Canonical Form for Graphs in Quasipolynomial Time*

Preliminary Report

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

ABSTRACT

We outline how to turn the author's quasipolynomial-time graph isomorphism test into a construction of a canonical form within the same time bound. The proof involves a nontrivial modification of the central symmetry-breaking tool, the construction of a canonical relational structure of logarithmic arity on the ideal domain based on local certificates.

CCS CONCEPTS

• **Theory of computation → Graph algorithms analysis**.

KEYWORDS

algorithms, complexity, graphs, isomorphism, group theory, canonical form

ACM Reference Format:
László Babai. 2019. Canonical Form for Graphs in Quasipolynomial Time: Preliminary Report. In *Proceedings of the 51st Annual ACM SIGACT Symposium on the Theory of Computing (STOC '19), June 23–26, 2019, Phoenix, AZ, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3313276.3316356

1 INTRODUCTION

Let C be a class of finite graphs. A *canonical form* for the class C is an assignment $F : C \to C$ such that

(i) $(\forall X \in C)(F(X) \cong X)$
(ii) $(\forall X, Y \in C)(X \cong Y \iff F(X) = F(Y))$

Given an efficiently computable canonical form, the isomorphism problem for graphs in C can also be efficiently solved. The converse is not known, but so far the discovery of graph isomorphism (GI) testers has been followed by canonical forms for the same class of graphs with the same efficiency. The first paper that used group theory in the design of a GI test was [Ba79]; that paper gave a polynomial-time Las Vegas algorithm (and, incidentally, introduced the term "Las Vegas algorithm") for testing isomorphism of vertex-colored graphs with bounded color multiplicity. This algorithm was soon derandomized [FHL80] and was followed by a canonical form for the same class of graphs [BaKL]. Following Luks's seminal paper [Lu82] that solved GI in polynomial time for graphs of bounded valence, [BaL83] and [FSS83] constructed canonical forms in polynomial time for graphs of bounded valence by adapting Luks's algorithm.

In this paper we construct a canonical form for all graphs in quasipolynomial ($\exp(O((\log n)^c))$) time, by adapting the author's GI algorithm of the same complexity [Ba15+].

In the past, the bulk of the task in such adaptations consisted in carefully laying the conceptual groundwork, and this paper retains some of that aspect. What is different, however, is that now the core part of the GI test in question, the construction of a canonical t-ary relation on the "ideal domain" based on the "local certificates algorithm," is not directly adaptable, and requires the addition of a new algorithm — the main technical contribution of this paper (Theorem 12.1).

The problem with the original construction is that it is only *pairwise canonical*, meaning that the structures constructed depend on both inputs X and Y (of which we wish to decide isomorphism), and satisfy the canonicity requirement only with respect to the 2-element class $C = \{X, Y\}$. This is sufficient for isomorphism testing but not for the construction of a canonical form.

For a detailed understanding of our procedure, some familiarity with [Ba15+] may be necessary. However, we tried to make this writing self-contained by explaining, in some detail, all the background needed. These explanations range from the informal to the rigorous depending on their connection to our main technical contribution.

2 STRING ISOMORPHISM

Let Σ be a finite alphabet and Ω a finite set. We refer to Ω as the set of *positions*.

Strings are functions $\mathbf{x} : \Omega \to \Sigma$ (Σ-strings over the domain Ω). They form the set Σ^Ω.

$\mathrm{Sym}(\Omega)$ denotes the symmetric group acting on Ω (all permutations of Ω). Let $G \le \mathrm{Sym}(\Omega)$ be a permutation group acting on Ω. (The "$\le$" sign between groups indicates "subgroup.") We shall refer to G as the *ambient group*. G acts on the strings by the rule $\mathbf{x}^\sigma(u) = \mathbf{x}(u^{\sigma^{-1}})$ $(u \in \Omega, \sigma \in G)$. A permutation $\sigma \in \mathrm{Sym}(\Omega)$ is a *G-isomorphism* from the string $\mathbf{x}$ to the string $\mathbf{y}$ $(\mathbf{x}, \mathbf{y} \in \Sigma^\Omega)$ if $\sigma \in G$ and $\mathbf{x}^\sigma = \mathbf{y}$. We write $\mathrm{Iso}_G(\mathbf{x}, \mathbf{y})$ to denote the set of

*This research was partially supported by NSF Grant CCF 1718902. The views expressed in the paper are those of the author and do not necessarily reflect the views of the NSF.

STOC '19, June 23–26, 2019, Phoenix, AZ, USA

ACM ISBN 978-1-4503-6705-9/19/06…$15.00
https://doi.org/10.1145/3313276.3316356

G-isomorphisms from $\mathbf{x}$ to $\mathbf{y}$. The strings $\mathbf{x}, \mathbf{y}$ are G-isomorphic, denoted $\mathbf{x} \cong_G \mathbf{y}$, if $\mathrm{Iso}_G(\mathbf{x}, \mathbf{y})$ is not empty.

The *string isomorphism* (SI) problem, introduced by Luks [Lu82], asks, given as input the sets Ω, Σ, the group G, and strings $\mathbf{x}, \mathbf{y} \in \Sigma^\Omega$, to decide whether $\mathbf{x} \cong_G \mathbf{y}$. (Permutation groups are always given by a list of generators.) The main result of [Ba15+] is the following.

Theorem 2.1 ([Ba15+]). *SI can be solved in quasipolynomial time.*

In this paper we consider the canonization version of the String Isomorphism (SI) problem; we refer to this problem as "String Canonization" (SC).

A *G-canonical form* of Σ-strings over the domain Ω is a function $F : \Sigma^\Omega \to \Sigma^\Omega$ that selects one member from each G-orbit of strings. In other words, a function $F : \Sigma^\Omega \to \Sigma^\Omega$ is a G-canonical form if for all $\mathbf{x}, \mathbf{y} \in \Sigma^\Omega$

(i) $F(\mathbf{x}) \cong_G \mathbf{x}$, and
(ii) if $\mathbf{x} \cong_G \mathbf{y}$ then $F(\mathbf{x}) = F(\mathbf{y})$.

A "canonical form of strings" is a function that takes as input the sets Ω, Σ, (a set of generators of) the group G, and a string $\mathbf{x} \in \Sigma^\Omega$, and returns a G-canonical form $F(\mathbf{x})$. In this note we sketch the proof of the following main result.

Theorem 2.2. *There is a canonical form of strings that can be computed in quasipolynomial time.*

Theorem 2.1 is a corollary. Another corollary is the result stated in the title of this paper.

Corollary 2.3. *There is a canonical form of graphs that can be computed in quasipolynomial time.*

To infer Cor. 2.3 from Theorem 2.2, one uses the natural encoding of v-vertex graphs by $(0,1)$-strings of length $\binom{v}{2}$ under the action $S_v^{(2)}$ of the symmetric group S_v on pairs, as observed by Luks.

Our conceptual setup essentially follows [BaL83]. The proof consists in a reinterpretation of the algorithm given in [Ba15+], with one essential new element: a canonical construction of a relational structure on the "ideal domain," to replace the "pairwise canonical" construction given in [Ba15+]. Both constructions are based on the "local certificates algorithm," the core algorithm of [Ba15+]. However, the construction in [Ba15+] is canonical only with respect to the pair $\{\mathbf{x}, \mathbf{y}\}$ whose isomorphism we wish to test, and not canonical over the set of all strings. The reason is that the auxiliary relational structures constructed in [Ba15+] depend both on $\mathbf{x}$ and $\mathbf{y}$. For canonical forms, the structures must depend on $\mathbf{x}$ alone. This requirement introduced new technical problems and conceptual issues; the latter may be obscured by the simplicity of the solution.

3 PERMUTATION GROUPS: DEFINITIONS, NOTATION

Most of the definitions in this section are standard; we indicate where this is not the case. Our standard reference on permutation groups is [DiM96]. For algorithms in permutation groups we refer to [Se03].

We use the notation $[n] = \{1, \dots, n\}$. For groups G, H, the relation $H \le G$ means H is a *subgroup* of G. The *normalizer* of H in G is the largest subgroup of G in which H is a normal subgroup; it can be defined as $N_G(H) = \{\sigma \in G \mid \sigma^{-1} H \sigma = H\}$.

A *permutation group* acting on the set Ω is a subgroup $G \le \mathrm{Sym}(\Omega)$. For $\sigma \in G$ we write its action in the exponent: $\sigma : x \mapsto x^\sigma$. For a set $S \subseteq G$ we write $x^S = \{x^\sigma \mid \sigma \in S\}$ and for a set $A \subseteq \Omega$ we write $A^\sigma = \{x^\sigma \mid x \in A\}$. We say that the set $A \subseteq \Omega$ is *G-invariant* (or invariant under G) is $A^\sigma = A$ for all $\sigma \in G$.

The *degree* of G is $n = |\Omega|$. The *order* of G is $|G|$. We call Ω the *permutation domain*. We use the generic notation S_n for $\mathrm{Sym}([n])$ or for any symmetric group of degree n if we don't want to specify the permutation domain. The *alternating group* $\mathrm{Alt}(\Omega) \le \mathrm{Sym}(\Omega)$ consists of the even permutations; the generic notation is A_n. For $n \ge 2$ we have $|S_n : A_n| = 2$. For $n \ge 5$, the group A_n is simple (has no nontrivial normal subgroups). For convenience, this author likes to call $\mathrm{Sym}(\Omega)$ and $\mathrm{Alt}(\Omega)$ collectively the ***giants*** on Ω.

An *action* of a group G on a set Γ is a homomorphism $\varphi : G \to \mathrm{Sym}(\Gamma)$. Given such an action, we call Γ a *G-set*. If $G \le \mathrm{Sym}(\Omega)$ then Ω is a G-set under the identity action.

We extend the notation x^σ to actions: for $x \in \Gamma$ we write x^σ to denote $x^{\varphi(\sigma)}$ if the action φ is understood from the context.

Definition 3.1. The *stabilizer* of $x \in \Gamma$ in the subgroup $G_x \le G$ consisting of the elements that fix x:

$$G_x = \{\sigma \in G \mid x^\sigma = x\}\,. \tag{1}$$

The *orbit* of $x \in \Gamma$ under G is the set

$$x^G = \{x^\sigma \mid \sigma \in G\}\,. \tag{2}$$

The *length* of the orbit is its size, $|x^G|$.

The index of the stabilizer is the length of the orbit:

$$|G : G_x| = |x^G|\,. \tag{3}$$

The G-action $G \to \mathrm{Sym}(\Gamma)$ is *transitive* if $x^G = \Gamma$ for some (and therefore every) $x \in \Gamma$. A partition $\Pi = \{B_1, \dots, B_k\}$ of Γ into blocks $B_i \ne \emptyset$ means $\Gamma = \bigsqcup_{i=1}^k B_i$. We say that Π is a *system of imprimitivity* for G if $\Pi^G = \Pi$, i. e., every $\sigma \in G$ takes blocks to blocks. The *trivial* systems of imprimitivity are the discrete partition (each block has size 1) and the unit partition (there is just one block, namely, Γ). The G-action is *primitive* if $|\Gamma| \ge 2$, the G-action is transitive, and has no nontrivial systems of imprimitivity.

We say that a permutation group $G \le \mathrm{Sym}(\Omega)$ is transitive (primitive) if its idenity action is transitive (primitive, resp.). The orbits and blocks of imprimitivity of G are the orbits and blocks, resp., of this action.

A special case of the stabilizer notation: the *setwise stabilizer* of $T \subseteq \Gamma$ in G is the subgroup

$$G_T = \{\sigma \in G \mid T^\sigma = T\}\,. \tag{4}$$

By Eq. (3) we have $|G : G_T| \le \binom{m}{t}$, where $m = |\Gamma|$ and $t = |T|$. We note that T is G-invariant if and only if $G_T = G$.
The *pointwise stabilizer* of T is the subgroup

$$G_{(T)} = \{\sigma \in G \mid (\forall x \in T)(x^\sigma = x)\} = \bigcap_{x \in T} G_x\,. \tag{5}$$

If $\sigma, \tau \in G$ then the element $\tau^\sigma := \sigma^{-1}\tau\sigma$ is called the *conjugate* of τ by σ. Conjugation by a fixed element σ, i. e., the map $\tau \mapsto \tau^\sigma$, is an automorphism of G. For $S \subseteq G$ we write $S^\sigma := \sigma^{-1} S \sigma = \{\tau^\sigma \mid \tau \in S\}$. We shall need the following observation.

Fact 3.2. Let us consider a G-action $G \to \mathrm{Sym}(\Gamma)$. If $\sigma \in G$ and $T \subseteq \Gamma$ then

$$G_{T^\sigma} = (G_T)^\sigma \ . \tag{6}$$

Definition 3.3. If Γ is a G-set and $T \subseteq \Gamma$ is a G-invariant subset then we write G^T for the *restriction* of G to T, i. e., G^T is the image of the restriction homomorphism $G \to \mathrm{Sym}(T)$.

The notation G^T_T makes sense regardless of whether T is G-invariant: it means first we reduce G to G_T, the setwise stabilizer of T, and then restrict the G_T-action to T.

We write $\binom{\Omega}{t}$ for the set of t-subsets of Ω. A group $G \le \mathrm{Sym}(\Omega)$ induces a G-action on $\binom{\Omega}{t}$. We denote this action by $G^{(t)} \le \mathrm{Sym}\left(\binom{\Omega}{t}\right)$. This author likes to refer to $S_n^{(t)}$ and $A_n^{(t)}$ as the *Johnson groups* because of their action on the *Johnson graphs* (a standard term). So the Johnson groups have degree $\binom{n}{t}$ and order $n!$ or $n!/2$.

Fact 3.4. For $1 \le t < n/2$, *the Johnson groups are primitive.*

This author calls an action $\varphi : G \to \mathrm{Sym}(\Gamma)$ a ***giant action*** if $\varphi(G)$ is a giant on Γ, i. e., $\varphi(G) \ge \mathrm{Alt}(\Gamma)$.

We define the central new concept introduced in [Ba15+].

Definition 3.5 (Affected element)**.** Let Ω, Γ be sets, $G \le \mathrm{Sym}(\Omega)$ and $\varphi : G \to \mathrm{Sym}(\Gamma)$ a giant action. We say that $x \in \Omega$ is *affected* by φ if $\varphi(G_x)$ is not a giant on Γ.

4 SUBCOSETS

For a group G, subsets $A, B \subseteq G$, and $g \in G$ we use the notation $A^{-1} = \{a^{-1} \mid a \in A\}$ and $AB = \{ab \mid a \in A, b \in B\}$ and $gA = \{ga \mid a \in A\}$.

A *subcoset* of a group G is a set of the form aH for some $H \le G$ and $a \in G$. We could call this a *left subcoset*, but right and left subcosets are the same: $aH = (aHa^{-1})a$. In particular, if L is a subcoset of G then for every $a, b \in G$ the set aLb is also a subcoset of G.

We shall say that $H \le G$ is the *right subgroup* corresponding to the subcoset L if $L = aH$ for some $a \in G$. This subgroup is uniquely determined by L, namely, $H = L^{-1}L$.

A subcoset L will always be represented concisely by a set of generators of its right subgroup H and a coset representative $a \in L$.

A subcoset K of a subcoset L of G is a subcoset of G contained in L. With this definition, the subcoset relation is transitive.

We shall use the term "possibly empty subcoset" to describe a set that is either a subcoset or empty. This relation is also transitive. The family of possibly empty subcosets of a group is closed under intersection. Therefore we can speak of the *subcoset C generated by a subset $S \subseteq G$*; C is the intersection of all subcosets containing S. If $S = \emptyset$ then $C = \emptyset$; otherwise $C = c \cdot \langle c^{-1}S \rangle$ where c is any element of C and $\langle S \rangle$ denotes the subgroup generated by S.

For subsets C_1, C_2 of a group G we write $C_1 \le_c C_2$ if C_1 is a possibly empty subcoset of C_2, i. e., C_1 is a possibly empty subcoset of G and $C_1 \subseteq C_2$.

The significance of this concept to us is in the fact that the set

$$\mathrm{Iso}_G(\mathbf{x}, \mathbf{y}) = \{\sigma \in G \mid \mathbf{x}^\sigma = \mathbf{y}\}, \tag{7}$$

the set of G-isomorphisms of strings $\mathbf{x}$ and $\mathbf{y}$, is a possibly empty subcoset of G.

We shall also need to consider the set of isomorphisms within a subcoset: for $C \le_c G$ we set

$$\mathrm{Iso}_C(\mathbf{x}, \mathbf{y}) = \{\sigma \in C \mid \mathbf{x}^\sigma = \mathbf{y}\}. \tag{8}$$

While this extension of the Iso_G operator will be very convenient, it is not more general than isomorphisms with respect to subgroups. Indeed, if $C = \sigma H$ where $H = C^{-1}C \le G$ is the right subgroup corresponding to C then

$$\mathrm{Iso}_C(\mathbf{x}, \mathbf{y}) = \sigma \, \mathrm{Iso}_H(\mathbf{x}^\sigma, \mathbf{y}). \tag{9}$$

5 CANONIZATION FROM LUKS'S SI ALGORITHM

The canonization version of Luks's SI algorithm [Lu82] is described in [BaL83]. We retain that method as the basic framework of our algorithm.

The algorithm starts with fixing an arbitrary ordering of Ω, so we shall treat Ω as an ordered set. This also defines a lexicographic ordering of the subsets of Ω by representing every subset as a string of its elements listed in increasing order.

Following [BaL83], in this section we introduce *canonical placement cosets* that give our basic conceptual setup. Then we extract from [BaL83] the two basic routines used by the algorithm; we call them the *Chain Rule* and *Descent*.

5.1 Canonical Placement Coset

For purposes of recursion, rather than just constructing the canonical form F, we construct the *canonical placement coset*

$$\mathrm{CP}_G(\mathbf{x}) = \mathrm{Iso}_G(\mathbf{x}, F(\mathbf{x})). \tag{10}$$

Also for the purposes of recursion, we need to extend the concept of canonical placement cosets in two directions. First, we replace the group G with a subcoset $C = \sigma H \le_c G$.

Second, we look at our string through a *window*, i. e., an H-invariant subset $W \subseteq \Omega$ where $H = C^{-1}C$ is the right subgroup of C. For $\mathbf{x} \in \Sigma^\Omega$, let $\mathbf{x}^W$ denote the string defined by

$$\mathbf{x}^W(x) = \begin{cases} \mathbf{x}(x) & \text{for } x \in W \text{ and} \\ \beta & \text{for } x \in \Omega \setminus W \end{cases}$$

where β is a "dummy symbol," not belonging to the alphabet Σ, so $\mathbf{x}^W \in (\Sigma \sqcup \{\beta\})^\Omega$. Canonization of the strings $\mathbf{x}^W$ over G is equivalent to canonization of the strings in Σ^W over the restriction of G to W. For $\mathbf{x}, \mathbf{y} \in \Sigma^\Omega$ we write $\mathrm{Iso}^W_C(\mathbf{x}, \mathbf{y}) := \mathrm{Iso}_C(\mathbf{x}^W, \mathbf{y}^W)$ and $\mathrm{CP}^W_C(\mathbf{x}) := \mathrm{CP}_C(\mathbf{x}^W)$.

Definition 5.1. A *canonical placement function* CP takes as input a set Ω, an alphabet Σ, a group $G \le \mathrm{Sym}(\Omega)$, a subcoset $C \le_c G$, a window $W \subseteq \Omega$ that is invariant under the right subgroup $H := C^{-1}C$, and a string $\mathbf{x} \in \Sigma^\Omega$. It outputs a subcoset $\mathrm{CP}^W_C(\mathbf{x}) \le_c C$. The function CP obeys the following rules for all $\mathbf{x} \in \Sigma^\Omega$.

(i) For all $\sigma \in G$ we have $\mathrm{CP}^W_{\sigma C}(\mathbf{x}) = \sigma \, \mathrm{CP}^W_C(\mathbf{x}^\sigma)$

(ii) $\mathrm{CP}^W_C(\mathbf{x}) = \tau \cdot \mathrm{Aut}_H(\mathbf{x}^\tau)$ for every $\tau \in \mathrm{CP}^W_C(\mathbf{x})$.

Fixing Ω and G, we say that CP is a canonical placement function for (Ω, G).

Remark 5.2. Some comments are in order to indicate the self-consistency of this definition. First, the right subgroup of C and

of σC is the same, so the left-hand side of item (i) satisfies the invariance condition for the window.

Second, item (i) is consistent with multiplication in G. Indeed, for $\sigma, \tau \in G$ we have $\mathrm{CP}^W_{\sigma\tau C}(\mathbf{x}) = \sigma\tau\, \mathrm{CP}^W_C(\mathbf{x}^{\sigma\tau}) = \sigma\, \mathrm{CP}^W_{\tau C}(\mathbf{x}^{\sigma})$.

Proposition 5.3. *If* CP *is a canonical placement function for* (Ω, G) *then for all strings* $\mathbf{x} \in \Sigma^\Omega$ *we have* $|\mathbf{x}^{\mathrm{CP}^\Omega_G(\mathbf{x})}| = 1$ *and setting* $\mathbf{x}^{\mathrm{CP}^\Omega_G(\mathbf{x})} = \{F(\mathbf{x})\}$ *we obtain a* G*-canonical form* F*. Moreover,* CP *and* F *satisfy Eq. (10). Uniqueness also holds for windows: for a subcoset* $C \le_c G$ *with right subgroup* $H = C^{-1}C$ *and an* H*-invariant window* W *we have* $|(\mathbf{x}^{\mathrm{CP}^W_C(\mathbf{x})})^W| = 1$.

Def. 5.1 focuses on left shifts $C \mapsto \sigma C$. We state the consequences of the definition regarding right shifts. We shall need these at the end of Sec. 16.

Observation 5.4. *Let* $G \le \mathrm{Sym}(\Omega)$ *and let* CP *be a canonical placement function for* (Ω, G)*. Let* $C \le_c G$ *be a subcoset of* G*. Let* $W \subseteq \Omega$ *be a* G*-invariant window. Then for all* $\sigma \in G$ *we have*

$$\mathrm{CP}^W_{C\sigma}(\mathbf{x}) = \sigma \cdot \mathrm{CP}^W_{C\sigma}(\mathbf{x}^\sigma)\,. \tag{11}$$

Proof. Combine item (i) of Def. 5.1 with the observation that $C\sigma = \sigma C^\sigma$. □

5.2 Chain Rule

Let $G \le \mathrm{Sym}(\Omega)$ and $C \le_c G$ a subcoset of G with right subgroup $H = C^{-1}C$. Let $W \subseteq \Omega$ be a nonempty H-invariant subset.

If H is intransitive on W, we may apply the Chain Rule which takes an ordered partition $\Pi = (W_1, \dots, W_k)$ of the window W into H-invariant subwindows W_i, i. e., $W = W_1 \sqcup \cdots \sqcup W_k$ and $W_i^H = W_i$. The algorithm processes the W_i in succession,

$$C_0 := C, \quad C_i = \mathrm{CP}^{W_i}_{C_{i-1}}(\mathbf{x}) \quad (i = 1, \dots, k) \tag{12}$$

and returns $\mathrm{CP}^W_C(\mathbf{x}) := C_k$.

Remark 5.5. We do not need that all windows be H-invariant; it suffices that W_i is invariant under $H_i = C_{i-1}^{-1}C_{i-1}$. This follows by repeatedly applying the case $k = 2$ to the partitions $(U_i, W \setminus U_i)$, where $U_i = W_1 \sqcup \cdots \sqcup W_i$. Viewing the process this way will be important for the CP version of the Local Certificates algorithm (Sec. 16).

Next we justify this as a valid recursive step.

Proposition 5.6. *If the function* $\mathrm{CP}^{W_i}_D$ *satisfies Def. 5.1 for each* i *and all* $D \le_c C$ *then the function* CP^W_C *constructed by the Chain Rule also satisfies Def. 5.1.*

5.3 Ordering the Windows

To apply the Chain Rule, we need an ordering of the subwindows (blocks of the partition). If no such ordering is prescribed by the algorithm, we use an ordering that does not depend on any of the input parameters $(G, C, \mathbf{x})$, except on the ordering of Ω. One recipe, given in [BaL83], is to order the blocks according to the lexicographic order inherited from the ordering of Ω. Another recipe we apply in the algorithm is to order the sets by magnitude, $|W_1| \ge |W_2| \ge \cdots \ge |W_k|$, breaking ties lexicographically.

5.4 Descent

Luks's second main operation is breaking up a group into cosets of a subgroup. More generally, let C be a subcoset of the group G and $C = \bigsqcup_{\sigma \in R} \sigma D$ where D is a subcoset of C corresponding to a subgroup $K \le H$, where $H = C^{-1}C$ and $K = D^{-1}D$. R is a set of coset representatives. In this case, evidently,

$$\mathrm{Iso}^W_C(\mathbf{x}, \mathbf{y}) = \bigsqcup_{\sigma \in R} \mathrm{Iso}^W_{\sigma D}(\mathbf{x}, \mathbf{y})\,. \tag{13}$$

Noting that, according to Eq. (9), computing the Iso_C operator requires computing Iso_H and Iso_D requires Iso_K, this equation reduces the calculation of an instance of Iso_H to $|H : K|$ instances of Iso_K. This means a multiplicative cost of $|H : K|$ which must be compensated for a significantly improved "quality" of the subgroup K compared to H. For instance, if H is transitive on W and K is an intransitive normal subgroup then we can descend from H to K and apply the Chain Rule to K with great efficiency—a key technique used by Luks [Lu82].

Here is the canonization version of this routine.

Let $C = \bigsqcup_{\sigma \in R} \sigma D$ where $D \le_c C \le_c G$. Note that, by Prop. 5.3, the set $\left(\mathbf{x}^{\mathrm{CP}^W_{\sigma D}(\mathbf{x})}\right)^W$ consists of a single element; call it $\mathbf{y}(\sigma)$. Let $\mathbf{y}_{\min}$ be the lexicographic leader among all the $\mathbf{y}(\sigma)$ and consider the set

$$\mathscr{D} = \left\{ \sigma D \,\middle|\, \sigma \in R \text{ and } \left(\mathbf{x}^{\mathrm{CP}^W_{\sigma D}(\mathbf{x})}\right)^W = \{\mathbf{y}_{\min}\} \right\}. \tag{14}$$

Define $\mathrm{CP}^W_C(\mathbf{x})$ as the subcoset generated by the union of the subcosets in $\mathscr{D}$.

In analogy with Prop. 5.6, we state that descent is a valid recursive step.

Proposition 5.7. *Let* $C \le_c G$ *and* $H = C^{-1}C$ *the right subgroup of* C*. Let* $K < H$*. If the function* CP^W_D *satisfies Def. 5.1 for every left coset* D *of the subgroup* K *then the function* CP^W_C *constructed by the descent algorithm also satisfies Def. 5.1.*

5.5 The CP Algorithm from [BaL83]

Initially let $W := \Omega$.

If $|W| = 1$, we set $\mathrm{CP}^W_C(\mathbf{x}) = C$.

If G is intransitive on W, we apply the Chain Rule to reduce to the transitive case. If G is transitive on W, we select the "first" minimal system of imprimitivity (maximal blocks), $\{W_1, \dots, W_k\}$, so $W = \bigsqcup W_i$. This can be done in polynomial time (see [BaL83], also for the definition of "first"). The G-action permutes the W_i; this induces a G-action $\varphi : G \to S_k$. Let $K = \ker(\varphi)$; so the image $\varphi(G)$ is a primitive group, isomorphic to G/K. Noting that the W_i are K-invariant, we descend to K and then apply the Chain Rule to the partition $W = \bigsqcup W_i$. For the Chain Rule to be applicable, we order the subwindows lexicographically, following the first recipe mentioned in Sec. 5.3.

5.6 Complexity

If we have a bound of the form $|G^*| \le k^{g(n)}$ then we get the following recurrence. Let $f(n, m)$ denote the maximum size of the recursion tree corresponding to evaluating $\mathrm{CP}^W_G(\mathbf{x}, \mathbf{y})$ over a hereditary class of groups G (closed under subgroups and quotients),

where $n = |\Omega|$ and $m = |W|$. Then

$$f(n, m) \le k^{g(n)+1} f(n, m/k)\,. \tag{15}$$

This recurrence evaluates to $f(n, n) \le n^{g(n)+1}$. The computational cost associated with each link (edge) in the recursion tree is polynomial, so the total cost is $n^{g(n)+O(1)}$.

For isomorphism of graphs of bounded valence, Luks [Lu82] noted that the relevant groups have bounded composition factors. Primitive groups with this property have polynomially bounded order [BaCP82], i. e., $g(n) = O(1)$; therefore CP for strings with respect to such groups can be constructed in polynomial time. This in turn results in polynomial-time CP for graphs of bounded valence.

We note that Luks's original implementation [Lu82] employed one more descent step which permitted an easier polynomial-time analysis.

We also note that if we set $g(n) = O(\log n)$ then we obtain $f(n, n) = n^{O(\log n)}$, our target threshold for the naive application of Luks's method. If we encounter a primitive group of order greater than $k^{2\log_2 n}$ (where, as before, k is the number of blocks of imprimitivity) then we invoke the new group-theoretic and combinatorial techniques from [Ba15+] to achieve a more favorable recurrence.

6 THE LUKS BOTTLENECK

Luks's SI algorithm and along with it the CP algorithm of [BaL83] described in Sec. 5.5 reaches a bottleneck when it encounters a large primitive group. For our purposes, the primitive group $G^* \le S_k$ is "large" if $|G^*| \ge k^{2\log_2 n}$. (Note that in our context, $k \le n$.)

Using Cameron's classification of large primitive groups [Cam81], one can show that these groups G^* have a (normal) subgroup G^{**} of index $\le k$ such that G^{**} has *giant action* on some set Γ to which we refer as the *ideal domain*. This action lifts to an epimorphim (surjective homomorphism) $\varphi : G \to H$ where $H = \mathrm{Sym}(\Gamma)$ or $\mathrm{Alt}(\Gamma)$.

Given G^*, the group G^{**}, the set Γ, and the epimorphism φ can be constructed from G^* in polynomial time.

So by applying descent to G^{**} we may assume we have a giant action $\varphi : G \to \mathrm{Sym}(\Gamma)$. We assume $|\Gamma| > 2t$ where $t = \max\{9, \lfloor 3 + \log_2 n \rfloor\}$ is an important threshold that derives from the "Unaffected Stabilizers Lemma" [Ba15+]. The letter t will denote this quantity throughout this note.

7 DIVIDE AND CONQUER STRATEGY WITH QUASIPOLYNOMIAL TARGET

We shall work with the two parameters $n = |\Omega|$ and $m = |\Gamma|$. We seek to reduce an instance of the problem to a moderate number of significantly smaller instances. The "moderate number" is the branching factor (number of children) in the recursion tree, to which we also refer as the **multiplicative cost** (see Eq. (16) below). We shall want to keep it quasipolynomially bounded. "Significantly smaller" means we reduce the relevant parameter by at least 10%.

Most of the time, the relevant parameter will be m, so we get the recursion

$$f(n, m) \le q(n) f(n, 0.9m) \tag{16}$$

where $f(n, m)$ is the worst cost under these parameters, and $q(n)$ is the multiplicative cost mentioned. This recursion bottoms out when m gets to small ($m < 10\log_2 n$). At that point we individualize all elements of Γ, resulting in a significant reduction of n. This gives us two nested loops, each permitting $O(\log n)$ iterations, so the total cost will be $q(n)^{O((\log n)^2)}$. This is quasipolynomially bounded as long as $q(n)$ is.

8 STRONGLY CANONICAL ASSIGNMENTS

We often assign structures to structures in a way that preserves isomorphisms. We call such assignments *strongly canonical* (called "canonical" in [Ba15+]). An example is the classical isomorphism rejection method called "naive vertex refinement." Initially we color each vertex of a graph by its degree, and then we refine the coloring: each vertex learns the number of its neighbors in each color and encodes this information in its own refined color. We repeat this refinement step until the color partition stabilizes. If X is the original graph and $\mathfrak{X}(X)$ is the colored set obtained then $\mathrm{Iso}(X, Y) \subseteq \mathrm{Iso}(\mathfrak{X}(X), \mathfrak{X}(Y))$.

Definition 8.1 (groupoid)**.** A *small category* is a category in which the objects form a set (as opposed to a proper class). A *groupoid* is a small category in which all morphisms are invertible, i. e., $\mathrm{Hom}(X, Y) = \mathrm{Iso}(X, Y)$ for all pairs (X, Y) of objects.

Definition 8.2 (Strongly canonical assignment)**.** Let $\mathscr{C}$ and $\mathscr{D}$ be groupoids. We say that an assignment $X \mapsto \mathfrak{X}(X)$ of $X \in \mathrm{Ob}(\mathscr{C})$ to $\mathfrak{X}(X) \in \mathrm{Ob}(\mathscr{D})$ is *strongly canonical* if it comes from a functor $F : \mathscr{C} \to \mathscr{D}$, i. e., there is a functor F such that $\mathfrak{X}(X) = F(X)$ for all $X \in \mathrm{Ob}(\mathscr{C})$.

The structure added by a strongly canonical assignment (such as a refined coloring) can help in computing canonical forms. The following easy observation formalizes this.

Definition 8.3 (Diagonal groupoid)**.** Let $\mathscr{C}$ and $\mathscr{D}$ be groupoids. Let $H : \mathscr{C} \to \mathscr{D}$ be a functor. Define the category $\mathrm{diag}_H(\mathscr{C}, \mathscr{D})$ by having objects $(X, H(X))$ ($X \in \mathrm{Ob}(\mathscr{C})$) and morphisms $(\pi, H(\pi))$ for morphisms π in $\mathscr{C}$.

Observation 8.4. *Assume F is a canonical form for the diagonal groupoid $\mathrm{diag}_H(\mathscr{C}, \mathscr{D})$. For $X \in \mathrm{Ob}(\mathscr{C})$ define $F^*(X)$ as the first component of $F(X, H(X))$. Then F^* is a canonical form for $\mathscr{C}$.*

Proof. (i) The fact that $(X, H(X)) \cong F(X, H(X))$ in the diagonal groupoid implies isomorphism of the first components, $X \cong F^*(X)$ in $\mathscr{C}$.
(ii) Now suppose $X \cong Y$ in $\mathscr{C}$ and let $\pi \in \mathrm{Iso}_{\mathscr{C}}(X, Y)$. Applying H it follows that $H(\pi) \in \mathrm{Iso}(H(X), H(Y))$ in $\mathscr{D}$ and therefore $(\pi, H(\pi)) \in \mathrm{Iso}((X, H(X)), (Y, H(Y)))$ in $\mathrm{diag}_H(\mathscr{C}, \mathscr{D})$. So these objects in $\mathrm{diag}_H(\mathscr{C}, \mathscr{D})$ are isomorphic; therefore $F(X, H(X)) = F(Y, H(Y))$. It follows that the first components of these objects are equal: $F^*(X) = F^*(Y)$. □

The groupoids we consider are G-sets. Strongly canonical assignments among G-sets have a simple description.

Definition 8.5 (G-sets as groupoids)**.** A G-set Ω can be viewed as a groupoid $\mathscr{C}$: $\mathrm{Ob}(\mathscr{C}) = \Omega$ and the morphisms are pairs (x, σ) where $x \in \Omega$ and $\sigma \in G$. The source of the morphism (x, σ) is x and the target is x^σ. Composition is done in the natural way. Let us denote this category $\mathscr{C}(\Omega, G)$.

Definition 8.6. Let Ω and Γ be G-sets. A function $f : \Omega \to \Gamma$ is *G-equivariant* if

$$(\forall x \in \Omega)(\forall \sigma \in G)(f(x^\sigma) = f(x)^\sigma). \tag{17}$$

Observation 8.7 (Functor between G-sets)**.** *If Ω and Γ are G-sets then there is a 1-1 correspondence between functors $F : \mathscr{C}(\Omega, G) \to \mathscr{C}(\Gamma, G)$ and G-equivariant maps $f : \Omega \to \Gamma$, given by the equations $F(x) = f(x)$ and $F(x, \sigma) = (f(x), \sigma)$ $(x \in \Omega, \sigma \in G)$.*

9 BREAKING THE SYMMETRY

9.1 Canonical Structure on Ideal Domain

The ideal domain Γ is homogeneous with respect to G: the action $\varphi : G \to \mathrm{Sym}(\Gamma)$ is a giant action. If $\mathrm{Aut}_G(\mathbf{x})$ acts as a giant on at least 90% of Γ, we can discover this fact (even though $\mathrm{Aut}_G(\mathbf{x})$ is not known) and produce CP via efficient recurrence (Chain Rule on $\ker(\varphi)$).

Otherwise, the *Local Certificates* algorithm makes the absence of such high symmetry explicit by finding a not too highly symmetrical (see Def. 9.1) t-ary relational structure $\mathfrak{X}$ on Γ, where $t = O(\log n)$, such that $\mathfrak{X}$ invariant under the action of $\mathrm{Aut}_G(\mathbf{x})$ on Γ (via φ). This is the central algorithm of [Ba15+].

We review the parameters:
$n = |\Omega|$ – the size of the set of positions; $G \le \mathrm{Sym}(\Omega)$
$m = |\Gamma|$ – the size of the ideal domain; we have $10 \log_2 n \le m \le n$
$t = 3 + \lfloor \log_2 n \rfloor\}$ — We refer to all t-subsets $T \subset \Gamma$ as *test sets*.

To avoid trivialities, we may assume that n is greater than any given constant. In particular, $n \ge 64$ guarantees $t \ge 9$, needed for the application of the Unaffected Stabilizers Lemma.

The Local Certificates algorithm [Ba15+] produces, for each test set T, either a *fullness certificate* or a *non-fullness certificate*. We define these in Section 10; let us only state here that a fullness certificate is a subgroup of $\mathrm{Aut}_G(\mathbf{x})$, and a non-fullness certificate is a subgroup of $\mathrm{Sym}(T)$.

Let $F \le \mathrm{Sym}(\Gamma)$ denote the subgroup of $\mathrm{Sym}(\Gamma)$ generated by the φ-images of the fullness certificates. If F fixes no more than 90% of Γ, we find a rich set of G-automorphisms of $\mathbf{x}$ that permits efficient recursion for CP by applying the Chain Rule, following essentially verbatim the description in the "Aggregation of certificates" section in [Ba15+].

In case F fixes more than 90% of Γ, we may assume, as in [Ba15+], that no t-tuple receives a fullness certificate. In [Ba15+], the non-fullness certificates are used to construct *pairwise canonical t-ary relational structures* $\mathfrak{X}$ and $\mathfrak{Y}$ on Γ corresponding to the input strings $\mathbf{x}$ and $\mathbf{y} \in \Sigma^\Omega$ of which we wish to decide G-isomorphism. Here "pairwise canonical" means canonicity with respect to the category having just two objects, $\mathbf{x}$ and $\mathbf{y}$. The structures $\mathfrak{X}, \mathfrak{Y}$ are required to have not too much symmetry.

Definition 9.1 (Symmetricity)**.** Let $\mathfrak{X} = (\Gamma, \mathcal{R})$ be a relational structure on the underlying set Γ. We say that a subset $\Delta \subseteq \Gamma$ is *symmetrical* if the setwise stabilizer of Δ in $\mathrm{Aut}(\mathfrak{X})$ acts as the symmetric group on Δ. The *symmetricity* of $\mathfrak{X}$ is the maximum size of a symmetrical subset.

It would suffice for our algorithm for the structures $\mathfrak{X}, \mathfrak{Y}$ to have symmetricity $\le 0.9|\Gamma|$. In fact, the structures constructed in [Ba15+] have tiny symmetricity ($\le t - 1$), and this continues to hold for the modified construction given in this note (see Claim 12.6).

This step (the construction of pairwise canonical t-ary relational structures on Γ) is followed by combinatorial partition algorithms ("Design Lemma" and "Split-or-Johnson") to find a subgroup $H \le G$ that "encases" the unknown group $\mathrm{Aut}_G(\mathbf{x})$ (i. e., $\mathrm{Aut}_G(\mathbf{x}) \le H$) such that, by certain measures, H is significantly smaller than G, thus making the symmetry breaking explicit.

9.2 Combinatorial Partitioning: Individualization, Refinement

The combinatorial partitioning algorithms employ two basic steps: *individualization* and *strongly canonical refinement*. Both of these are classical isomorphism–refutation tools.

Individualization means assigning a special color to an element x of a G-space (colors are preserved by isomorphisms by definition). This amounts to a descent from G to G_x and incurs a multiplicative cost of $|G : G_x| = |x^G|$.

Canonical refinement refines the coloring in a manner that is *strongly canonical*.

Both of these operations work for CP without change (individualization by the CP version of descent, Sec. 5.4, and canonical refinement trivially).

So what we need to focus on is replacing the pairwise canonical construction (of the t-ary relational structure with small symmetricity) by a canonical construction. Our canonical construction will yield a $(2t)$-ary relational structure, doubling the arity used in [Ba15+].

10 FULLNESS AND NON-FULLNESS CERTIFICATES

Let $T \subseteq \Gamma$ be a test set (a subset of size t). While in general, computing setwise stabilizers is Cook-equivalent to the SI problem, the setwise stabilizer G_T is easy to compute since it is the setwise stabilizer in a giant action: $G_T = \varphi^{-1}(\mathrm{Sym}(\Gamma)_T)$. Note that $\mathrm{Sym}(\Gamma)_T = \mathrm{Sym}(T) \times \mathrm{Sym}(\Gamma \setminus T)$.

Let ψ_T denote the epimorphism $G_T \twoheadrightarrow \mathrm{Sym}(T)$ obtained by restricting the domain of φ to G_T and then restricting the codomain to $\mathrm{Sym}(T)$. (This is a surjection because $|\Gamma| \ge t + 2$.)

We say that the test set T is *full* (with respect to the input string $\mathbf{x}$) if $\psi_T : \mathrm{Aut}_{G_T}(\mathbf{x}) \to \mathrm{Sym}(T)$ is a giant action. A *fullness certificate* is a subgroup $K(T) \le \mathrm{Aut}_G(\mathbf{x})$ such that $\psi_T(K(T)) = \mathrm{Alt}(T)$ or $\mathrm{Sym}(T)$. As mentioned above, we may assume that *none of the test sets is full.*

A *non-fullness certificate* is a subgroup $M(T, \mathbf{x}) \le \mathrm{Sym}(T)$ that is not a giant, i. e., $M(T, \mathbf{x}) \not\ge \mathrm{Alt}(T)$, such that $\psi_T(\mathrm{Aut}_{G_T}(\mathbf{x})) \le M(T, \mathbf{x})$. Note that the group $\mathrm{Aut}_{G_T}(\mathbf{x})$ is not known; the algorithm nevertheless must guarantee the stated inclusion. In [Ba15+], such a certificate is constructed for each T, along with a "window" $W(T, \mathbf{x}) \subseteq \Omega$ such that

(a) $W(T, \mathbf{x})$ is invariant under $\mathrm{Aut}_{G_T}(\mathbf{x})$,
(b) $M(T, \mathbf{x}) = \psi_T(\mathrm{Aut}_{G_T}(\mathbf{x}^{W(T,\mathbf{x})}))$,

where $\mathbf{x}^{W(T,\mathbf{x})}$ is the restriction of $\mathbf{x}$ to $W(T, \mathbf{x})$, all other positions being filled with the dummy symbol $\beta \notin \Sigma$.

Both $W(T, \mathbf{x})$ and $\mathrm{Aut}_{G_T}(\mathbf{x}^{W(T,\mathbf{x})})$ are computed via efficient recursion, applying the Chain Rule to the window, thanks to the Affected Orbit Lemma (part (b) of Theorem 13.1).

10.1 Strong Canonicity of Local Certificates

The next observation is the starting point of the proof of Theorem 12.1.

Lemma 10.1. *The assignment* $(T, \mathbf{x}) \mapsto (T, W(T, \mathbf{x}), M(T, \mathbf{x}))$ *is strongly canonical. Here we view each side naturally as elements in a* G*-set.*

Proof. First we note that the set $\binom{\Gamma}{t} \times \Sigma^\Omega$ is a G-set under the actions natural in our context. Indeed, G acts on Γ via φ and this defines an induced G-action on $\binom{\Gamma}{t}$. Moreover, the G-action on Ω induces an action on Σ^Ω. Let us now consider the set $\mathcal{H} = \mathcal{P}(\Omega) \times \mathcal{M}$ where $\mathcal{P}(\Omega)$ is the powerset of Ω and $\mathcal{M}$ is the set of pairs (T, M) where $T \in \binom{\Gamma}{t}$ and $M \le \mathrm{Sym}(T)$. So $\mathcal{H}$ is also a G-set, where G acts in the obvious way on $\mathcal{P}(\Omega)$ and we define the action of $\pi \in G$ on $\mathcal{M}$ by $(T, M)^\pi = (T^\pi, M^{\varphi(\pi)})$ where $M^\sigma = \sigma^{-1} M \sigma$. Now $\left\{(T, W(T, \mathbf{x}), M(T, \mathbf{x})) \mid \mathbf{x} \in \Sigma^\Omega, T \in \binom{\Gamma}{t}\right\}$ is also a G-set. Finally we need to observe that the assignment $(T, \mathbf{x}) \mapsto (T, W(T, \mathbf{x}), M(T, \mathbf{x}))$ is G-equivariant and therefore comes from a G-functor according to Obs. 8.7. So we need to show that if $\sigma \in G$ then $W(T, \mathbf{x})^\sigma = W(T^\sigma, \mathbf{x}^\sigma)$ and $M(T, \mathbf{x})^\sigma = M(T^\sigma, \mathbf{x}^\sigma)$. This follows by noting that W and M are constructed iteratively in the course of the Local Certificates algorithm (reproduced here in Sec. 17), and it is clear that the objects constructed in each iteration are strongly canonical (satisfy the same G-equivariance condition). □

11 LARGE WINDOW

If there exists T such that $|W(T, \mathbf{x})| \ge n/10$ then we individualize such a T, i. e., we descend to G_T (at a multiplicative cost of $\binom{m}{t} = n^{O(\log n)}$) and proceed by Luks-recurrence: we use the Chain Rule on the ordered partition $(W(T, \mathbf{x}), \Omega \setminus W(T, \mathbf{x}))$. On $W(T, \mathbf{x})$ we can compute the automorphism group by efficient recursion via the Chain Rule, thanks to the Affected Orbit Lemma; the rest is at most 90%, a significant reduction.

We may therefore assume that for all test sets T we have $|W(T, \mathbf{x})| \le n/10$. Actually, all we shall use is that $W(T, x) \ne \Omega$.

12 STRONGLY CANONICAL RELATION

Our main technical contribution is the following result.

Theorem 12.1. *Assume none of the test sets is full. Then we can construct a strongly canonical* $(2t)$*-ary relation on* Γ *with symmetricity* $\le t - 1$*. The construction uses quasipolynomial time and makes a quasipolynomial number of calls to significantly smaller instances of the CP algorithm.*

Proof. This proof will run through Sections 12–16.

Given the input string $\mathbf{x}$, we define the $(2t)$-ary relation $R(\mathbf{x})$ on Γ as follows. We include the $(2t)$-tuple $(x_1, \dots, x_{2t}) \in \Gamma^{2t}$ in $R(\mathbf{x})$ if $x_1, \dots, x_t$ are all distinct, and, setting $T = \{x_1, \dots, x_t\}$, there exists $\tau \in M(T, \mathbf{x})$ such that $x_{i+t} = x_i^\tau$ for $i = 1, \dots, t$. The assignment $\mathbf{x} \mapsto R(\mathbf{x})$ is strongly canonical; this follows from Lemma 10.1.

For an ℓ-ary relation $P \subseteq \Gamma^\ell$, let P_T denote the restriction of P to T, i. e., $P_T = P \cap T^\ell$.

Next we consider the structure $(T, R(\mathbf{x})_T)$ – the substructure of $(\Gamma, R(\mathbf{x}))$ induced on T. The following key observation helps identify our difficulty.

Proposition 12.2. $\mathrm{Aut}(T, R(\mathbf{x})_T) = N_{\mathrm{Sym}(T)}(M(T, \mathbf{x}))$.

The right-hand side denotes the normalizer of $M(T, \mathbf{x})$ in $\mathrm{Sym}(T)$ (see the second paragraph of Sec. 3).

Proof. Let $\rho \in \mathrm{Sym}(T)$. Let $T = \{u_1, \dots, u_t\}$. Then $\rho \in \mathrm{Aut}(T, R(\mathbf{x})_T)$ if and only if $(\forall \tau \in M(T, \mathbf{x}))(\exists \xi \in M(T, \mathbf{x}))(\forall i)(u_i^{\tau\rho} = u_i^{\rho\xi})$. But this is equivalent to saying that $\rho^{-1}\tau\rho \in M(T, \mathbf{x})$. This must hold for every $\tau \in M(T, \mathbf{x})$, so $\rho^{-1} M(T, \mathbf{x}) \rho \le M(T, \mathbf{x})$ which is equivalent to $\rho^{-1} M(T, \mathbf{x}) \rho = M(T, \mathbf{x})$, i. e., $\rho \in N_{\mathrm{Sym}(T)}(M(T, \mathbf{x}))$. □

We say that the relation $P \subseteq \Gamma^\ell$ is *trivial on* T if $\mathrm{Aut}(T, P_T)$ is a giant on T.

Corollary 12.3. $R(\mathbf{x})$ *is trivial on* T *if and only if* $|M(T, \mathbf{x})| = 1$.

Proof. By Prop. 12.2, R is trivial on T if and only if $N_{\mathrm{Sym}(T)}(M(T, \mathbf{x}))$ is a giant on T, hence $M(T, \mathbf{x})$ is normal in $\mathrm{Sym}(T)$ or $\mathrm{Alt}(T)$. But $t \ge 5$, so $\mathrm{Alt}(T)$ is simple, therefore the only options for $M(T, \mathbf{x})$ are the identity or a giant. The latter is impossible by the definition of non-fullness certificates. □

We say that a test set T is *asymmetric* (with respect to $\mathbf{x}$) if $R(\mathbf{x})$ is trivial on T, i. e., $|M(T, \mathbf{x})| = 1$.

Asymmetric test sets cause considerable headache.

Let $\mathcal{F}(\mathbf{x})$ denote the set of asymmetric test sets.

Claim 12.4. *We can strongly canonically select an element* $u(T) \in T$ *from each* $T \in \mathcal{F}(\mathbf{x})$*, meaning that the assignment* $T \mapsto u(T)$ *will be strongly canonical:* $u(T^\sigma) = u(T)^\sigma$ *for all* $\sigma \in G$*. The complexity of the procedure is the same as the complexity statement in Theorem 12.1.*

Remark 12.5. In fact, we could even define a strongly canonical linear order on each $T \in \mathcal{F}(\mathbf{x})$, but selecting one element will suffice.

The existence of a strongly canonical element (or linear order) is obvious: we just pick an element (or a linear order) for one representative of each G-isomorphism class in $\mathcal{F}(\mathbf{x})$ and use G to translate it to all other members of the class. Because of the asymmetry of T, this will not lead to conflict. Making this selection efficient is our problem.

Given an element $u \in T$, let $R_u(T) \subseteq T^{2t}$ be the set of those $(2t)$-tuples that include u.

Once we have made a strongly canonical assignment of an element $u(T) \in T$ to each $T \in \mathcal{F}(\mathbf{x})$, we modify $R(\mathbf{x})$ by replacing $R(\mathbf{x})_T$ (which is trivial) by $R_{u(T)}(T)$ for all $T \in \mathcal{F}(\mathbf{x})$. Let us write $\widetilde{R}(\mathbf{x})$ for the updated relation $R(\mathbf{x})$. So $\mathbf{x} \mapsto \widetilde{R}(\mathbf{x})$ is a strongly canonical assignment of a $(2t)$-ary relation on Γ.

Claim 12.6. *The symmetricity of* $\widetilde{R}(\mathbf{x})$ *is at most* $t - 1$.

Proof. Let T be a test set. We need to show that $\mathrm{Aut}(\widetilde{R}(\mathbf{x}))_T^T \ne \mathrm{Sym}(T)$. To this end it suffices to show that $\mathrm{Aut}(\widetilde{R}(\mathbf{x})_T) \ne \mathrm{Sym}(T)$. This is equivalent to saying that $\widetilde{R}(\mathbf{x})_T$ is nontrivial. If T is not asymmetric, then, by definition, $R(\mathbf{x})_T$ is nontrivial and $\widetilde{R}(\mathbf{x})_T =$

$R(\mathbf{x})_T$. On the other hand, if T is asymmetric then $\widetilde{R}(\mathbf{x})_T = R_{u(T)}(T)$ and T has a special element with respect to this relation, namely, $u(T)$, which is fixed by all automorphisms of $R_{u(T)}(T)$. So again $\widetilde{R}(\mathbf{x})_T$ is nontrivial. □

This completes the proof of Theorem 12.1, modulo Claim 12.4. We spend the rest of this note trying to select a strongly canonical element from each asymmetric test set.

13 GROUP THEORY FROM [Ba15+]

We need the central group theoretic results from [Ba15+], appearing in the section titled "Alternating quotients of a permutation group."

13.1 Affected/Unaffected

For the definition of "affected," see Def. 3.5.

Theorem 13.1. *Let $G \le \mathrm{Sym}(\Omega)$ be a permutation group of degree n. Let $\varphi : G \to \mathrm{Sym}(\Gamma)$ be a giant action of G on a set Γ with $|\Gamma| = m$. Let $U \subseteq \Omega$ denote the set of elements of Ω not affected by φ. Then the following hold.*

(a) (Unaffected Stabilizers Lemma) *Assume $m > \max\{8, 2 + \log_2 n\}$. Then φ maps $G_{(U)}$, the pointwise stabilizer of U, onto a giant on Γ (so $\varphi : G_{(U)} \to \mathrm{Sym}(\Gamma)$ is still a giant action). In particular, $U \neq \Omega$ (at least one element is affected).*

(b) (Affected Orbit Lemma) *Assume $m \ge 5$. 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 $\le |\Delta|/m$.*

These results are central to the analysis of the *Local Certificates* algorithm. Part (a) is the main group theoretic result in [Ba15+]. The companion result, Part (b), is a simple observation, but it plays an important role by ensuring efficient recursion via the Chain Rule applied to the kernel of the giant homomorphism on affected orbits. We reference it multiple times; in particular, it is the basis of the "Recompute $H(W)$" routine, see Sec. 17.

13.2 The Jordan–Liebeck Sandwich Theorem

As in [Ba15+], we need the following classical result about the structure of subgroups of not too large index in the symmetric group. We cite it from the version given in [DiM96, Thm. 5.2A,B].

Theorem 13.2 (Jordan–Liebeck). *Let $\mathrm{Alt}(\Omega) \le K \le \mathrm{Sym}(\Omega)$. Let $H \le K$ and $1 \le r < n/2$ where $n = |\Omega| \ge 9$. Assume $|K : H| < \binom{n}{r}$. Then there exists a unique $S \subset \Omega$ with $|S| < n/2$ such that $\mathrm{Alt}(\Omega)_{(S)} \le H \le \mathrm{Sym}(\Omega)_S$. This unique S satisfies $|S| < r$.*

Let us denote this unique set S by $\mathrm{JL}(H)$.

We shall use the following corollary which appears as parts (a) and (b) of the "Main Stucture Theorem" in [Ba15+].

Corollary 13.3. *Let $G \le \mathrm{Sym}(\Omega)$ be a permutation group and $\varphi : G \to \mathrm{Sym}(\Gamma)$ a giant action. Assume $t = |\Gamma| \ge \max\{9, 2\log_2 n\}$.*

Then for every $x \in \Omega$ there exists a unique subset $S(x) \subset \Gamma$ such that $|S(x)| < t/4$ and

$$\mathrm{Alt}(\Gamma)_{(S(x))} \le \varphi(G_x) \le \mathrm{Sym}(\Gamma)_{S(x)}. \tag{18}$$

The element $x \in \Omega$ is affected by φ if and only if $|S(x)| \ge 1$.

Proof. Observe that

$$\binom{t}{\lfloor t/4 \rfloor} > 2^{t/2} \ge n \ge |x^G| = |G : G_x| \ge |\varphi(G) : \varphi(G_x)|. \tag{19}$$

Now the existence and uniqueness of the set $S(x)$ follows from the Jordan–Liebeck theorem, setting $K = \varphi(G)$, $H = \varphi(G_x)$, $S(x) = \mathrm{JL}(\varphi(G_x))$, $n = t$, and $r = \lfloor t/4 \rfloor$. The last sentence of Cor. 13.3 is immediate from Eq. (18) and the definition of being "affected." □

14 IDENTIFYING A TEST SET FROM THE SET OF POSITIONS

14.1 Superposition of Strings

Let $\mathbf{x}_i : \Omega \to \Sigma_i$ be strings for $i = 1, 2$. We define the superposition of these strings as the string $\mathbf{z} = \mathbf{x}_1 * \mathbf{x}_2$ where $\mathbf{z}(x) = (\mathbf{x}_1(x), \mathbf{x}_2(x))$, so $\mathbf{z} \in (\Sigma_1 \times \Sigma_2)^\Omega$. Clearly,

$$\mathrm{Iso}_G(\mathbf{x}_1 * \mathbf{x}_2, \mathbf{y}_1 * \mathbf{y}_2) = \mathrm{Iso}_G(\mathbf{x}_1, \mathbf{y}_1) \cap \mathrm{Iso}_G(\mathbf{x}_2, \mathbf{y}_2)\,. \tag{20}$$

14.2 Characteristic String

The *characteristic string* of a subset $\Delta \subseteq \Omega$ is the string $\chi_\Delta \in \{0,1\}^\Omega$ defined by $\chi_\Delta(x) = 1$ if $x \in \Delta$ and $\chi_\Delta(x) = 0$ otherwise.

14.3 Affected Elements Reveal Test Set

Notation 14.1 (Affected elements)**.** Let $G \le \mathrm{Sym}(\Omega)$ and let $\varphi : G \to \mathrm{Sym}(\Gamma)$ be a giant action on the ideal domain Γ. For $T \subseteq \Gamma$ of size $|T| = t$, let $\Delta(T)$ denote the set of elements of Ω affected by the epimorphism $\psi_T : G_T \to \mathrm{Sym}(T)$.

Proposition 14.2. *Assume $G \le \mathrm{Sym}(\Omega)$ is transitive and $\varphi : G \to \mathrm{Sym}(\Gamma)$ is a giant action on the ideal domain Γ. Let $t \ge \max\{9, 2\log_2 n\}$ and assume $m = |\Gamma| > 2t$. Assume $\Delta(T) \neq \Omega$ for some test set T (see Notation 14.1). Then for any test sets T_1, T_2 we have*

$$\mathrm{Iso}_G(\chi_{\Delta(T_1)}, \chi_{\Delta(T_2)}) = \{\sigma \in G \mid T_1^\sigma = T_2\}\,. \tag{21}$$

This statement seems surprisingly nontrivial; transitivity of G should not be necessary and $\Delta(T) \neq \Omega$ should be automatic.

Proof. Note that for $\sigma \in G$ we have $\Delta(T^\sigma) = (\Delta(T))^\sigma$. It follows by the transitivity of G that

$$\bigcup_{\sigma \in G} \Delta(T^\sigma) = \Omega\,. \tag{22}$$

We need to show that for test sets T_1, T_2 and $\sigma \in G$ we have $\Delta(T_1^\sigma) = \Delta(T_2)$ if and only if $T_1^\sigma = T_2$. The "if" part is clear.

To see the "only if" part, assume $\Delta(T_1)^\sigma = \Delta(T_2)$. Let $T_1^\sigma = T_3$. We need to show that $T_3 = T_2$. The equality $T_1^\sigma = T_3$ implies that $\Delta(T_1)^\sigma = \Delta(T_3)$ and therefore $\Delta(T_2) = \Delta(T_3)$. Now the relation "$\Delta(T_2) = \Delta(T_3)$" is a G-invariant equivalence relation on the set $\binom{\Gamma}{t}$. But the G-action on Γ is a giant and therefore the G-action on $\binom{\Gamma}{t}$ is primitive (it is a Johnson group, see Fact 3.4).

If the equivalence relation is discrete, the desired conclusion $T_2 = T_3$ follows. The only other option is that the set $\Delta(T) = \Delta$ does not depend on T. But by Eq. (22), this means $\Delta = \Omega$, contrary to our assumption. □

15 STRONGLY CANONICAL SELECTION OF A POINT FROM EACH ASYMMETRIC TEST SET

Using the Chain Rule we may assume G is transitive on the overall window referenced in the algorithm (Sec. 5.5). Moreover, we may ignore everything outside the window, so we may assume G is transitive on Ω.

Furthermore, by Sec. 11, we may assume $|W(T,\mathbf{x})| \le n/10$ for every test set T, where $n = |\Omega|$. Note that the set $\Delta(T)$ of points $x \in \Omega$ affected by G_T is a subset of $W(T,\mathbf{x})$ for every $\mathbf{x}$,

$$\Delta(T) \subseteq W(T,\mathbf{x}), \tag{23}$$

since $\Delta(T)$ is the window of the first round of the Local Certificates algorithm. In particular, we have $\Delta(T) \neq \Omega$.

Let $W^- = W^-(T,\mathbf{x})$ denote the penultimate window associated with the test set T (the window obtained in the next to last execution of the **while** loop in the Local Certificates algorithm). Let $A^- = A^-(T,\mathbf{x}) := \mathrm{Aut}_{G_T}(\mathbf{x}^{W^-})$. The homomorphism ψ_T maps A^- onto a giant on T. Applying Cor. 13.3 to this giant action, we associate each $x \in W^-$ with a unique subset $S(x) \subset T$ such that $|S(x)| < t/4$ and

$$(\mathrm{Alt}(T))_{(S(x))} \le \psi_T(A^-_x) \le (\mathrm{Sym}(T))_{S(x)}\,. \tag{24}$$

The sets $S(x)$ are not empty for $x \in W^-$ because by construction, x is affected by the restriction of ψ_T to A^-. (The fact that affected points exist follows by part (a) of Theorem 13.1.) Moreover, for each A^--orbit $\Phi \subseteq W^-$, the sets $S(x)$ $(x \in \Phi)$ uniformly cover T. Now for $u \in T$ let

$$\Omega(T,u) = \{x \in W^- \mid u \in S(x)\}. \tag{25}$$

Lemma 15.1. *Let Φ be an A^--orbit in W^-. Let $u, v \in T$. If $(\forall x \in \Phi)(u \in S(x) \leftrightarrow v \in S(x))$ then $u = v$.*

Proof. Assume $u \neq v$. For $t \ge 4$ the group $\mathrm{Alt}(T)$ is doubly transivite. Therefore, if $S \subset T$ is a nonempty proper subset of T then there exist $\tau_1, \tau_2 \in \mathrm{Alt}(T)$ such that $u \in S^{\tau_1}$ and $u \notin S^{\tau_1}$. Noting that $\psi_T(A^-) \ge \mathrm{Alt}(T)$ we can lift this statement to $A-$ to conclude that for any $x \in \Phi \subseteq W^-$ there exist $\sigma_1, \sigma_2 \in A^-$ such that $u \in S(x)^{\sigma_1} = S(x^{\sigma_1})$ and $v \notin S(x)^{\sigma_2} = S(x^{\sigma_2})$, a contradiction. Here we relied on the fact that $S(x)$ is not empty (see above) and $S(x) \neq T$ because $|S(x)| < t/4$). □

Continuing our proof toward the title of this section, for $u \in T$ let $\mathbf{x}(T,u)$ denote the string superposition $\mathbf{x}^{W(T,\mathbf{x})} * \chi_{\Delta(T)} * \chi_{\Omega(T,u)}$ (see Def. 14.1). (We just quadrupled the size of the alphabet.) It is important to note that both $\Delta(T)$ and $\Omega(T,u)$ are subsets of the window $W(T,\mathbf{x})$. Indeed, $\Omega(T,u) \subseteq W^-$ by definition; for $\Delta(T)$, see the explanation of Eq. (23). As a consequence, in $\mathbf{x}(T,u)$ (just as in $\mathbf{x}^{W(T,\mathbf{x})}$), all positions in $\Omega \setminus W(T,\mathbf{x})$ are filled with the dummy symbol β. This will be crucial for the proof of the next statement.

Claim 15.2. *We can compute, by efficient recurrence via the Chain Rule, a G-canonical form F^* of the strings $\mathbf{x}(T,u)$ for all $T \in \mathcal{F}(\mathbf{x})$ and $u \in T$.*

Again, the efficiency of the application of the Chain Rule depends on the Affected Orbit Lemma.

Before proving Claim 15.2, we show how it leads to the proof of Claim 12.4 (strongly canonical selection of an element from each asymmetric test set), completing the proof of Theorem 12.1.

Let $\mathbf{z}(T,u) = F^*(\mathbf{x}(T,u))$.

Proposition 15.3. *Let $u, v \in T$. If $\mathbf{z}(T,u) = \mathbf{z}(T,v)$ then $u = v$.*

This statement is not trivial; it relies on Prop. 14.2. In particular, this is where we use that G is transitive on Ω and that $\Delta(T) \neq \Omega$, as declared at the beginning of this section.

Proof. If $\mathbf{z}(T,u) = \mathbf{z}(T,v)$ then by definition $\mathbf{x}(T,u) \cong_G \mathbf{x}(T,v)$. Let $\sigma \in G$ be such a G-isomorphism, so $\mathbf{x}(T,u)^\sigma = \mathbf{x}(T,v)$. But then $\sigma \in \mathrm{Aut}_G(\chi_{\Delta(T)})$ and therefore, by Prop. 14.2, $T^\sigma = T$ (using the assumptions stated before this proof), so $\sigma \in G_T$. Since both $\mathbf{x}(T,u)$ and $\mathbf{x}(T,v)$ are refinements of $\mathbf{x}^{W(T,\mathbf{x})}$, we have that $\sigma \in \mathrm{Aut}_{G_T}(\mathbf{x}^{W(T,\mathbf{x})})$. Since T is asymmetric, it follows that $\psi_T(\sigma) = 1$ (the identity permutation of T). In particular, $u^\sigma = u$, so $\mathbf{x}(T,u) = \mathbf{x}(T,v)$, therefore $\Omega(T,u) = \Omega(T,v)$ from which we conclude by Lemma 15.1 that $u = v$. □

Let us now consider the set $Z(T) := \{\mathbf{z}(T,v) \mid v \in T\}$ of canonical forms. They are pairwise not equal by Prop. 15.3. So select $u = u(T)$ to correspond to the lexicographic leader in $Z(T)$. The strong canonicity of this choice is clear. This completes the proof of the statement in the title of this section modulo Claim 15.2.

16 CANONICAL PLACEMENT OF THE STRINGS $\mathbf{x}(T,u)$

Finally, we prove Claim 15.2.

To do so, we need to delve into the details of the Local Certificates algorithm which we reproduce in the Appendix.

The procedure builds a strictly increasing chain of windows, $\emptyset = W_0 \subset W_1 \subset \cdots \subset W_k \subseteq \Omega$ where $W_i = W_i(T,\mathbf{x})$ and $k = k(T,\mathbf{x})$. Simultaneously, we also build the sequence of groups $H_i = H_i(T,\mathbf{x}) := \mathrm{Aut}^{W_i}_{G_T}(\mathbf{x})$, starting with $H_0 = G_T$. Each W_i is invariant under H_{i-1}.

Let K_i be the kernel of the $H_i \to \mathrm{Sym}(T)$ action (restriction of ψ_T to H_i). This is a giant action on T for $i = 0, \ldots, k-1$, and W_{i+1} is defined as the set of elements of Ω affected by this action. In particular, $W_1 = \Delta(T)$, $W_{k-1} = W^-$, and $W_k = W(T,\mathbf{x})$.

By the Affected Orbit Lemma (part (b) of Theorem 13.1), each orbit of H_i in W_{i+1} breaks into at least t orbits of equal length under K_i, permitting an efficient combination of descent to K_i (at a multiplicative cost $\le t! < n^{O(\log\log n)}$) and the Chain Rule to compute H_{i+1} (and K_{i+1}). This is the Recompute $H(W)$ routine (see Sec. 17). The iterative computation of the H_i can be directly adapted to iteratively computing $\mathrm{CP}_{G_T}(\mathbf{x}^{W_i(T,\mathbf{x})})$ using the CP versions of Descent and the Chain Rule, described in Sec. 5.

Next we wish to compute, for each $u \in T$, a canonical placement for $\mathbf{x}(T,u)^{W(T,\mathbf{x})}$ with respect to G_T. We do this by applying the Chain Rule to the ordered partition $W(T,\mathbf{x}) = \bigsqcup_{i=1}^{k}(W_i \setminus W_{i-1})$. Let $H_i(u) = \mathrm{Aut}^{W_i}_{G_T}(\mathbf{x}(T,u))$ and $K_i(u) = K_i \cap H_{i-1}(u)$. Note that $H_i(u) \le H_i$ (since $\mathbf{x}(T,u)$ is a refinement of $\mathbf{x}$). Therefore W_{i+1} is invariant under $H_i(u)$, so the Chain Rule applies (cf. Rem. 5.5). We note further that $K_i(u)$ is the kernel of the restriction of ψ_T to $H_i(u)$ and therefore can be computed in polynomial time given H_{i-1} and ψ_T.

Let Π_i denote the partition of W_{i+1} into K_i-orbits. To compute the canonical placement of $\mathbf{x}(T,u)^{W_{i+1}}$ during the i-th round, we

descend to $K_i(u)$ and apply the Chain Rule to the partition Π_i. Since $K_i(u) \le K_i$, the orbit partition for $K_i(u)$ is a refinement of the orbit partition for K_i, so the blocks of Π_i are $K_i(u)$-invariant.

At the end of this process we have found a canonical placement with respect to G_T for each $\mathbf{x}(T,u)^{W(T,\mathbf{x})}$ $(u \in T)$.

Now we need canonicity with respect to G rather than G_T. We achieve this by descending from G to G_T (multiplicative cost $\binom{m}{t} < n^{O(\log n)}$). We need to be careful, though: calling the descent discussed in Sec. 5.4 *left descent* (decomposition into left shifts of a subcoset), here we need *right descent*: $G = \bigsqcup_{\sigma\in R} G_T\sigma$. We see the difference by comparing item (i) of Def. 5.1 and Eq. (11) in Obs. 5.4. Left shifts would require the recursive evaluation of expressions of the form $\mathrm{CP}_{G_T}(\mathbf{x}^\sigma)$, but $\mathrm{Aut}_{G_T}(\mathbf{x}^\sigma)$ may be very different from $\mathrm{Aut}_{G_T}(\mathbf{x})$ (it is even possible that T is not full with respect to $\mathbf{x}$ but full with respect to $\mathbf{x}^\sigma$), whereas right shifts will require the recursive evaluation of expressions of the form $\mathrm{CP}_{(G_T)^\sigma}(\mathbf{x}^\sigma)$. Now, noting that $(G_T)^\sigma = G_{T^\sigma}$ (Fact 3.2), we see that $\mathrm{Aut}_{(G_T)^\sigma}(\mathbf{x}^\sigma) = (\mathrm{Aut}_{G_T}(\mathbf{x}))^\sigma$. In particular, T is asymmetric with respect to $\mathbf{x}$ if and only if T^σ is asymmetric with respect to $\mathbf{x}^\sigma$.

This step completes the proof of Claim 15.2 and with it, Theorem 12.1. □

17 APPENDIX: THE LOCAL CERTIFICATES ALGORITHM

This is the core algorithm of the SI test [Ba15+]; for easier reference, we reproduce it here. For a detailed explanation we refer to the "Local Certificates" section of [Ba15+].

Procedure LocalCertificates

Input: $G \le \mathrm{Sym}(\Omega)$, epimorphism $\psi_T : G_T \to \mathrm{Sym}(\Gamma)$, test set $T \in \binom{\Gamma}{t}$
Output: decision: "T full/not full," group $K(T) \le \mathrm{Sym}(\Omega)$ (if full) or $M(T) \le \mathrm{Sym}(T)$ (if not full), set $W(T) \subseteq \Omega$
Notation: $H(W) := \mathrm{Aut}^W_{G_T}(\mathbf{x})$ (to be updated as W is updated)

01 $W := \emptyset$ (: so $H(W) = G_T$:)
02 **while** $H(W)^T \ge \mathrm{Alt}(T)$ and $\mathrm{aff}(H(W), \psi_T) \not\subseteq W$
03 $W \leftarrow \mathrm{aff}(H(W), \psi_T)$ (: enlarging the window :)
04 recompute $H(W)$
05 **end(while)**
06 $W(T) \leftarrow W$
07 **if** $H(W)^T \ge \mathrm{Alt}(T)$ (: so $\mathrm{aff}(H(W), \psi_T) \subseteq W$:)
08 **then** $K(T) \leftarrow H(W)_{(\overline{W})}$ where $\overline{W} = \Omega \setminus W$
09 **return** $W(T)$, $K(T)$, "T full," **exit** (: fullness certificate :)
10 **else** $M(T) \leftarrow H(W)^T$
11 **return** $W(T)$, $M(T)$, "T not full," **exit**
(: non-fullness certificate :)

Note that on line 08, we take the pointwise stabilizer of the complement of the window $W(T)$ in the *W-local* automorphism group $H(W)$ (consisting of automorphisms of the partial string $\mathbf{x}^W$). This is the key local-to-global step of the algorithm; the stabilizer $K(T)$ consists of *global* automorphisms (automorphisms of the full string $\mathbf{x}$), and the Unaffected Stabilizers Lemma (part (a) of Theorem 13.1) guarantees that there are plenty of them.

We need to show how to recompute $H(W)$ on line 04. We write W_{old} for the value of W before the execution of line 03 and W_{new} after. Recall Def. 3.3 for the restriction notation G^T.

Procedure Recompute $H(W)$

04a $N \leftarrow H(W_{\mathrm{old}})^T_{(T)}$ (: kernel of $H(W_{\mathrm{old}}) \to \mathrm{Sym}(T)$ map :)
04b $L \leftarrow \emptyset$ (: L will collect elements of $H(W_{\mathrm{new}})$:)
04c **for** $\overline{\sigma} \in H(W_{\mathrm{old}})^T$ (: $H(W_{\mathrm{old}})^T = \mathrm{Alt}(T)$ or $\mathrm{Sym}(T)$:)
04d select $\sigma \in H(W_{\mathrm{old}})$ such that $\sigma^T = \overline{\sigma}$ (: lifting $\overline{\sigma}$ to Ω :)
04e $L(\overline{\sigma}) \leftarrow \mathrm{Aut}^{W_{\mathrm{new}}}_{N\sigma}(\mathbf{x})$ (: descent to N :)
04f $L \leftarrow L \cup L(\overline{\sigma})$
04g **end(for)**
04h **return** $H(W_{\mathrm{new}}) \leftarrow L$

We note that the efficiency of the recursive call on line 04e is based on the fact that the affected orbits of $H(W)$ split into much shorter orbits of N as a consequence of the "Affected Orbit Lemma" (part (b) of Theorem 13.1.

ACKNOWLEDGMENTS

I'd like to express my gratitude to Gene Luks for decades of friendship and collaboration, including recent discussions of the concept of canonical forms. I wish to thank my former student John Wilmes, who, in his capacity of chair of a session at the "Symmetry vs. Regularity" conference (Plzeň, Czech Republic, July 2018), reminded me to focus on the main issue addressed in this paper. Last but not least, I'd like to commend the STOC PC for the impressive feat of collecting as many as seven (!) substantial peer reviews for this paper, and express my thanks to the anonymous reviewers whose authoritative reviews and detailed comments greatly helped improve the presentation.

REFERENCES

[Ba79] László Babai: Monte Carlo algorithms in graph isomorphism testing. Université de Montréal Tech. Rep. DMS 79-10, 1979 (pp. 42). Accessible at http://people.cs.uchicago.edu/~laci/lasvegas79.pdf
[Ba15+] László Babai: Graph isomorphism in quasipolynomial time. arXiv:1512.03547, 2015–2019.
[BaCP82] 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.
[BaKL] László Babai, Paul Klingsberg, Eugene M. Luks: Canonical labelling for vertex coloured graphs. Unpublished, 1980.
[BaL83] László Babai and Eugene M. Luks: Canonical labeling of graphs. In: *Proc. 15th STOC*, ACM 1983, pp. 171–183.
[Cam81] Peter J. Cameron: Finite permutation groups and finite simple groups, *Bull. London Math Soc.* **13** (1981) 1–22.
[DiM96] John D. Dixon, Brian Mortimer: *Permutation Groups*. Springer Grad. Texts in Math. vol. 163, 1996
[FG11] Lance Fortnow and Joshua A. Grochow: Complexity classes of equivalence problems revisited. *Information and Computation* **209** (2011) 748–763.
[FHL80] Merrick Furst, John Hopcroft, Eugene Luks: Polynomial-time algorithms for permutation groups. *In: Proc. 21st IEEE FOCS*, 1980, pp. 36–41.
[FSS83] Martin Fürer, Walter Schnyder, and Ernst Specker: Normal forms for trivalent graphs and graphs of bounded valence. In: *Proc. 15th STOC*, ACM 1983, pp. 161-âĂŞ170.
[Lu82] Eugene M. Luks: Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comput. Syst. Sci.* **25(1)** (1982) 42-âĂŞ65.
[Se03] Ákos Seress: *Permutation Group Algorithms*. Cambridge Univ. Press, 2003