

If $x \in \Omega$ is affected by (H, φ) then all elements of the orbit x^H are affected by (H, φ). In other words, $\mathrm{Aff}(H, \varphi)$ is an H-invariant set. So we can speak of *affected orbits* of H (of which all elements are affected).

We observe the dual monotonicity of the Aff operator:

$$\text{If } H_1 \leq H_2 \leq G \text{ then } \mathrm{Aff}(H_1, \varphi) \supseteq \mathrm{Aff}(H_2, \varphi). \qquad (5)$$

The algorithm will consider the input in an increasing sequence of "windows" $W \subseteq \Omega$; in each round, the part of the input outside the window will be ignored. The group $A(W)$ will be the subgroup of G_T that respects the string $\mathfrak{x}^W$, the restriction of $\mathfrak{x}$ to W.

The initial window is the empty set (the input is wholly ignored), so the initial group is G_T. Then in each round we add to W the set of elements of Ω affected by the current group $A(W)$. I like to visualize this process as "growing the beard" (W being the beard). By the second round $W \neq \emptyset$ because $\mathrm{Aff}(G_T, \psi_T)$ cannot be empty (by the Unaffected Stabilizer Theorem).

As an increasing segment of $\mathfrak{x}$ is taken into account, the group $A(W)$ (the automorphism group of this segment) decreases, and thereby the set of elements affected by $A(W)$ increases. (Previous windows will always be invariant under $A(W)$.)

We stop when one of two things happens: either ψ_T restricted to $A(W)$ is no longer a giant homomorphism, or the beard stops growing: no element outside W is affected by $A(W)$.

In the former case we declare that our test set T is *not full* (witnessed by a non-giant group $M(T) := A(W)^{\psi_T} \leq \mathfrak{S}(T)$). Note that the reason $M(T)$ is not a giant is still "local," it only depends on the restriction of $\mathfrak{x}$ to the current window.

In the latter case we declare that T is *full*, and bring as witness the group $K(T) = A(W)_{(U)}$, the pointwise stabilizer of $U := \Omega \setminus W$ in $A(W)$. We claim two things about $K(T)$. First, $K(T)^{\psi_T} \geq \mathfrak{A}(\Gamma)$. This follows from the Unaffected Stabilizers Lemma (part (a) of Thm. 6) since none of the elements of U is affected. (This is why the beard stopped growing.) Second, we observe that $K(T) \leq \mathrm{Aut}_G(\mathfrak{x})$. Indeed, $K(T)$ respects the letters of the string $\mathfrak{x}$ on W (this is an invariant of the algorithm); and it fixes all elements outside W, so the letters of the string restricted to U are automatically respected.

Here is the algorithm in pseudocode.

Proof of Theorem 16. For $W \subseteq \Omega$ let

$$A(W) = \mathrm{Aut}_{G_T}^{W}(\mathfrak{x}). \qquad (6)$$

All sets denoted T, T', and T_i below will be subsets of Γ of size k (the "test sets"). An invariant of the **while** loop will be that T is invariant under the action of the group $A(W)$, i. e., $A(W) \leq G_T$.

```
Procedure LocalCertificates

Input: $G \leq \mathfrak{S}(\Omega)$, epimorphism $\psi_T : G_T \to \mathfrak{S}(\Gamma)$,
        test set $T \in \binom{\Gamma}{k}$
Output: decision "T full/not full," group $K(T)$ (if full) or
        $M(T)$ (if not full), set $W(T) \subseteq \Omega$
Note: The group $A(W)$ defined in Eq. (6) is updated when
        $W$ is updated
```

```
01   $W := \emptyset$                                    (: so $A(W) = G_T$ :)
02   while $A(W)^{\psi_T} \geq \mathfrak{A}(T)$ and $\text{Aff}(A(W), \psi_T) \not\subseteq W$
03      $W \leftarrow \text{Aff}(A(W), \psi_T)$        (: growing the beard :)
04      recompute $A(W)$
05   end(while)
06   $W(T) \leftarrow W$
07   if $A(W)^{\psi_T} \geq \mathfrak{A}(T)$        (: so $\text{Aff}(A(W), \psi_T) \subseteq W$:)
08      then $K(T) \leftarrow A(W)_{(U)}$ where $U = \Omega \setminus W$
09      return $W(T)$, $K(T)$, "$T$ full," exit
                               (: certificate of fullness found :)
10   else $M(T) \leftarrow A(W)^{\psi_T}$
11      return $W(T)$, $M(T)$, "$T$ not full," exit
                           (: certificate of non-fullness found :)
```

We need to show how to recompute $A(W)$ on line 4. We write W_{old} for the value of W before the execution of line 03 and W_{new} after.

Procedure Recompute $A(W)$

```
04a  $N \leftarrow A(W_{\text{old}})^{\psi_T}_{(T)}$  (: kernel of $A(W_{\text{old}}) \to \mathfrak{S}(T)$ map :)
04b  $L \leftarrow \emptyset$            (: $L$ will collect elements of $A(W_{\text{new}})$ :)
04c  for $\overline{\sigma} \in A(W_{\text{old}})^{\psi_T}$   (: $A(W_{\text{old}})^{\psi_T} = \mathfrak{A}(T)$ or $\mathfrak{S}(T)$ :)
04d     select $\sigma \in A(W_{\text{old}})$ such that $\sigma^{\psi_T} = \overline{\sigma}$
                                          (: lifting $\overline{\sigma}$ to $A(W_{\text{old}})$ :)
04e     $L(\overline{\sigma}) \leftarrow \text{Aut}^{W_{\text{new}}}_{N\sigma}(\mathfrak{x})$
                       (: performing descent to $N$ :)
04f     $L \leftarrow L \cup L(\overline{\sigma})$
04g  end(for)
04h  return $A(W_{\text{new}}) \leftarrow L$
```

Justification. First we observe that on each iteration of the **while** loop on lines 02–05, $A(W_{\text{new}}) \leq A(W_{\text{old}})$ and $W_{\text{new}} \supseteq W_{\text{old}}$. In fact, these inclusions are proper or else we exit on line 02. In particular, T is invariant under $A(W)$ throughout the process because it is invariant in line 01. It also follows that on line 07 we actually have $\text{Aff}(A(W), \psi_T) = W$. We also note that the **while** loop will be executed at least once (by the comment on line 01).

Claim 18. *On line 08, $K(T)^{\psi_T} \geq \mathfrak{A}(T)$ and $K(T) \leq \text{Aut}_G(\mathfrak{x})$. In particular, T is full.*

Proof. $K(T) \geq \mathfrak{A}(T)$ is the crucial consequence of the Unaffected Stabilizers lemma (part (a) of Thm 6), applied to the giant representation $\overline{\psi}_T : A(W_{\text{old}}) \to \mathfrak{S}(T)$. ($\overline{\psi}_T$ denotes the restriction of ψ_T to $A(W_{\text{old}})$.)

To show that $K(T) \leq \text{Aut}_G(\mathfrak{x})$ let $\sigma \in K(T)$ and $u \in \Omega$. We need to show that $\mathfrak{x}(u^\sigma) = \mathfrak{x}(u)$. If $u \in W$ then this follows because $\sigma \in A(W) = \text{Aut}^W_G(\mathfrak{x})$. If $u \in U$ then $u^\sigma = u$. □

Claim 19. *If T is not full then we reach line 10 with $M(T) \not\geq \mathfrak{A}(T)$ and $\text{Aut}_G(\mathfrak{x})^{\psi_T}_T \leq M(T)$.*

Proof. We reach line 10 by Claim 18. We then have $\text{Aut}_G(\mathfrak{x})^{\psi_T}_T \leq M(T)$ because the relation $\text{Aut}_G(\mathfrak{x})^{\psi_T}_T \leq A(W)$ is an invariant of the process. □

Next we justify procedure Recompute $A(W)$. Correctness of the output is immediate from the fact that $W_{\text{new}} \supseteq W_{\text{old}}$ and the observation

$$A(W_{\text{old}}) = \bigcup_{\overline{\sigma}} N\sigma \tag{7}$$

where the union extends over $\overline{\sigma} \in A(W_{\text{old}})^{\psi_T}$.

Finally we need to justify the complexity assertion. This is where the Affected Orbit Lemma (part (b) of Thm. 6) plays a critical role.

The **while** loop is executed at most n times (because W strictly increases in each round; we exit on line 02 when the "beard" stops growing), so the dominant component of the complexity is in recomputing $A(W)$. We have reduced this to $\leq k!$ instances of string N-isomorphism on the window W_{new}.

By the Affected Orbit Lemma, each orbit of N in W_{new} has length $\leq n/k$.

We conclude that descent to N and the application of the Chain Rule to N reduces the recomputation of $A(W)$ to $\leq n \cdot k!$ instances of String Isomorphism on windows of size $\leq n/k$, justifying the stated complexity estimate.

Our procedure does more than stated in Theorem 16. It also returns the set $W(T)$. We summarize key properties of this assignment.

Proposition 20. *As in Theorem 16, let a "test set" be a subset $T \subseteq \Gamma$ with $|T| = k$ elements where $\max\{8, 2 + \log_2 n\} < k \leq m/10$. For all test sets T we have*

(i) $\Omega(T) \subseteq W(T) \subseteq \Omega$

(ii) $W(T)$ *is invariant under* $\text{Aut}_{G_T}(\mathfrak{x})$

(iii) *if* T *is full then* $W(T) = \text{Aff}(\text{Aut}^{W(T)}_{G_T}(\mathfrak{x}))$

(iv) *if* T *is full then* $K(T)^{\psi_T}$ *fixes all elements of* $\Omega \setminus W(T)$

(v) *the assignment* $T \mapsto W(T)$ *is canonical.*

We need to highlight one more fact about the structures we obtained.

Notation 21 (Truncation of strings). *Let $*$ be a special symbol not in the alphabet Σ. For the string $\mathfrak{x} : \Omega \to \Sigma$ and "window" $W \subseteq \Omega$ we define the string $\mathfrak{x}^W : \Omega \to (\Sigma \cup \{*\})$ by setting $\mathfrak{x}^W(u) = \mathfrak{x}(u)$ for $u \in W$ and $\mathfrak{x}^W(u) = *$ for $u \in \Omega \setminus W$.*

Notation 22 (Coloring of strings). *For the string $\mathfrak{x} : \Omega \to \Sigma$ and the "test set" $T \subseteq \Gamma$ we define the string $\mathfrak{x}_T : \Omega \to (\Sigma \times \{0,1\})$ by setting $\mathfrak{x}_T(u) = (\mathfrak{x}(u), 1)$ if $u \in W(T)$ and $\mathfrak{x}_T(u) = (\mathfrak{x}(u), 0)$ if $u \notin W(T)$.*

Proposition 23 (Comparing local certificates). *For all test sets $T, T' \subseteq \Gamma$ with $|T| = |T'| = k$ and all strings $\mathfrak{x}, \mathfrak{x}' : \Omega \to \Sigma$ we can compute $\text{Iso}_G\left(\mathfrak{x}^{W(T)}_T, (\mathfrak{x}')^{W(T')}_{T'}\right)$ by making $\leq k!n^2$ calls to String Isomorphism problems on domains of size $\leq n/k$ and performing $k!\,\text{poly}(n)$ computation.*

Proof. Run procedure LocalCertificates simultaneously on $(\mathfrak{x}, T)$ and on $(\mathfrak{x}', T')$, maintaining the variable W for (x, T) and the variable W' for $(\mathfrak{x}', T')$. Further maintain the set $Q = \text{Iso}_G(\mathfrak{x}^W_T, (\mathfrak{x}')^{W'}_{T'})$. On line 01 we shall have $Q = G_T\sigma$ for any $\sigma \in G$ that takes T to T'.

Change line 04 to "recompute $A(W)$ and Q."
Here is the modified "Recompute" code.

Procedure Recompute $A(W)$ and Q

```
04a    $N \leftarrow A(W_{\text{old}})^{\psi_T}_{(T)}$    (: kernel of $A(W_{\text{old}}) \to \mathfrak{S}(T)$ :)
04b1   $L \leftarrow \emptyset$        (: $L$ will collect elements of $A(W_{\text{new}})$ :)
04b2   $R \leftarrow \emptyset$          (: $R$ will collect elements of $Q_{\text{new}}$ :)
04c0   fix $\pi_0 \in Q_{\text{old}}$
04c1   for $\overline{\sigma} \in A(W_{\text{old}})^{\psi_T}$
                        (: $A(W_{\text{old}})^{\psi_T} = \mathfrak{A}(T)$ or $\mathfrak{S}(T)$ :)
04d1       select $\sigma \in A(W_{\text{old}})$ such that $\sigma^{\psi_T} = \overline{\sigma}$
                          (: lifting $\overline{\sigma}$ to $A(W_{\text{old}})$ :)
04d2       $\pi \leftarrow \sigma\pi_0$         (: $\pi \in Q_{\text{old}}$ :)
04e1       $L(\overline{\sigma}) \leftarrow \text{Aut}^{W_{\text{new}}}_{N\sigma}(\mathfrak{x})$
04e2       $R(\overline{\sigma}) \leftarrow \text{Iso}_{N\pi}(\mathfrak{x}^{W_{\text{new}}}_{T}, (\mathfrak{x}')^{W'_{\text{new}}}_{T'})$
                          (: performing descent to $N$ :)
04f1       $L \leftarrow L \cup L(\overline{\sigma})$  (: collecting automorphisms :)
04f2       $R \leftarrow R \cup R(\overline{\pi})$  (: collecting isomorphisms :)
04g    end(for)
04x    if $R = \emptyset$ then reject isomorphism, exit
04h    else return $A(W_{\text{new}}) \leftarrow L$ and $Q \leftarrow R$
```

The analysis is analogous with the analysis of the Recompute $A(W)$ routine. □

Prop. 23 will allow us to combine the non-fullness certificates into a canonical k-ary relation.

4.2 Aggregating the Local Certificates

We continue the notation of the previous section.

Theorem 24 (AggregateCertificates). *Let $\varphi : G \to \mathfrak{S}(\Gamma)$ be a giant representation, where $G \leq \mathfrak{S}(\Omega)$, $|\Omega| = n$, and $|\Gamma| = m$. Let $\max\{8, 2 + \log_2 n\} < k < m/10$. Then, at a multiplicative cost of $m^{O(k)}$, we can either find*

(a) a good canonical coloring of Γ, or

(b) a good canonical equipartition of Γ, or

(c) a good canonically embedded k-ary relational structure with relative symmetry defect $\geq 1/2$,

or reduce the determination of $\text{Iso}_G(\mathfrak{x}, \mathfrak{y})$ to $n^{O(1)}$ instances of size $\leq 2n/3$.

We preface the proof with a fact about permutation groups.

Definition 25. *We say that a group $G \leq \mathfrak{S}(\Omega)$ is t-transitive if its induced action on the set of $n(n-1)\cdots(n-t+1)$ ordered t-tuples of distinct elements is transitive (where $n = |\Omega|$). The* degree of transitivity *$d(G)$ is the largest t such that G is t-transitive.*

We have $d(\mathfrak{S}_n) = n$ and $d(\mathfrak{A}_n) = n - 2$. For all other permutation groups, $d(G) \leq 5$ and in fact if $n \geq 25$ then $d(G) \leq 3$.

One more fact we shall need.

Proposition 26. *The symmetry defect of a nontrivial regular graph is $\geq 1/2$.*

Proof of Thm. 24. We describe the procedure, interspersed with the justification.

Run the LocalCertificates routine for both inputs $\mathfrak{x}, \mathfrak{y}$ and all test sets $A \in \binom{\Gamma}{k}$.

Run the CompareLocalCertificates routine for all pairs $((\mathfrak{x}, T), (\mathfrak{x}', T'))$ where $\mathfrak{x}$ is fixed, $\mathfrak{x}' \in \{\mathfrak{x}, \mathfrak{y}\}$, and $T, T' \in \binom{\Gamma}{k}$ are test sets (a total of $2\binom{m}{k}^2$ runs).

Let $F(\mathfrak{x})$ be the subgroup generated by the groups $K(T)$ for all full subsets $T \in \binom{\Gamma}{k}$ with reference to input string $\mathfrak{x}$. So $F(\mathfrak{x})$, and with it $F(\mathfrak{x})^{\varphi}$, are canonically associated with $\mathfrak{x}$. In particular, if $F(\mathfrak{y})$ is analogously defined for $\mathfrak{y}$, then $F(\mathfrak{x})^{\varphi}$ is permutationally isomorphic to $F(\mathfrak{y})^{\varphi}$, i. e., there exists a permutation $\alpha \in \mathfrak{S}(\Gamma)$ such that $F(\mathfrak{y})^{\varphi} = \alpha^{-1}F(\mathfrak{x})^{\varphi}\alpha$.

Below we ignore $\mathfrak{y}$ and focus on $\mathfrak{x}$, omitting it from the notation, so we write $F = F(\mathfrak{x})$. But our guide is the above consequence of canonicity.

(1) **if** the support of F^{φ} (set of elements in Γ not fixed by F^{φ}) has size $\geq m/4$ and no orbit of F^{φ} has length $> 3m/4$, color the elements according to the length of their F^{φ}-orbit. Now either no color class has size $> 3m/4$ (item (a) accomplished) or a large color class is nontrivially equipartitioned into orbits (item (b) accomplished), **exit**

(2) **else if** F^{φ} has an orbit $C \subseteq \Gamma$ of length $|C| > 3m/4$
(: since $|C| > m/2$, this orbit is canonical :)
let F^C denote the restriction of F^{φ} to C

2a **if** $F^C \geq \mathfrak{A}(C)$ then we are doen by the comment after the proof of Cor. 14, **exit**

2b **else** let d be the degree of transitivity of F^C
(: so $1 \leq d \leq 5$:)
individualize the elements of a set $S \in \binom{C}{d-1}$
(: so $F^C_{(S)}$ is transitive but not doubly transitive on $C' := C \setminus S$:)
Let the binary relational structure
$\mathfrak{X} = (C'; R_1, \ldots, R_r)$ be the "orbital configuration" of $F^C_{(S)}$ on C', meaning that the R_i are the orbits of $F^C_{(S)}$ on $C' \times C'$. Note that $r \geq 3$ because $F^C_{(S)}$ is not doubly transitive on C'. (: Warning: the numbering of the R_i is not canonical; isomorphisms may permute the R_i :)
Let $R_1 = \text{diag}(C')$ be the diagonal
(: so for $i \geq 2$ the constituents $X_i = (C', R_i)$ are nontrivial biregular digraphs :)
Individualize one of the X_i ($i \geq 2$) (: multiplicative cost $r - 1 \leq m - 1$:)
if X_i is undirected ($R_i = R_I^{-1}$), **return** X_i, **exit**
else if the out-degree of the vertices of X_i is less than $(m-1)/2$ then **return** the graph $(C', R_i \cup R_i^{-1})$ (: again a nontrivial regular graph :)
else individualize a vertex $x \in C'$; this will split the remaining vertices of C' into two equal halves, namely, the in-neighbors and the out-neighbors; color them accordingly (item (a) accomplished), **exit**

(3) **else** $|D| \geq 3m/4$ where $D \subseteq \Gamma$ is the set of fixed points of F^{φ}. So in this case, if $T \subset D$ then T is not full. (In fact even if $T \cap D \neq \emptyset$ then T is not full.)
Claim (Turning local asymmetry into global irregularity)
In time $m^{O(k)}$ we can construct a canonical k-ary relational structure on D with symmetry defect (much) greater than $1/2$.

Proof. For a set S with $|S| = s$ elements, let $S^{\langle k \rangle}$

denote the set of $s(s-1)\dots(s-k+1)$ of ordered k-tuples of distinct elements of S.

For notational simplicity let us denote our input strings by $\mathfrak{x}_1$ and $\mathfrak{x}_2$ (rather than $\mathfrak{x}$ and $\mathfrak{y}$). Let D_i be the subset $D \subseteq \Gamma$ derived from input $\mathfrak{x}_i$. Let $D_i' = D_i \times \{i\}$ (to make the sets disjoint).

Consider the following category $\mathcal{L}$. The objects of the category correspond to the pairs (T, i) where $T \in \binom{D_i}{k}$ is a test set. The set of morphisms $(T, i) \to (T', j)$ are the bijections $T \to T'$ corresponding to the set $\mathrm{Iso}_G\left((\mathfrak{x}_i)_T^{W_i(T)}, (\mathfrak{x}_j)_{T'}^{W_j(T')}\right)$ for all $T, T' \in \binom{\Gamma}{k}$, where W_i corresponds to W under input $\mathfrak{x}_i$. These sets have been computed in Prop. 23.

The morphisms define an equivalence relation on $(D_1')^{\langle k\rangle} \cup (D_2')^{\langle k\rangle}$ (because $\mathcal{L}$ is a category (closed under composition of morphisms)). Let $R_1, \dots, R_r$ denote the equivalence classes and let $R_j(i) = R_j \cap (D_i')^{\langle k\rangle}$. Then $\mathfrak{X}_i = (D_i'; R_1(i), \dots, R_r(i))$ is a canonical k-ary relational structure on D_i' since any G-isomorphism $\mathfrak{x}_{i_1} \to \mathfrak{x}_{i_2}$ induces a a morphisms in $\mathcal{L}$ on any test set ($i_1, i_2 \in \{1, 2\}$).

The symmetry defect of $\mathfrak{X}_i$ is at least $|D_i| - k + 1 > m/2$. Indeed, otherwise $\mathrm{Aut}(\mathfrak{X}_i)$ could act as a giant on a test set in D_i, contrary our assumption. □

Now **return** this canonical k-ary relational structure, **exit**

This completes the procedure and the proof.

5. CONCLUDING REMARKS

5.1 Dependence on the Classification of Finite Simple Groups

The analysis of the algorithm, as stated, depends on the Classification of Finite Simple Groups (CFSG) at three points:

(a) via Cameron's classification of large primitive permutation groups;

(b) in the bound on the degree of transitivity of permutation groups;

(c) in the proof of the Unaffected Stabilizers lemma.

The role of item (a) was in identifying the obstructions to the efficiency of Luks's method. This was already recognized (by Gene Luks and this author) in 1980. The dependence of this characterization on CFSG has ever since been believed to be inevitable since there seems to be no hope to prove Cameron's result without a detailed use of the CFSG.

Yet in the present algorithm we can replace the application of Cameron's result with an application of the Split-or-Johnson routine combined with an elementary bound on the degree of transitivity (see item (b)), and the modification does not even make the algorithm more complicated. So this dependence on CFSG has been eliminated.

Item (b) was used in the Aggregation of fullness-certificates. If d is the degree of transitivity in question then the algorithms individualizes $d-1$ points at a multiplicative cost of n^{d-1}. By CFSG, we have $d \le 5$. However, by an elementary 1934 result by Wielandt [Wi34], we have $d \le 3 \ln n$, and we can afford individualizing $O(\log n)$ points. (An even more elementary $O(\log^2 n / \log\log n)$ bound was given by Bochert in the 19th century [Bo92]; see a concise proof in [BaS]. Even this weaker bound would suffice for us.) (We may also note that if in a run of the algorithm we would ever encounter a case with $d > 5$, the algorithm would yield an explicit permutation representation of a currently unknown finite simple group.)

Item (c), specifically the proof of Lemma 7, depends on the CFSG through Schreier's Hypothesis that states that the outer automorphism group of every finite simple group is solvable. László Pyber recently announced an elementary proof of Lemma 7 under the stronger assumption $k > (\log n)^c$ for some costant c (instead of $k > 2 + \log_2 n$). This suffices for our algorithmic result with a slightly increased but still quasipolynomial running time.

Pyber's result then entirely eliminates the dependence of the analysis of the algorithm on CFSG.

5.2 How Easy is Graph Isomorphism?

The first theoretical evidence against the possibility of NP-completeness of GI was the equivalence of existence and counting [Ba77, Mat], not observed in any NP-complete problem. The second, stronger evidence came from the early theory of interactive proofs: GI belongs to the complexity class coAM, and therefore if GI is NP-complete then the polynomial-time hierarchy collapses to the second level (Goldreich–Micali–Wigderson 1987 [GoMW]). Our result provides a third piece of evidence: GI is not NP-complete unless all of NP can be solved in quasipolynomial time.

A number of questions remain. The first one is of course whether GI is in P. Such expectations should be tempered by the status of the *Group Isomorphism* problem[3]: given two groups by their Cayley tables, are they isomorphic? It is easy to reduce this problem to GI. In fact, Group Isomorphism seems much easier than GI; it can trivially be solved in time $n^{O(\log n)}$ where n is the order of the group. But in spite of considerable effort and the availability of powerful algebraic machinery, Group Isomorphism is still not known to be in P. We are not even able to decide Group Isomorphism[4] in time $n^{o(\log n)}$.

[3] In complexity theory, the "Group Isomorphism Problem" refers to groups given by Cayley tables; in other words, complexity is compared to the order of the group. From the point of view of applications, this complexity measure is of little use; in computational group theory, groups are usually given in compact representations (permutation groups, matrix groups given by lists of generators, p-groups given by power commutator presentation, etc.). But the fact remains that even in the unreasonably redundant representation by Cayley tables, we are unable to solve the problem is polynomial time.

[4] A simple algorithm, proposed by Tim Gowers on Dick Lipton's blog in November 2011, has a chance of running in $n^{O(\sqrt{\log n})}$. Let the *k-profile* of a finite group G be the function f on isomorphism types of k-generated groups where $f(H)$ counts those k-tuples of elements of G that generate a subgroup isomorphic to H. For what k do k-profiles discriminate between nonisomorphic groups of order n? It is known that $k < (1/2)\sqrt{\log_2 n}$ is insufficient for infinitely many values of n (Glauberman, Grabowski [GlG]). Whether some k that is not much greater than $\sqrt{\log n}$ suffices is an open question that I think would deserve attention. The test case is p-groups of class 2; the Glauberman–Grabowski examples belong to this class.

A closely related challenge that deserves attention is the String Isomorphism problem on $n = p^k$ points, with respect to the linear group $\mathrm{GL}(k,p)$. The order of this group is about $p^{k^2} = n^{\log_p n}$; the question is, can this problem be solved in time $p^{o(k^2)}$ (or perhaps even in $\mathrm{poly}(n)$ time). I note that this problem can be encoded as a GI problem for graphs with $\mathrm{poly}(n)$ vertices so if GI $\in$ P then this problem is in P as well.

The result of the present paper amplifies the significance of the Group Isomorphism problem (and the challenge problem stated) as a barrier to placing GI in P. It is quite possible that the intermediate status of GI (neither NP-complete, nor polynomial time) will persist.

In fact, even putting GI in coNP faces the same obstacle: Group Isomorphism is not known to be in coNP.

5.3 How Hard is Graph Isomorphism?

Paradoxically, from a structural complexity point of view, GI (still) seems harder than factoring integers. The decision version of Factoring (given positive integers x, y, does x have divisor d in the interval $2 \le d \le y$?) is in NP $\cap$ coNP while the best we can say about GI is NP $\cap$ coAM. Factoring can be solved in polynomial time on a quantum computer, but no quantum advantage has yet been found for GI. On the other hand, apparently hard instances of factoring abound, whereas we don't know how to construct hard instances of GI. Could this be an indication that in structural complexity maybe we are not asking the right questions?

Even more baffling is another complexity arena, where GI is provably hard, on par with many NP-hard problems: relaxation hierarchies in proof complexity theory (Sherali–Adams, Sum-of-Squares hierarchies). Building on the seminal paper by Cai, Furer, and Immerman [CaiFI], increasingly powerful hierarchies have recently been shown to be unable to refute isomorphism of graphs on sublinear levels [AtM, OWWZ, SnSC], showing that GI tests based on these hierarchies necessarily have exponential (even factorial) complexity. However, hard-to-distinguish CFI pairs of graphs and the related pairs of which isomorphism is hard to refute in these hierarchies are vertex-colored graphs with bounded color classes. Testing isomorphism of such pairs of graphs was shown to be in polynomial time via the first application of group theory (1979/80) that used hardly more than Lagrange's Theorem from group theory [Ba79a, FuHL]. One lesson is that these hierarchies have difficulty capturing the power of even the most naive applications of group theory. Given that hardness with respect to these hierarchies can now be proved by reduction from GI, this raises the question, in what sense these hierarchies indicate hardness.

Another natural question in the direction of stuctural hardness of Graph Isomorphism was raised by Jacobo Torán: Can one prove that GI is P-hard (under Logspace reductions)? (See the survey [ArT].)

Torán has shown that GI is hard under AC^0-reductions for the complexity class DET of problems NC^1-reducible to the determinant [To].

5.4 Outlook

On the bright side, a number of GI-related questions may look a bit more hopeful now.

We state three questions that this paper does not address but to which a positive answer now seems plausible.

1. Do graphs admit quasipolynomial-time computable *canonical forms?*

2. Can one test graph isomorphism in quasipolynomial time and *polynomial space?*

3. Can one test isomorphism of *hypergraphs* in time, quasipolynomial in the number vertices and polynomial in the number of edges?

The second problem was suggested by Gene Luks. The third problem looks the most challenging. More challenging yet might be the next set of problems.

While GI is complete over the isomorphism problems of *explicit structures*, there are interesting classes of non-explicit structures where progress may be possible. Two important examples are *equivalence of linear codes* and *conjugacy (permutational equivalence) of permutation groups.* The former easily reduces to the latter. Both of these problems belong[5] to NP $\cap$ coAM and therefore they are not NP-complete unless the polynomial-time hierarchy collapses. In spite of this complexity status, no moderately exponential ($\exp(n^{1-c})$) algorithm is known for either problem. GI reduces to each of these problems [Lu93][6]. Regarding both problems, see also [BaCGQ, BaCQ].

It would be of great interest to find stronger structural results that would better correspond to the "local → global symmetry" philosophy. This raises difficult mathematical questions that our algorithmic divide-and-conquer techniques bypass, but results of this flavor could make the algorithm more elegant and more efficient.

Finally two more concrete questions.

The first is about coherent configurations (see [Ba81, SuW, Ba15] for the definition and basic properties). Let $\mathfrak{X} = (V; \mathcal{R})$ be a homogeneous coherent configuration with n vertices. (Homogeneity means all vertices have the same color.) Let $W \subseteq V$, $|W| \ge \alpha n$. Suppose that the induced configuration $\mathfrak{X}[W]$ is a Johnson scheme. Is there a constant $\alpha < 1$ such that this implies that $\mathfrak{X}$ itself is a Johnson scheme?

A result in this direction could be a step toward an elementary characterization of the Cameron groups as the only primitive groups of large order, or somewhat less ambitiously, an elementary characterization of the Johnson groups as the only primitive groups of large order, without an imprimitive subgroup of small index. Steps toward these goals have been made in [Ba81] for the case $|G| > \exp(n^{1/2+\epsilon})$ and in a remarkable recent paper by Sun and Wilmes [SuW] for the case $|G| > \exp(n^{1/3+\epsilon})$. The Sun–Wilmes result is also significant in that it can replace the rather complicated Split-or-Johnson routine to obtain an $\exp(\widetilde{O}(n^{1/3}))$ GI test which would already break the old $\exp(\widetilde{O}(n^{1/2}))$ barrier; further progress in this direction has the potential of greatly simplifying the algorithm at the cost of heavy combinatorial structure theory.

Our last question is about finding *encasing subgroups.* If H is a proper subgroup of $\mathfrak{S}(\Omega)$ then we say that the group G *encases* H if $H \le G < \mathfrak{S}(\Omega)$. Typically H is the unknown target and G provides a step toward zooming in on H.

[5] To see that these problems belong to coAM, one can adapt the GMW protocol [GoMW] by conjugating the group by a random permutation and choosing a uniform random set of $O(n)$ generators.

[6] Luks's reduction is explained by Miyazaki in a post on The Math Forum, Sep. 29, 1996.

(a) Given a nontrivial graph X (not empty or complete), find a group that encases $\mathrm{Aut}(X)$. Can this be done in polynomial time?

(b) Given a nontrivial regular graph $X = (V, E)$ with $|V| = v$ vertices, find a small subset $S \subset V$ and a group G that encases the pointwise stabilizer $\mathrm{Aut}(X)_{(S)}$ and has exponentially large index in $\mathfrak{S}(V)$, i. e., $v!/|G| > c^v$ for some constant $c > 1$. Ideally we would want S to have bounded size and we would like to do this in polynomial time. – Instead of fixing vertices, we can also fix members of some other small class of objects canonically associated with X; here "small" should ideally mean "polynomially bounded." (In algorithmic terms, the bound corresponds to the multiplicative cost.)

These problems can be solved in quasipolynomial time by simply computing $\mathrm{Aut}(X)$. Finding such encasing groups more efficiently could replace the Split-or-Johnson rountine and make the algorithm more efficient and potentially more elegant.

5.5 Analyze This!

The purpose of the present paper is to give a guaranteed upper bound (worst-case analysis); it does not contribute to practical solutions. It seems, for all practical purposes, the Graph Isomorphism problem is solved; a suite of remarkably efficient programs is available (nauty, saucy, Bliss, conauto, Traces). The article by McKay and Piperno [McP] gives a detailed comparison of methods and performance.

These algorithms provide ingenious shortcuts in backtrack search. The success of Piperno's "experimantal paths" in his "Traces" program seems especially intriguing. An important question facing the theorist in this area is to analyze these algorithms.

Can these heuristics contribute to better worst-case bounds? As the heuristics get more sophisticated, it seems increasingly difficult to find families of pairs of graphs that fool them. Interesting families of graphs may turn up in the search for such pairs. Daniel Neuen and Pascal Schweitzer [NeS] have recently found an iterated CFI construction that may challenge the otherwise lightning fast "Traces" program by Adolfo Piperno [Pi].

The comparison charts in [McP] seem to suggest that we lack true benchmarks – difficult classes of graphs on which to compare the algorithms. Encoding class-2 p-groups as graphs could provide quasipolynomially difficult examples, but right now we have no guarantee that the heuristics could not be tricked into much worse, (moderately?) exponential behavior.

Acknowledgments

I am happy to acknowledge the inspiration gained from my recent collaboration on the structure, automorphism group, and isomorphism problem for highly regular combinatorial structures with my student John Wilmes as well as with Xi Chen, Xiaorui Sun, and Shang-Hua Teng [BaW1, BaCh+, BaW2]. The recent breakthrough on primitive coherent configurations by Sun and Wilmes [SuW] was particularly encouraging; at one point during the weeks before the completion of the present work, it served as a tool to breaking the decades-old $\exp(\widetilde{O}(\sqrt{n}))$ barrier.

The most direct forerunner of this paper was my joint work with Paolo Codenotti on hypergraph isomorphism [BaCo]; that paper combined the group theory method with a web of combinatorial partitioning techniques. In particular, I found an early version of the Design Lemma in the wake of that work. (A much simpler observation is called "Design Lemma" in that paper.)

Some of the group theory used in the present paper was inspired by my joint work with Péter Pál Pálfy and Jan Saxl [BaPS]; in particular, the rendering of a result of Feit and Tits [FeT] in that paper turned out to be particularly handy in the proof of the main group theoretic lemma of this paper ("Unaffected Stabilizers Lemma," Theorem 6).

I am grateful to three long-time friends who helped me verify critical parts of this paper: Péter Pál Pálfy and László Pyber the proof of various versions of the key group-theoretic result, and Gene Luks the LocalCertificates procedure (Sec. 4.1), the core algorithm of the paper. Their comments helped improve the presentation, and, more significantly, raised my confidence that these items actually worked. All other parts of the paper seem quite "fault-tolerant," with multiple solutions, and a bag of tricks to rely on, should any gaps be found. Naturally, any errors that may remain in these items (or any other part of the paper) are my sole responsibility.

I wish to thank several colleagues, and especially Thomas Klimpel and Péter P. Pálfy, for their careful reading of parts of the first arXiv version and pointing out a large number of typos and some inaccuracies. The second arXiv version corrected these and added some clarifications.

6. REFERENCES

[ArT] V. Arvind and Jacobo Torán: Isomorphism testing: Perspectives and open problems. *Bull. EATCS* **86**, June 2005.

[AsS] Michael Aschbacher and Leonard L. Scott: Maximal subgroups of finite groups. *J. Algebra* **92** (1985), 44–80.

[AtM] Albert Atserias and Elitza Maneva: Graph Isomorphism, Sherali–Adams Relaxations and Indistinguishability in Counting Logics. *SIAM J. Comp.* **42(1)**, 2013, 112–137.

[Ba77] László Babai: On the isomorphism problem. Manuscript, 1977. Cited in [Mat]

[Ba79a] László Babai: Monte Carlo algorithms in graph isomorphism testing. Tech. Rep. 79–10, Dép. Math. et Stat., Université de Montréal, 1979 (pp. 42) http://people.cs.uchicago.edu/∼laci/lasvegas79.pdf

[Ba79b] László Babai: Lectures on Graph Isomorphism. University of Toronto, Department of Computer Science. Mimeographed lecture notes, October 1979

[Ba81] László Babai: On the order of uniprimitive permutation groups. *Annals of Math.* **113(3)** (1981) 553–568.

[Ba83] László Babai: *Permutation Groups, Coherent Configurations and Graph Isomorphism.* D.Sc. Thesis (Hungarian), Hungarian Academy of Sciences, April 1983.

[Ba08] László Babai: Coset intersection in moderately exponential time. Manuscript, 2008. http://people.cs.uchicago.edu/∼laci/int.pdf

[Ba15] László Babai: Graph Isomorphism in quasipolynomial time. arXiv:1512.03547, 2015.

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

[BaCh+] László Babai, Xi Chen, Xiaorui Sun, Shang-Hua Teng, John Wilmes: Faster Canonical Forms For Strongly Regular Graphs. *In: 54th IEEE FOCS*, 2013, pp. 157–166.

[BaCo] László Babai, Paolo Codenotti: Isomorphism of hypergraphs of low rank in moderately exponential time. *In: Proc. 49th IEEE FOCS*, 2008, pp. 667–676.

[BaCGQ] László Babai, Paolo Codenotti, Joshua A. Grochow, Youming Qiao: Code Equivalence and Group Isomorphism. *In: Proc. 22nd Ann. Symp. on Discrete Algorithms (SODA'11)*, ACM-SIAM, 2011, pp. 1395–1408.

[BaCQ] László Babai, Paolo Codenotti, Youming Qiao: Polynomial-time Isomorphism Test for Groups with no Abelian Normal Subgroups (Extended Abstract). *In: Proc. 39th Internat. Colloq. on Automata, Languages and Programming (ICALP'12)*, Springer LNCS 7391, 2012, pp. 51–62.

[BaKL] László Babai, William M. Kantor, Eugene M. Luks: Computational complexity and the classification of finite simple groups. *In: Proc. 24th IEEE FOCS*, 1983, pp. 162–171.

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

[BaLS] László Babai, Eugene M. Luks, Ákos Seress: Permutation groups in NC. *In: Proc. 19th ACM STOC*, 1987, pp. 409–420.

[BaPS] László Babai, Péter P. Pálfy, Jan Saxl: On the number of p-regular elements in finite simple groups. *LMS J. Comput. and Math.*, **12** (2009) 82–119.

[BaS] László Babai, Ákos Seress: On the degree of transitivity of permutation groups: a short proof. *J. Combinatorial Theory—A* **45** (1987) 310–315.

[BaW1] László Babai, John Wilmes: Quasipolynomial-time canonical form for Steiner designs. *In: Proc. 45th ACM STOC*, 2013, pp. 261–270.

[BaW2] László Babai, John Wilmes: Asymptotic Delsarte cliques in distance-regular graphs. *J. Algebraic Combinatorics,* to appear. See arXiv:1503.02746

[Bo89] Alfred Bochert: Über die Zahl verschiedener Werthe, die eine Funktion gegebener Buchstaben durch Vertauschung derselben erlangen kann. *Math. Ann.* **33** (1889) 584–590.

[Bo92] Alfred Bochert: *Math. Ann.* **40** (1892) 176–193

[Bo97] Alfred Bochert: Über die Classe der transitiven Substitutionengruppen II. *Math. Ann.* **49** (1897) 133–144.

[CaiFI] Jin-Yi Cai, Martin Fürer, Neil Immerman: An optimal lower bound on the number of variables for graph identification. *Combinatorica* **12** (1992) 389–410.

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

[CST] Xi Chen, Xiaorui Sun, Shang-Hua Teng: Multi-stage design for quasipolynomial-time isomorphism testing of Steiner 2-systems. *In: Proc. 45th ACM STOC*, 2013, pp. 271–280.

[DiM] John D. Dixon, Brian Mortimer: *Permutation Groups.* Springer Grad. Texts in Math. vol. 163, 1996

[FeT] Walter Feit and Jacques Tits: Projective representations of minimum degree of group extensions. *Canad. J. Math.* **30** (1978) 1092–1102.

[FuHL] Merrick Furst, John Hopcroft, Eugene Luks: Polynomial-time algorithms for permutation groups. *In: Proc. 21st IEEE FOCS*, 1980, pp. 36–41.

[GlG] George Glauberman and Łukasz Grabowski: Groups with identical k-profiles. *Theory of Computing* **11(15)** (2015) 395–401

[GoMW] Oded Goldreich, Silvio Micali, and Avi Wigderson: Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. *In: Proc. 27th IEEE FOCS*, 1986, pp. 174–187.

[ImL] Neil Immerman, Eric S. Lander: Describing graphs: a first-order approach to graph canonization. *In: Complexity Theory Retrospective — in honor of Juris Hartmanis on the occasion of his 60th birthday, July 5, 1988* (Alan Selman, ed.), Springer 1990, pp. 59–81.

[Jor1] Camille Jordan: *Traité des substitutions et des equations algébriques.* Gauthier–Villars, 1870. (Reprinted 1957, Paris, Albert Blanchard)

[Jor2] Camille Jordan: Nouvelles recherches sur la limite de transivité des groupes qui ne contiennent pas le groupe alterné. *J. de Math. Pures et Appliquées* **1** (1895), 35–60.

[KlL] Peter Kleidman and Martin Liebeck: *The Subgroup Structure of the Finite Classical Groups.* London Math. Soc. Lecture Note Ser. Vol. 129, Cambridge Univ. Press, 1990.

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

[Lie83] Martin W. Liebeck: On graphs whose full automorphism group is an alternating group or a finite classical group. *Proc. London Math. Soc. (3)* **47** (1983) 337-362

[LiePS] Martin W. Liebeck, Cheryl E. Praeger, Jan Saxl: On the O'Nan–Scott theorem for finite primitive permutation groups. *J. Austral. Math. Soc. (A)* **44** (1988) 389–396

[Lu82] Eugene M. Luks: Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comput. Syst. Sci.* **25(1)** (1982) 42–65.

[Lu87] Eugene M. Luks: Computing the composition factors of a permutation group in polynomial time. *Combinatorica* **7** (1987) 87–99.

[Lu93] Eugene M. Luks: Permutation groups and polynomial-time computation. *In: Groups and Computation,* DIMACS Ser. in Discr. Math. and Theor. Computer Sci. **11** (1993) 139–175.

[Lu99] Eugene M. Luks: Hypergraph Isomorphism and Structural Equivalence of Boolean Functions. *In: 31st ACM STOC*, 1999, pp. 652-658.

[Mar] Attila Maróti: On the orders of primitive groups. *J. Algebra* **258(2)** (2002) 631–640.

[Mat] Rudi Mathon: A note on the graph isomorphism counting problem. *Info. Proc. Lett.* **8** pp. 131–132.

[McP] Brendan D. McKay and Adolfo Piperno: Practical Graph Isomoprhism, II. `arXiv:1301.1493`, 2013.

[Mi] Takunari Miyazaki: Luks's reduction of Graph isomorphism to code equivalence. Comment on The Math Forum, Sep. 29, 1996. `http://mathforum.org/kb/thread.jspa?forumID=253&threadID=561418&messageID=1681072#1681072`

[NeS] Daniel Neuen, Pascal Schweitzer: Iterated CFI graphs. Personal communication by P. S., 2015.

[OWWZ] Ryan O'Donnell, John Wright, Chenggang Wu, Yuan Zhou: Hardness of robust graph isomorphism, Lasserre gaps, and asymmetry of random graphs. *In: Proc. 25th ACM–SIAM Symp. Disr. Alg. (SODA'14)*, 2014, pp. 1659–1677.

[Pi] Adolfo Piperno: Search Space Contraction in Canonical Labeling of Graphs. `arXiv:0804.4881`, 2008, v2 2011.

[Py] László Pyber: On the orders of doubly transitive permutation groups, elementary estimates. *J. Combinatorial Theory, Ser A* **62(2)** (1993) 361–366.

[Py] László Pyber: Personal comunication, 2016.

[Ro] Joseph Rotman: *An Introduction to the Theory of Groups.* 4th ed., Springer, 1995.

[Sco] Leonard L. Scott: Representations in characteristic *p*. *In: The Santa Cruz Conference on Finite Groups*, 1980, Amer. Math. Soc., pp. 319–322.

[Se] Ákos Seress: *Permutation Group Algorithms.* Cambridge Univ. Press, 2003

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

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

[SnSC] Aaron Snook, Grant Schoenebeck, Paolo Codenotti: Graph Isomorphism and the Lasserre Hierarchy. `arXiv:1401.0758`

[Sp] Daniel A. Spielman: Faster Isomorphism Testing of Strongly regular Graphs. In: *Proc. 28th ACM STOC*, 1996, pp. 576–584.

[SuW] Xiaorui Sun and John Wilmes: Faster canonical forms for primitive coherent configurations. *In: Proc. 47th STOC*, 2015, pp. 693–702.

[To] Jacobo Torán: On the hardness of Graph Isomorphism. *SIAM J. Comp.* **33(5)** (2004) 1093–1108

[We] Boris Weisfeiler (ed.): *On Construction and Identification of Graphs.* Springer Lect. Notes in Math. Vol 558, 1976.

[WeL] Boris Weisfeiler, Andrei A. Leman: A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya* **9** (1968) 12–16.

[Wi34] Helmut Wielandt: Abschätzungen für den Grad einer Permutationsgruppe von vorgeschriebenem Transitivitätsgrad. Dissertation, Berlin, 1934. *Schriften Math. Seminars Inst. Angew. Math. Univ. Berlin* **2** (1934) 151–174.

[Wi60] Helmut Wielandt: Über den Transitivitätsgrad von Permutationsgruppen. *Math. Z.* **74** (1960) 297–298.

[Wi3] Helmut Wielandt: *Finite Permutation Groups.* Acad. Press, New York 1964.

[ZKT] Viktor N. Zemlyachenko, Nikolai M. Korneenko, Regina I. Tyshkevich: Graph isomorphism problem. *Zapiski Nauchnykh Seminarov LOMI* **118** (1982) 83–158, 215.