Code Equivalence and Group Isomorphism

László Babai*, Paolo Codenotti, Joshua A. Grochow
{laci, paoloc, joshuag}@cs.uchicago.edu (University of Chicago)
and
Youming Qiao†
jimmyqiao86@gmail.com (Tsinghua University)

Abstract

The isomorphism problem for groups given by their multiplication tables has long been known to be solvable in time $n^{\log n+O(1)}$. The decades-old quest for a polynomial-time algorithm has focused on the very difficult case of class-2 nilpotent groups (groups whose quotient by their center is abelian), with little success. In this paper we consider the opposite end of the spectrum and initiate a more hopeful program to find a polynomial-time algorithm for *semisimple groups*, defined as groups without abelian normal subgroups. First we prove that the isomorphism problem for this class can be solved in time $n^{O(\log\log n)}$. We then identify certain bottlenecks to polynomial-time solvability and give a polynomial-time solution to a rich subclass, namely the semisimple groups where each minimal normal subgroup has a bounded number of simple factors. We relate the results to the filtration of groups introduced by Babai and Beals (1999).

One of our tools is an algorithm for equivalence of (not necessarily linear) codes in simply-exponential time in the length of the code, obtained by modifying Luks's algorithm for hypergraph isomorphism in simply-exponential time in the number of vertices (FOCS 1999).

We comment on the complexity of the closely related problem of permutational isomorphism of permutation groups.

1 Introduction

1.1 Group isomorphism - bottlenecks and approach. The isomorphism problem for groups asks to determine if two groups, given by their Cayley tables (multiplication tables), are isomorphic. Tarjan is credited for pointing out that if one of the groups is generated by k elements then isomorphism can be decided in time $n^{k+O(1)}$ where n is the order of the groups; indeed one can list all isomorphisms within this time bound (cf. [27]). Since $k \le \log n$ for all groups, this in particular gives an $n^{\log n+O(1)}$-time algorithm for all groups (log to the base 2) and a polynomial-time algorithm for finite simple groups (because the latter are generated by 2 elements, a consequence of their classification [14]).

In spite of considerable attention to the problem over the past quarter century, no general bound with a sublogarithmic exponent has been obtained.

While the abelian case is easy ($O(n)$ according to Kavitha [19], improving Savage's $O(n^2)$ [30] and Vikas's $O(n\log n)$ [34]), just one step away from the abelian case lurk what appear to be the most notorious cases: nilpotent groups of class 2. These groups G are defined by the property that the quotient $G/Z(G)$ is abelian, where $Z(G)$ is the center of G. No complete structure theory of such groups is known; recent work in this direction by James Wilson [35, 36] commands attention.

Recently, other special classes of solvable groups have been considered; the isomorphism problem of extensions of an abelian group by a cyclic group of relatively prime order has been solved very efficiently (sublinear time in the black-box model) [22]. We note that the structure of such groups is well understood.

While class-2 nilpotent groups have long been recognized as the chief bottleneck in the group isomorphism problem, this intuition has never been formalized. The ultimate formalization would reduce the general case to this case. As a first step, we consider a significant class without a chance of a complete structure theory at the opposite end of the spectrum: groups without abelian normal subgroups. Following [29], we call such groups *semisimple*[1]. Our project is to show that semisimple groups admit a polynomial-time isomorphism test.

*László Babai's work was supported in part by NSF Grant CCF-0830370.

†Youming Qiao's work was supported in part by the National Natural Science Foundation of China Grant No.60553001, and the National Basic Research Program of China Grant Nos.2007CB807900, 2007CB807901.

[1] We note that authors use the term 'semisimple group' in several different meanings (see e. g. [33]).

1.2 A general result. The *solvable radical* $\mathrm{Rad}(G)$ of a group G is the unique maximal solvable normal subgroup of G. A group G is *semisimple* if and only if $\mathrm{Rad}(G) = 1$. For every group G, the quotient $G/\mathrm{Rad}(G)$ is semisimple. This fact indicates the richness of the class of semisimple groups.

Our first result, to be proved in Section 4 (see Corollary 4.2), concerns the entire class.

THEOREM 1.1. *Isomorphism of two semisimple groups of order n can be decided in time $n^{O(1)+c\log\log n}$, where $c = 1/\log(60) \approx 0.16929$. In fact, all isomorphisms can be listed within this time bound.*

REMARK. Because the algorithm above lists all the isomorphisms, we cannot hope to get a better bound on the running time for pairs of groups with that many isomorphisms. Such groups do indeed exist. For example, consider the group $G = A_5^k$, the direct product of k copies of the alternating group of order 60. The group A_5^k is semisimple and has $120^k k! > n^{c\log\log n}$ automorphisms ($n = |G| = 60^k$), where $c = 1/\log(60)$.

Recall that the trivial algorithm to check isomorphism takes time $n^{O(1)+k}$, where k is the number of generators of our groups[2]. We point out that Theorem 1.1 is not a special case of the $n^{O(1)+k}$ bound.

FACT 1.1. *There exist semisimple groups which require at least $\log_{120} n$ generators.*

For example, S_5^k is semisimple (where S_5 is the symmetric group of degree 5 and order 120), but every set of generators of S_5^k has size at least k, since S_5^k has a quotient isomorphic to $\mathbb{Z}_2^k$. Here $k = \log_{120}(n)$.

1.3 The main result. We now deal with cases when it is not possible to list all the isomorphisms within the desired time bound. The set of isomorphisms of two groups G and H is either empty or a coset $\mathrm{Aut}(G)\sigma$ of $\mathrm{Aut}(G)$, which we will represent by a list of generators of the automorphism group of G and a particular isomorphism $\sigma : G \to H$.

Every minimal normal subgroup is characteristically simple, and hence it is the direct product of isomorphic simple groups. (See Section 2.5 for definitions.)

We parametrize our groups G by a parameter $t(G)$ and solve the case of bounded $t(G)$ in polynomial time, and the general case in time $n^{O(\log(t(G)+1))}$. We define $t(G)$ as the smallest t such that each minimal normal subgroup of G has at most t simple factors. Our main result is the following.

[2]Throughout this paper, n denotes the order of the groups to be tested for isomorphism.

THEOREM 1.2. *Isomorphism of semisimple groups G and H of order n can be decided, and the coset of isomorphisms found, in time $n^{O(1)+c\log(t(G)))}$, where $c = 6/\log(60) \approx 1.0158$.*

We prove this result in Section 6 (see Corollary 6.2).

Note that $t(G) \le \log_{60} n$, and hence this result subsumes Theorem 1.1 (but the algorithm of Theorem 1.1 is much simpler).

Every semisimple group is an extension of a group G with $t(G) = 1$ by a permutation group of logarithmic degree (Fact 7.3). Therefore a key ingredient of the yet unsolved part of the project will be to decide *permutational isomorphism* of permutation groups of degree k in time polynomial in 2^k and the order of the groups. That doing so is indeed necessary is shown in Prop. 7.1. While we cannot claim that it is also sufficient (cf. Appendix Section 7.6), we believe that a solution of the stated complexity for the permutational isomorphism problem, combined with the methods of the present paper, will get us close to a polynomial-time solution of group isomorphism for all semisimple groups. We solve the case of bounded orbits in the required time (see Theorem 7.2). We note that this case includes equivalence of linear codes over prime fields of bounded order (see Proposition 7.2)

1.4 Codes. We reduce the isomorphism problem for semisimple groups to equivalence of group codes. We consider the code equivalence problem as a separate problem of interest in its own right. A *code* of length n over a finite alphabet Γ is a subset of Γ^A for some set A with $|A| = n$. An *equivalence* of the codes $\mathcal{A} \subseteq \Gamma^A$ and $\mathcal{B} \subseteq \Gamma^B$ is a bijection $A \to B$ that takes $\mathcal{A}$ to $\mathcal{B}$. If $|\Gamma| = 2$ then the code is a Boolean function or hypergraph, so the code equivalence problem is a generalization of the hypergraph isomorphism problem. Modifying and extending Luks's C^n dynamic programming algorithm for hypergraph isomorphism [26] to treat code equivalence, we obtain the following result, proved in Section 5.2.

THEOREM 1.3. *The set of equivalences of two codes of length n over an alphabet of size k can be found in time $(ck)^{2n}$, for some absolute constant c.*

As before, the set of equivalences is a coset, given by generators and a coset representative.

We remark that our algorithm, while inspired by Luks's, is different from his even in the special case of hypergraph isomorphism. We obtain some simplification by eliminating a divide-and-conquer aspect of Luks's algorithm; the cost is somewhat lesser efficiency.

Now let Γ be a group.

DEFINITION 1.1. *A* Γ*-code of length* n *(or a group-code of length* n *over* Γ*) is a subgroup of* Γ^n*.*

We shall apply Theorem 1.3 to group codes. To be more precise, we need to extend the concept, and Theorem 1.3, to multiple alphabets, where the alphabet used depends on the position (see Theorem 5.1).

One would hope for a more efficient algorithm for group codes that does not ignore the group structure. The first author [6] found such a faster algorithm for linear codes (see Appendix, Section 7.1).

1.5 Strategy for the main result. The *socle* of a group is defined as the product of its minimal normal subgroups. The socle of a semisimple group is the direct product of nonabelian simple groups.

First we observe that isomorphism of groups that are direct products of simple groups can be tested in polynomial time (Proposition 2.1). So we can assume that our semisimple groups G and H have isomorphic socles which decompose "isomorphically" into the direct product of minimal normal subgroups.

We find a "small" canonical class of labellings of each minimal normal subgroup. Once such a labelling is fixed for each minimal normal subgroup, the problem is reduced to group code isomorphism where the alphabets are the automorphism groups of the minimal normal subgroups.

1.6 Organization. The remainder of the paper is organized as follows. Section 2 introduces group-theoretic terminology and constructions. In Section 3 we preove the key lemma that is the basis for our algorithms. In Section 4 we prove Theorem 1.1. Our algorithms for code equivalence are presented in Section 5. in Section 6 we present our main algorithm. In the Appendix, Section 7.1 we present the algorithm for linear code equivalence. In Section 7.2 we reduce problem of deciding the permutational isomorphism of permutation groups within the stated time bounds to solving semisimple group isomorphism in polynomial time, formally showing the necessity of the former. In Sections 7.4 and 7.3 we discuss the complexity of permutational isomorphism of permutation groups. In Sections 7.5 and 7.6 we relate our results to the Babai-Beals filtration. Finally, we collect open questions in Section 7.7.

1.7 Conventions. We list some conventions that we use throughout the paper. Unless indicated otherwise,

- log is to the base 2;
- groups are finite;
- n is the order of the group G;
- 'simple groups' are non-abelian.

2 Group Theoretic Preliminaries

2.1 Permutation groups. $\mathrm{Sym}(A)$ denotes the symmetric group acting on the set A, i. e., the group of all permutations of A. S_n denotes $\mathrm{Sym}([n])$ where $[n] = \{1, \ldots, n\}$. Permutation groups acting on the permutation domain A are subgroups $G \leq \mathrm{Sym}(A)$. If $|A| = n$ then G is a permutation group of *degree* n. For $a \in A$ and $\pi \in G$ we use a^π to denote the image of a under π. The *orbit* of $a \in A$ is the set $a^G := \{a^\pi : \pi \in G\}$. The orbits partition the permutation domain. The *length* of an orbit is its size.

A coset of G is $G\pi = \{g\pi \colon g \in G\}$ for some $\pi \in \mathrm{Sym}(A)$. The intersection of cosets Gg and Hh ($G, H \leq \mathrm{Sym}(A)$, $g, h \in \mathrm{Sym}(A)$) is either empty or a coset of $G \cap H$. A coset Gg is given by a coset representative $g' \in Gg$ and a list of generators of the group G.

Given two finite sets A and B, an element $\sigma \in \mathrm{Sym}(A)$, and a bijection $\pi \colon A \to B$, we define $\sigma^\pi \in \mathrm{Sym}(B)$ by $\sigma^\pi = \pi^{-1}\sigma\pi$. Given a set of elements $\Sigma \subseteq \mathrm{Sym}(A)$, we define $\Sigma^\pi = \{\sigma^\pi : \sigma \in \Sigma\}$.

If $K \leq \mathrm{Sym}(A)$ and $L \leq \mathrm{Sym}(B)$ are permutation groups, then a bijection $\pi \colon A \to B$ is a *permutational isomorphism* $K \to L$ if $K^\pi = L$. We denote the set of all $K \to L$ permutational isomorphisms by $\mathrm{PISO}(K, L)$, and we say K and L are *permutationally isomorphic* if $\mathrm{PISO}(K, L) \neq \emptyset$. Note that $\mathrm{PISO}(K, K) = N_{\mathrm{Sym}(A)}(K)$, the normalizer of K in $\mathrm{Sym}(A)$. We shall comment on the complexity of determining $\mathrm{PISO}(K, L)$ in Sections 7.3 and 7.4.

2.2 Algorithms for permutation groups For permutation groups given by a list of generators, the basic tasks of membership testing, computing the order, finding the normal closure can be done in polynomial time [32, 15, 21], cf. [31] and in fact in NC [12]. Many more advanced tasks, such as finding a composition series can also be done in polynomial time [25] and even in NC [12].

A particularly important problem for permutation groups is the *Coset Intersection problem:* given two cosets of subgroups of $\mathrm{Sym}(A)$, find their intersection. Graph Isomorphism can be Karp-reduced to Coset Intersection [24]. The Coset Intersection problem for permutation groups of degree n can be solved in $\exp(\widetilde{O}(\sqrt{n}))$ time [2] (see also [7, 10]).

2.3 Abstract groups. In algorithms, a group G is specified by its multiplication table, consisting of $|G|^2$ group entries.

Given two groups G and H, a bijection $\pi \colon G \to H$ is a *group isomorphism* if it is a homomorphism, i. e.,

$(g_1g_2)^\pi = g_1^\pi g_2^\pi$ for all $g_1, g_2 \in G$. An isomorphism $G \to G$ is an *automorphism*. We denote the group of automorphisms by $\mathrm{Aut}(G)$ and the set of $G \to H$ isomorphisms by $\mathrm{ISO}(G, H)$. We say G and H are isomorphic if $\mathrm{ISO}(G, H)$ is not empty. As in the case of code equivalence $\mathrm{ISO}(G, H)$ is either empty or a coset of $\mathrm{Sym}(G \cup H)$.

An *embedding* is an injective group homomorphism. The notation $\varphi\colon G \hookrightarrow H$ means that φ is an embedding of G into H.

If $N \trianglelefteq G$ is a normal subgroup of G, then G acts on N by conjugation. This action defines a homomorphism $\gamma = \gamma_{G,N}\colon G \to \mathrm{Aut}(N)$. For $g \in G$ and $n \in N$, we write $n^g = n^{\gamma(g)} = g^{-1}ng$. G is said to *act faithfully* on N if γ is injective. Note that $\ker(\gamma) = C_G(N)$, the *centralizer* of N in G. If $C_G(N) = 1$, then γ is an embedding. If we take $N = G$, then an automorphism of the form $\gamma(g)$ is called *inner*. The group of inner automorphisms is denoted $\mathrm{Inn}(G)$; it is a normal subgroup of $\mathrm{Aut}(G)$, and $\mathrm{Inn}(G) \cong G/Z(G)$ where $Z(G)$ denotes the center of G. If $Z(G) = 1$ then $\gamma_G = \gamma_{G,G}$ is a canonical isomorphism $G \cong \mathrm{Inn}(G)$.

LEMMA 2.1. *Let G be a group, and $N \trianglelefteq G$ a normal subgroup with trivial centralizer. Then every isomorphism $\varphi\colon N \to M$ extends uniquely to an embedding $\Phi\colon G \hookrightarrow \mathrm{Aut}(M)$ with $\Phi|_N = \varphi\gamma_M$. In particular, there is a bijection between $\mathrm{ISO}(N, M)$ and the set of embeddings $\Phi\colon G \hookrightarrow \mathrm{Aut}(M)$ such that $\Phi(N) = \mathrm{Inn}(M)$.*

2.4 Direct, semidirect, subdirect, and wreath products. Given groups $G_1, \ldots, G_r$, we write $\prod_{i=1}^r G_i$ for the direct (Cartesian) product $G_1 \times \cdots \times G_r$. We write $\pi_j\colon \prod_{i=1}^r G_i \to G_j$ for the projection map onto the j-th factor. A *subdirect product* of $G_1, \ldots, G_r$ is a subgroup $H \leq \prod_{i=1}^r G_i$ such that $\pi_j(H) = G_j$ for each j.

Given a group K with an action on another group H given by $\theta\colon K \to \mathrm{Aut}(H)$, the *semidirect product* $H \rtimes_\theta K$ is a group with underlying set $H \times K = \{(h, k)\colon h \in H, k \in K\}$ and multiplication defined by: $(h_1, k_1)(h_2, k_2) = (h_1 h_2^{\theta(k_1^{-1})}, k_1k_2)$. When the action θ is understood, we write simply $H \rtimes K$.

If $\theta\colon K \to \mathrm{Sym}(A)$ is a permutation action of K on the set A, we define the *wreath product* $H \wr_\theta K$ as $H^A \rtimes_{\overline{\theta}} K$, where $\overline{\theta}\colon K \to \mathrm{Aut}(H^A)$ is the action of K on H^A by permuting the factors. That is, $(h_1, \ldots, h_n)^{\overline{\theta}(k)} = (h_{1^{\theta(k)}}, \ldots, h_{n^{\theta(k)}})$, where we have assumed $A = [n]$. If $K \leq \mathrm{Sym}(A)$ is a permutation group, we write simply $H \wr K = H^A \rtimes K$.

2.5 Characteristically simple groups; the socle.

DEFINITION 2.1. Let $H \leq G$ be a subgroup. H is a *characteristic subgroup* if H is invariant under all automorphisms of G. A group is *characteristically simple* if it has no nontrivial characteristic subgroups.

FACT 2.1. Every characteristically simple group is the direct product of isomorphic (abelian or non-abelian) simple groups.

PROPOSITION 2.1. *Let G be a direct product of simple groups.*
(a) The (unique) direct product decomposition of G into its simple factors can be found in polynomial time.
(b) Isomorphism of G and any other group H can be decided, and the set of isomorphisms found, in polynomial time.

Proof. (a) Take the normal closure of each element; take the minimal ones among the subgroups obtained. These are the direct factors. (b) Do the same to H and verify that a direct decomposition was found; if not, reject isomorphism. Otherwise decide which pairs among the simple factors of G and H are isomorphic and find all their isomorphisms. If the multiplicities of isomorphism types don't match, reject isomorphism. Otherwise, find an isomorphism along matched factors of G and H; combine this with the automorphim group of G. The automorphism group of $G = \prod T_i^{k_i}$ is $\prod \mathrm{Aut}(T_i) \wr S_{k_i}$.

We used the fact that all isomorphisms of two simple groups (and therefore all automorphisms of a simple group) can be listed in polynomial time. The reason, as pointed out in the Introduction, is that simple groups can be generated by 2 elements. □

While the proof above is straightforward, we mention that the same result holds, nontrivially, in the context of permutation groups; in fact, if a permutation group is a product of simple groups, it can be split into its simple factors in NC [12]. We mention that direct decomposition of a permutation group is also known to be computable in polynomial time even if the direct factors are not simple; this was done in [20] for groups given by Cayley tables and by Wilson [37] for permutation groups.

$N \triangleleft G$ is a minimal normal subgroup if $|N| > 1$ and N does not contain any nonidentity normal subgroup of G other than itself.

FACT 2.2. Every minimal normal subgroup is characteristic.

Recall that the socle of a group G, denoted by $\mathrm{Soc}(G)$, is the product of the minimal normal subgroups of G.

Now let us look at the case where G is semisimple. In this case $\mathrm{Soc}(G)$ is the direct product of all minimal normal subgroups. We group this direct product by isomorphism types of the minimal normal subgroups as

$$\mathrm{Soc}(G) = \prod_{i=1}^{d}\prod_{j=1}^{z_i} N_{i,j} \cong \prod_{i=1}^{d} K_i^{z_i}, \tag{2.1}$$

where the $N_{i,j}$ are the minimal normal subgroups and $(\forall i,j)(N_{i,j} \cong K_i)$. The K_i are pairwise non-isomorphic characteristically simple groups.

We refine the decomposition (2.1) to simple factors, and then lump the isomorphic simple factors together to obtain the following decomposition:

$$\mathrm{Soc}(G) = \prod_{i=1}^{r}\prod_{j=1}^{k_i} V_{i,j} \cong \prod_{i=1}^{r} T_i^{k_i}, \tag{2.2}$$

where $(\forall i,j)(V_{i,j} \cong T_i)$, and the T_i are pairwise non-isomorophic simple groups.

By Proposition 2.1, we can decide isomorphism of $\mathrm{Soc}(G)$ and $\mathrm{Soc}(H)$ in polynomial time. In fact, we can find the two product decompositions of the socles described above in polynomial time, and decide isomorphism of the factors.

2.6 Diagonals and diagonal respecting isomorphisms.

Definition 2.2. Let $V_1, \ldots, V_r$ be isomorphic groups, $(\forall i)(V_i \cong T)$. A *diagonal* of $(V_1, \ldots, V_r)$ is an embedding $\phi : T \hookrightarrow \prod_{i=1}^{r} V_i$ such that $\mathrm{Im}(\phi)$ is a subdirect product of the V_i.

More generally if we have a system of groups $(V_{1,1}, \ldots, V_{1,k_1}), \ldots, (V_{r,1}, \ldots, V_{r,k_r})$, where for every $i \le r$, and every $j \le k_i$, we have $V_{i,j} \cong T_i$. Then a *diagonal product* of the system $(V_{i,j})$ is an embedding $\phi_i \times \cdots \times \phi_r : T_1 \times \cdots \times T_r \hookrightarrow \prod_{j=1}^{k_1} V_{1,j} \times \cdots \times \prod_{j=1}^{k_r} V_{r,j}$, where each ϕ_i is a diagonal of $(V_{i,j})_{j=1}^{k_i}$.

A diagonal establishes an identification of the factors.

The *standard diagonal* of T^k is the map $\Delta : t \to (t, \ldots, t)$. Similarly, the *standard diagonal product* of $\prod_{i=1}^{r} T_i^{k_i}$ is the map $\Delta = \Delta_1 \times \cdots \times \Delta_r$, where for every i, Δ_i is the standard diagonal for $T_i^{k_i}$.

We will be interested in isomorphisms that respect diagonals. In order to even define this concept, we need to talk about isomorphisms that respect the decomposition of the groups into direct products.

Definition 2.3. (ISOp) Given two groups X, Y, along with product decompositions $X = \prod_{i=1}^{r}\prod_{j=1}^{k_i} V_{i,j}$, $Y = \prod_{i=1}^{r}\prod_{j=1}^{k_i} U_{i,j}$, where $(\forall i,j)(U_{i,j} \cong V_{i,j} \cong T_i)$, we say that an isomorphism $\chi : G \to H$ *respects* the decompositions $\mathcal{V} = (V_{i,j})$ and $\mathcal{U} = (U_{i,j})$ if $(\forall i,j)(\exists j')(\chi(V_{i,j}) = U_{i,j'})$. We denote the set of isomorphisms that respect decompositions $\mathcal{V}$ and $\mathcal{U}$ by $\mathrm{ISOp}((X,\mathcal{V}),(Y,\mathcal{U}))$, where the 'p' stands for product decomposition. If the decompositions are understood from context, we will write $\mathrm{ISOp}(X,Y)$.

Definition 2.4. (ISOd) Let X, Y be two groups with product decompositions $X = \prod_{i=1}^{r}\prod_{j=1}^{k_i} V_{i,j}$, and $Y = \prod_{i=1}^{r}\prod_{j=1}^{k_i} U_{i,j}$, where $(\forall i,j)(U_{i,j} \cong V_{i,j} \cong T_i)$, and let φ, ψ be diagonal products of $\mathcal{V} = (V_{i,j})$ and $\mathcal{U} = (U_{i,j})$ respectively. We say that an isomorphism $\chi \in \mathrm{ISOp}((X,\mathcal{V}),(Y,\mathcal{U}))$ *respects* the diagonal products φ and ψ if $\varphi\chi = \psi$. We denote the set of diagonal product respecting isomorphisms by $\mathrm{ISOd}((X,\mathcal{V}),(Y,\mathcal{U});\varphi,\psi)$. Again, if $\mathcal{V}$ and $\mathcal{U}$ are understood from context, we will omit them.

Lemma 2.2. *Let X, Y, be two groups with product decompositions $X = \prod_{i=1}^{r}\prod_{j=1}^{k_i} V_{i,j}$, $Y = \prod_{i=1}^{r}\prod_{j=1}^{k_i} U_{i,j}$, where $(\forall i,j)(U_{i,j} \cong V_{i,j} \cong T_i)$. Let $\mathcal{V} = (V_{i,j})$, and $\mathcal{U} = (U_{i,j})$ be the product decompositions. Fix a diagonal product $\varphi = (\varphi_1, \ldots, \varphi_r)$ of $(V_{i,j})$, and let $\mathcal{D}$ be the set of all diagonal products of $(U_{i,j})$. Then*

$$\mathrm{ISOp}((X,\mathcal{V}),(Y,\mathcal{U})) = \bigcup_{\psi\in\mathcal{D}} \mathrm{ISOd}((X,\mathcal{V}),(Y,\mathcal{U});\varphi,\psi).$$

Proof. We need to show that given a $\chi \in \mathrm{ISOp}((X,\mathcal{V}),(Y,\mathcal{U}))$, there is some $\psi \in \mathcal{D}$ such that χ respects φ, ψ. Let $\psi = \varphi\chi$. Then ψ is a diagonal product of Y, since χ respects the product decomposition. Moreover, χ respects φ, ψ by definition. □

Lemma 2.3. *The number of diagonal products of a system of groups $((V_{i,j})_{j=1}^{k_i})_{i=1}^{r}$, where $(\forall i,j)(V_{i,j} \cong T_i)$ is $\prod_{i=1}^{r} |\mathrm{Aut}(T_i)|^{k_i}$.*

Proof. For each i, let d_i be the number of diagonals $\varphi_i : T_i \to \prod_{j=1}^{k_i} V_{i,j}$. The number of diagonal products will be $\prod_{i=1}^{r} d_i$. Now fix some i. The set of diagonals $\varphi_i : T_i \to \prod_{j=1}^{k_i} V_{i,j}$ is in bijective correspondence to the set $\prod_{j=1}^{k_i} \mathrm{ISO}(T_i, V_{i,j})$, since $\varphi_i(T_i)$ is a subdirect product. But $T_i \cong V_{i,j}$ by assumption, and hence $(\forall j)(|\mathrm{ISO}(T_i, V_{i,j})| = |\mathrm{Aut}(T_i)|)$. Therefore $d_i = |\mathrm{Aut}(T_i)|^{k_i}$. □

We now look at the case where the factors of the decomposition are simple, since this will be a case we encounter in our algorithms.

FACT 2.3. *If X is the direct product of simple groups, then X decomposes uniquely as*

$$X = \prod_{i=1}^{r} \prod_{j=1}^{k_i} V_{i,j}, \tag{2.3}$$

where $V_{i,j} \cong T_i$, and the T_i are pairwise non-isomorphic simple groups.

In the above fact, the subgroups $V_{i,j}$ are unique, not just their isomorphism types. Note how this is not true for $\mathbb{Z}_p \times \cdots \times \mathbb{Z}_p$. (Recall that when we say 'simple group' we mean simple non-abelian.)

DEFINITION 2.5. If T is simple, and $X \cong T^k$, then a *diagonal* of X is a diagonal of the unique decomposition $X = \prod_{i=1}^{r} V_i$ into factors $V_i \cong T$. Moreover, if $T_1, \ldots, T_r$ are non-isomorphic simple groups, and $X \cong \prod_{i=1}^{r} T_i^{k_i}$, then a *diagonal product* of X is a diagonal product of the unique decomposition $X = \prod_{i=1}^{r} \prod_{j=1}^{k_i} V_{i,j}$ into factors $V_{i,j} \cong T_i$.

In particular, if X and Y are groups that are direct products of simple groups, when we write $\mathrm{ISOp}(X, Y)$, and $\mathrm{ISOd}(X, Y; \varphi, \psi)$ omitting the decomposition, we mean the unique decomposition given by Equation (2.3).

3 Restriction of isomorphisms to the socle: the key lemma

The following lemma is central to both algorithms in this paper (Theorems 1.1 and 1.2).

LEMMA 3.1. *Let G and H be groups and $R \lhd G$ and $S \lhd H$ normal subgroups with trivial centralizers. Let $\alpha : G \to G^* \leq \mathrm{Aut}(R)$ and $\beta : H \to H^* \leq \mathrm{Aut}(S)$ be the faithful permutation representations of G and H via conjugation action on R and S, resp. Let $f : R \to S$ be an isomorphism. Then f extends to an isomorphism $\hat{f} : G \to H$ if and only if f is a permutational isomorphism between G^* and H^*; and if so, $\hat{f} = \alpha f^* \beta^{-1}$ where f^* is the isomorphism $G^* \to H^*$ induced by f.*

Proof. By applying the inverse of f, we may assume $R = S$, and f is the identity. We claim that if f^* exists, it must be the identity. Suppose f^* exists. Let $\hat{f}$ denote the corresponding $G \to H$ isomorphism. So $\hat{f}|_R = \mathrm{id}$. Let $g \in G$ and let $g^* = G^*$ be the corresponding automorphism of R. We need to show that $f^*(g^*) = g^*$, that is, for all $r \in R$, $r^g = r^{\hat{f}(g)}$, i. e.,

$$g^{-1} r g = \hat{f}(g)^{-1} r \hat{f}(g). \tag{3.4}$$

But $\hat{f}(g)^{-1} r \hat{f}(g) = \hat{f}(g)^{-1} \hat{f}(r) \hat{f}(g) = \hat{f}(g^{-1} r g) = g^{-1} r g$ because $g^{-1} r g \in R$, proving (3.4). □

PROPOSITION 3.1. *Given two permutation groups G and H given by generators, and a bijection f of the domains, we can decide whether f is a permutational isomorphism of G and H in polynomial time.*

Proof. We can check membership of the f-images of the generators of G in H and vice versa [15]. □

COROLLARY 3.1. *Let G and H be two groups given by Cayley tables. Let $R \lhd G$ and $S \lhd H$ be normal subgroups with trivial centralizers. Assume $f : R \to S$ is an isomorphism. Then (a) f extends in at most one way to an isomorphism $\hat{f} : G \to H$; and (b) given f we can decide if $\hat{f}$ exists, and find it if it does, in polynomial time.*

Proof. Part(a) follows from Lemma 3.1. Part (b) follows from (a) and Proposition 3.1. □

4 The $n^{O(\log \log n)}$ algorithm

4.1 Semisimple groups: reduction to fixed diagonal products of the socles. Let G and H be two semisimple groups, with $\mathrm{Soc}(G) \cong \mathrm{Soc}(H) \cong \prod_{i=1}^{r} T_i^{k_i}$, where the T_i are pairwise non-isomorphic non-abelian simple groups. Corollary 3.1 applied to G and H, with $R = \mathrm{Soc}(G)$, $S = \mathrm{Soc}(H)$ implies that the isomorphisms between G and H are determined by the isomorphisms of their socles. For φ, ψ diagonal products of $\mathrm{Soc}(G)$ and $\mathrm{Soc}(H)$ respectively, let the set of isomorphisms that respect diagonal products of the socle be:

$$\begin{aligned}&\mathrm{ISOds}(G, H; \varphi, \psi) = \\ &\{\chi \in \mathrm{ISO}(G, H) : \\ &\quad \chi|_{\mathrm{Soc}(G)} \in \mathrm{ISOd}(\mathrm{Soc}(G), \mathrm{Soc}(H); \varphi, \psi)\}.\end{aligned}$$

Combining Corollary 3.1 and Lemma 2.2 we get the following corollary.

COROLLARY 4.1. *Let G, H semisimple, φ a diagonal product of $\mathrm{Soc}(G)$, and $\mathcal{D}$ the set of diagonal products of $\mathrm{Soc}(H)$. Then*

$$\mathrm{ISO}(G, H) = \bigcup_{\psi \in \mathcal{D}} \mathrm{ISOds}(G, H; \varphi, \psi).$$

The next lemma shows that this reduces ISO to polynomially many instances of ISOds.

LEMMA 4.1. *Let $\mathcal{D}$ be defined as in the previous corollary. Then $|\mathcal{D}| \leq |H|^2$.*

Proof. Each T_i is simple, hence it is generated by 2 elements. Therefore $|\mathrm{Aut}(T_i)| = |T_i|^2$. Since an automorphism is determined by the images of the generators.

So, by Lemma 2.3, the number of diagonal products is bounded by $\prod_{i=1}^{r}\prod_{j=1}^{k_i}|\operatorname{Aut}(T_i)| = \prod_{i=1}^{r}\prod_{j=1}^{k_i}|T_i|^2 \le |\operatorname{Soc}(H)|^2 \le |H|^2$. □

NOTATION. If a group G is a direct product $G = \prod_{i=1}^{k} G_i$, we will denote the set of factors of G by $\operatorname{Fac}(G) = \{G_1, \ldots, G_k\}$.

DEFINITION 4.1. *Given two groups $G = \prod_{i=1}^{k} G_i$, $H = \prod_{i=1}^{k} H_i$, where the G_i, H_i are simple, a bijection f between* $\operatorname{Fac}(G)$ *and* $\operatorname{Fac}(H)$ *is said to* respect the isomorphism types *if $f(G_i) \cong G_i$.*

LEMMA 4.2. *Let G and H be semisimple groups, with $\operatorname{Soc}(G) \cong \operatorname{Soc}(H) \cong \prod_{i=1}^{r} T_i^{k_i}$. Then (a) every isomorphism $\chi \in \operatorname{ISOds}(G, H; \varphi, \psi)$ is determined by the isomorphism respecting bijection it induces between* $\operatorname{Fac}(\operatorname{Soc}(G))$ *and* $\operatorname{Fac}(\operatorname{Soc}(H))$*; and (b) given a bijection $f : \operatorname{Fac}(\operatorname{Soc}(G)) \to \operatorname{Fac}(\operatorname{Soc}(H))$ that respects isomorphism types, we can check whether it arises as the action of some $\chi \in \operatorname{ISOds}(G, H; \varphi, \psi)$, and if so find that unique χ, in polynomial time.*

Proof. By Corollary 3.1, it suffices to prove the statement for every $\chi \in \operatorname{ISOd}(\operatorname{Soc}(G), \operatorname{Soc}(H); \varphi, \psi)$. Applying decomposition (2.2), let us write $\operatorname{Soc}(G) = \prod_{i=1}^{r}\prod_{j=1}^{k_i} V_{i,j}$, and $\operatorname{Soc}(H) = \prod_{i=1}^{r}\prod_{j=1}^{k_i} U_{i,j}$, where $(\forall i, j)(V_{i,j} \cong U_{i,j} \cong T_i)$. Let $\varphi = \varphi_1 \times \cdots \times \varphi_r$, where $\varphi_i : T_i \hookrightarrow \prod_{j=1}^{k_i} V_{i,j}$ is a diagonal. Similarly define $(\psi_i)_{i=1}^{r}$. For all i, j, let $\pi_{i,j} : \operatorname{Soc}(G) \to V_{i,j}$ and $\rho_{i,j} : \operatorname{Soc}(H) \to U_{i,j}$ be the projection maps onto the components. Notice that for every i, j, $(\pi_{i,j} \circ \varphi_i)$ is an isomorphism between $V_{i,j}$ and T_i. Now let $\chi(V_{i,j}) = U_{i,\ell}$, then we claim this determines $\chi|_{V_{i,j}}$. Indeed in order for χ to respect the diagonal products, we must have

$$\chi|_{V_{i,j}} = (\pi_{i,j} \circ \varphi^i)^{-1} \circ (\pi_{i,\ell} \circ \psi^i).$$

To prove (b), construct $\chi_f : \operatorname{Soc}(G) \to \operatorname{Soc}(H)$ as follows. For every $i \le r, j \le k_i$, let $f(V_{i,j}) = U_{i,\ell}$. Then $\chi_f|_{V_{i,j}} = (\pi_{i,j} \circ \varphi^i)^{-1} \circ (\pi_{i,\ell} \circ \psi^i)$. Now $\chi_f \in \operatorname{ISOd}(\operatorname{Soc}(G), \operatorname{Soc}(H); \varphi, \psi)$, and by Corollary 3.1, we can check in polynomial time if it extends to an isomorphism of G and H. □

4.2 Algorithms that list all the isomorphisms.

THEOREM 4.1. *Let G and H be two semisimple groups, with $\operatorname{Soc}(G) \cong \operatorname{Soc}(H) \cong \prod_{i=1}^{r} T_i^{k_i}$, where the T_i are pairwise non-isomorphic simple groups. Then we can decide isomorphism of G and H in time $n^{O(1)}\prod_{i=1}^{r} k_i!$. In fact, all isomorphisms can be listed within this time bound.*

Proof. Let $\operatorname{Soc}(G) \cong \operatorname{Soc}(H) \cong \prod_{i=1}^{r} T_i^{k_i}$.

By Corollary 4.1 and Lemma 4.1, $\operatorname{ISO}(G, H)$ reduces to n^2 instances of $\operatorname{ISOds}(G, H; \varphi, \psi)$. To find $\operatorname{ISOds}(G, H; \varphi, \psi)$, iterate over all isomorphism respecting bijections f between the factors of the socles, and apply Lemma 4.2(b) to each such f. Notice that Lemma 4.2(a) guarantees that we will find all of $\operatorname{ISOds}(G, H; \varphi, \psi)$ this way. This algorithm computes $\operatorname{ISOds}(G, H; \varphi, \psi)$ in time $n^{O(1)}|\prod_{i=1}^{r} S_{k_i}| = n^{O(1)}\prod_{i=1}^{r} k_i!$. Hence The running time to find $\operatorname{ISO}(G, H)$ will be $n^2 n^{O(1)}\prod_{i=1}^{r} k_i! = n^{O(1)}\prod_{i=1}^{r} k_i!$. □

Let $k = \sum_{i=1}^{r} k_i$ be the total number of direct factors of the socle. We note that $\prod_{i=1}^{r} k_i! \le (\max k_i)^k$. Moreover, each component is simple, and hence has order at least 60, and the product of the components is a subgroup of G. Therefore

$$k \le \log_{60} n. \tag{4.5}$$

The following corollaries are now immediate.

COROLLARY 4.2. *Isomorphism of two semisimple groups G and H of order n can be decided in time $n^{O(1)+c\log\log n}$, where $c = 1/\log(60) \approx 0.16929$. In fact, all isomorphisms can be listed within this time bound.*

The following corollary answers a question raised by V. Arvind [1].

COROLLARY 4.3. *Let G and H be semisimple. If the k_i are bounded (each simple groups occurs a bounded number of times as a factor of the socle), then we can decide isomorphism and list all isomorphisms in polynomial time.*

Recall that k, the number of simple factors of the socle, is at most $\log n$. If it happens to be $O(\log n/\log\log n)$, then we have a stronger conclusion.

COROLLARY 4.4. *Let G and H be semisimple. If $k = O(\log n/\log\log n)$, then we can decide isomorphism of G and H , and list all the isomorphisms between G and H in polynomial time.*

Note that the condition $k = O(\log n/\log\log n)$ necessarily holds if at least a constant fraction of the simple factors of the socle has order $(\log n)^{\Omega(1)}$.

5 Code Equivalence

5.1 Codes. A *string of length n* over a finite alphabet Γ is a map $x\colon A \to \Gamma$, where $|A| = n$. For a string $x \in \Gamma^A$ and a bijection $\pi\colon A \to B$ we define the string $x^\pi \in \Gamma^B$ by setting $x^\pi(i) = x(i^{\pi^{-1}})$ $(i \in B)$.

DEFINITION 5.1. *A* code *of length n over Γ with index set A ($|A| = n$) is a subset of Γ^A. For a bijection $\pi\colon A \to B$, we define $\mathcal{A}^\pi = \{x^\pi : x \in \mathcal{A}\} \subseteq \Gamma^B$.*

We require a generalization of the above to multiple alphabets: let $\Gamma_1, \ldots, \Gamma_r$ be disjoint finite alphabets. A string of length $(k_1, \ldots, k_r)$ over $(\Gamma_1, \ldots, \Gamma_r)$ is a set of maps $x_i\colon A_i \to \Gamma_i$, denoted collectively as x, where $|A_i| = k_i$. The set of all such strings is $\prod_{i=1}^r \Gamma_i^{A_i}$. For a string $x \in \prod_{i=1}^r \Gamma_i^{A_i}$ and bijections $\pi_i\colon A_i \to B_i$, denoted collectively by $\pi = (\pi_1, \ldots, \pi_r)\colon \bigcup_{i=1}^r A_i \to \bigcup_{i=1}^r B_i$, we define a string $x^\pi \in \prod_{i=1}^r \Gamma_i^{B_i}$ by setting $x^\pi(i,j) = x_i^{\pi_i}(j) = x_i(j^{\pi_i^{-1}})$, where (i,j) denotes the element $j \in A_i$.

DEFINITION 5.2. *A code of length $(k_1, \ldots, k_r)$ with domain $(A_1, \ldots, A_r)$ is a subset $\mathcal{A} \subseteq \prod_{i=1}^r \Gamma_i^{A_i}$. We define $\mathcal{A}^\pi$ as before.*

DEFINITION 5.3. *If $\mathcal{A} \subseteq \prod_{i=1}^r \Gamma_i^{A_i}$ and $\mathcal{B} \subseteq \prod_{i=1}^r \Gamma_i^{B_i}$ are codes, then a set of bijections $\pi_i\colon A_i \to B_i$ ($i = 1, \ldots, r$) is a* code equivalence *if $\mathcal{A}^\pi = \mathcal{B}$ where $\pi = (\pi_1, \ldots, \pi_r)$.*

The set of all $\mathcal{A} \to \mathcal{B}$ code equivalences will be denoted by $\mathrm{EQ}(\mathcal{A}, \mathcal{B})$. Each $\pi \in \mathrm{EQ}(\mathcal{A}, \mathcal{B})$ naturally induces a bijection between $\mathcal{A}$ and $\mathcal{B}$, by sending x to x^π, $x \in \mathcal{A}$. We denote by $\hat{\pi}$ the induced map on strings, and let $\widehat{\mathrm{EQ}}(\mathcal{A}, \mathcal{B}) = \{\hat{\pi} : \pi \in \mathrm{EQ}(\mathcal{A}, \mathcal{B})\}$.

Note that if $\pi \in \mathrm{EQ}(\mathcal{A}, \mathcal{B})$, then $\mathrm{EQ}(\mathcal{A}, \mathcal{B}) = \mathrm{EQ}(\mathcal{A}, \mathcal{A})\pi$, so $\mathrm{EQ}(\mathcal{A}, \mathcal{B})$ is either empty or a coset of the permutation group $\mathrm{EQ}(\mathcal{A}, \mathcal{A})$.

5.2 Algorithm for general code equivalence. We describe a modification of Luks's hypergraph isomorphism test [26], to solve equivalence of explicitly given codes. The proof is based on Luks's dynamic programming idea (table lookup), and therefore, as in Luks's case, requires not only simply exponential time but also simply exponential space in terms of n, the length of the strings (or the number of vertices of the hypergraph). Note, however, that in the case of dense sets of strings (or dense hypergraphs), the length of the input is also exponential, and the algorithms—Luks's as well as ours—are polynomial-time (quadratic).

THEOREM 5.1. *Given the codes $\mathcal{A} \subseteq \prod_{i=1}^r \Gamma_i^{A_i}$ and $\mathcal{B} \subseteq \prod_{i=1}^r \Gamma_i^{B_i}$ (as explict lists of strings), the set of their equivalences can be found in time $\prod_{i=1}^r (c|\Gamma_i|)^{2k_i}$ for some absolute constant c, where $k_i = |A_i| = |B_i|$.*

Proof. For subsets $U_i \subseteq A_i$ we call the functions $y\colon \bigcup_{i=1}^r U_i \to \bigcup_{i=1}^r \Gamma_i$ mapping U_i into Γ_i "partial strings over $A = (A_1, \ldots, A_r)$." We call the tuple $(|U_1|, \ldots, |U_r|)$ the *length* of y. For every partial string y over A, let $\mathcal{A}_y$ be the set of those strings in $\mathcal{A}$ that are extensions of y; we make analogous definitions for $\mathcal{B}$.

Our dynamic programming table will consist of the following sets: for every pair y, z of partial strings, y over A and z over B, of equal length and with equal distribution of letters on their respective ranges, we store the set $I(y,z)$ of equivalences of the restriction of $\mathcal{A}_y$ to $A \setminus \mathrm{dom}(y)$ with the restriction of $\mathcal{B}_z$ to $B \setminus \mathrm{dom}(z)$. Note that these sets are either empty or cosets, so we store them by a set of generators and a coset representative.

We start with full strings y, z and work our way down to $\mathrm{dom}(y) = \mathrm{dom}(z) = \emptyset$, at which point we shall have constructed all $\mathcal{A} \to \mathcal{B}$ equivalences.

When y, z are full strings, we have $|\mathcal{A}_y| \leq 1$, $|\mathcal{B}_z| \leq 1$, and the problem is trivial.

Now let y, z be proper partial strings. To construct $I(y,z)$ we augment the domain of y with one element $r \in A$, and the domain of z with one element, $s \in B$. We fix r, say $r \in A_i$, and make all possible choices of $s \in B_i$. To find those elements of $I(y,z)$ that take r to s, we consider each of the $|\Gamma_i|$ possible values our strings can take at r; to find the corresponding coset for each value is a table lookup; and we take the intersection of the $|\Gamma_i|$ cosets. Then we take the union for all s.

Analysis. The number of partial strings over A is $\prod_{i=1}^r (|\Gamma_i| + 1)^{k_i}$, where $|A_i| = k_i$, so the number of sets to store is less than $\prod_{i=1}^r (|\Gamma_i| + 1)^{2k_i}$. The cost of coset intersection is $\exp(\widetilde{O}(\sqrt{n}))$ (where $n = \sum_i k_i$), negligible compared to the size of the dynamic programming table. □

6 The main algorithm

Now we can present our main algorithm, which will use code equivalence as a subroutine.

6.1 Reduction to fixed diagonal products of the systems of minimal normal subgroups. Let G and H be semisimple groups. Recall that the socles are the product of the minimal normal subgroups. Let us group the terms in this product based on their isomorphism types as follows. Applying decomposition (2.1), let us write $\mathrm{Soc}(G) = \prod_{i=1}^d \prod_{j=1}^{z_i} N_{i,j}$, and $\mathrm{Soc}(H) = \prod_{i=1}^d \prod_{j=1}^{z_i} M_{i,j}$, where the $N_{i,j}$ and $M_{i,j}$ are the minimal normal subgroups of G and H respectively, and $(\forall i,j)(N_{i,j} \cong M_{i,j} \cong K_i)$. Let φ be a diagonal product of the system $\mathcal{N} = (N_{i,j})$, and ψ a diagonal product of $\mathcal{M} = (M_{i,j})$. Define the set of isomorphisms respecting diagonal products of the system of minimal normal subgroups:

$$\mathrm{ISOdn}(\mathrm{Soc}(G), \mathrm{Soc}(H); \varphi, \psi) =$$

$$\mathrm{ISOd}((\mathrm{Soc}(G),\mathcal{N}),(\mathrm{Soc}(H),\mathcal{M});\varphi,\psi).$$

And let us denote the extensions of the isomorphisms to isomorphisms of G and H by

$$\begin{aligned}&\mathrm{ISOdns}(G,H;\varphi,\psi) =\\ &\{\chi\in\mathrm{ISO}(G,H):\\ &\quad \chi|_{\mathrm{Soc}(G)}\in\mathrm{ISOdn}(\mathrm{Soc}(G),\mathrm{Soc}(H);\varphi,\psi)\}.\end{aligned}$$

Recall that every minimal normal subgroup is characteristically simple, and hence the direct product of isomorphic simple groups. Therefore for every i, we have $K_i = T_i^{t_i}$, for some simple group T_i. Let $\tau = \prod_{i=1}^d (t_i!)^{z_i}$. We will show that we can reduce ISO to $n^{O(1)}\tau$ instances of ISOdns, and that we can solve each instance of ISOdns in time $n^{O(1)}\tau^2$, by transforming it into an instance of code equivalence.

LEMMA 6.1. *Let G and H be defined as above. Fix a diagonal product φ of $(N_{i,j})$, and let $\mathcal{D}$ be the set of all diagonal products of $(M_{i,j})$. Then,*

$$\mathrm{ISO}(G,H) = \bigcup_{\psi\in\mathcal{D}} \mathrm{ISOdns}(G,H;\varphi,\psi).$$

Proof. Note that it is possible that a $\chi \in \mathrm{ISO}(\mathrm{Soc}(G),\mathrm{Soc}(H))$ does not respect the decomposition of the socles as the direct product of minimal normal subgroups, i. e., $\chi(N_{i,j})$ is not one of the $M_{i,j'}$. However, such a χ will not extend to an isomorphism of G and H. In particular, by Corollary 3.1, every isomorphism of G and H is the unique extension of an isomorphism of $\mathrm{ISOp}((\mathrm{Soc}(G),\mathcal{N}),(\mathrm{Soc}(H),\mathcal{M}))$. Therefore the result follows from Lemma 2.2. □

LEMMA 6.2. *Let G be a semisimple group of order n, and $(N_{i,j})$ the system of minimal normal subgroups of G defined as above. Then the number of diagonal products of the $(N_{i,j})$ is bounded by $n^{O(1)}\prod_{i=1}^d (t_i!)^{z_i}$.*

Proof. By Lemma 2.3, there are $\prod_{i=1}^d |\operatorname{Aut}(K_i)|^{z_i}$ diagonal products of this system. But

$$\begin{aligned}|\operatorname{Aut}(K_i)| &= |\operatorname{Aut}(T_i^{t_i})| = |\operatorname{Aut}(T_i)\wr S_{t_i}|\\ &= |\operatorname{Aut}(T_i)|^{t_i} t_i! = |T_i|^{2t_i}(t_i!).\end{aligned}$$

Hence the number of diagonal products is bounded by:

$$\prod_{i=1}^d |T_i|^{2t_i z_i}(t_i!)^{z_i} \le |\mathrm{Soc}(G)|^2 \prod_{i=1}^d t_i!^{z_i} \le n^{O(1)} \prod_{i=1}^d t_i!^{z_i}.$$

□

6.2 Embedding into the automorphism group of the socle. Let G and H be semisimple groups, and consider decomposition (2.1) of $\mathrm{Soc}(G)$ and $\mathrm{Soc}(H)$ as the product of the minimal normal subgroups. Let φ, ψ be diagonal products of the systems $(N_{i,j})$ and $(M_{i,j})$ respectively. For notational convenience, let $\mathcal{K} = \prod_{i=1}^d K_i^{z_i}$. Pick an $\alpha_\varphi \in \mathrm{ISOdn}(\mathrm{Soc}(G),\mathcal{K};\varphi,\Delta)$, and a $\beta_\psi \in \mathrm{ISOdn}(\mathrm{Soc}(H),\mathcal{K};\psi,\Delta)$, where Δ is the standard diagonal product of $\prod_{i=1}^d K_i^{z_i}$. By Lemma 2.1, the conjugation action of G and H on their socles gives us corresponding embeddings $\alpha_\varphi^* : G \hookrightarrow \mathrm{Aut}(\mathcal{K})$, and $\beta_\psi^* : H \hookrightarrow \mathrm{Aut}(\mathcal{K})$, with $\mathrm{Soc}(G^*) = \mathrm{Soc}(H^*) = \mathrm{Inn}(\mathcal{K})$. Let $G^* = \alpha_\varphi^*(G)$, and $H^* = \beta_\psi^*(H)$. Notice that in fact $G^*, H^* \le \prod_{i=1}^d Aut(K_i)^{z_i}$, since the conjugation action of G and H on their minimal normal subgroup fixes them (they are normal). Moreover,

$$\mathrm{ISO}(G,H) = \alpha_\varphi^* \,\mathrm{ISO}(G^*,H^*)(\beta_\psi^*)^{-1},$$

and

$$\begin{aligned}&\mathrm{ISOdns}(G,H;\varphi,\psi)\\ &= \alpha_\varphi^* \,\mathrm{ISOdns}(G^*,H^*;\Delta,\Delta)(\beta_\psi^*)^{-1}.\end{aligned} \tag{6.6}$$

6.3 Reduction to code equivalence. From the previous subsection, finding $\mathrm{ISOdns}(G,H;\varphi,\psi)$ reduces to the case of two groups $G^*, H^* \le \prod_{i=1}^d \mathrm{Aut}(K_i)^{z_i}$, with $\mathrm{Soc}(G^*) = \mathrm{Soc}(H^*) = \mathrm{Inn}(\mathcal{K})$, and we need to compute the isomorphisms of G^* and H^* that preserve the standard diagonal product of the system of minimal normal subgroups of the socle. We will show how to formulate this problem as an instance of code equivalence. Let us call $\mathrm{ISO}^*(G^*,H^*) = \mathrm{ISOdns}(G^*,H^*;\Delta,\Delta)$ the set of isomorphisms that respect the standard diagonal products of the system of minimal normal subgroups of the socles.

Since $G^*, H^* \le \prod_{i=1}^d \mathrm{Aut}(K_i)^{z_i}$, we can view G^* and H^* as codes over the alphabets $\Gamma_i = \mathrm{Aut}(K_i)$. Let $\overline{G}$ and $\overline{H}$ be these codes.

LEMMA 6.3.

$$\mathrm{ISO}^*(G^*,H^*) = \widehat{\mathrm{EQ}}(\overline{G},\overline{H})$$

Proof. The $\subseteq$ follows by Corollary 3.1. □

6.4 The algorithm. Next we state our main result in detail.

THEOREM 6.1. *Given G and H semisimple. Let $\mathrm{Soc}(G) = \prod_{i=1}^d \prod_{j=1}^{z_i} N_{i,j}$, and $\mathrm{Soc}(H) = \prod_{i=1}^d \prod_{j=1}^{z_i} M_{i,j}$, where the $N_{i,j}$ and $M_{i,j}$ are the minimal normal subgroups of G and H respectively, and $(\forall i,j)(N_{i,j} \cong M_{i,j} \cong T_i^{t_i})$, where the T_i*

are simple. Then we can find $\mathrm{ISO}(G,H)$ *in time* $n^{O(1)}(\prod_{i=1}^{d}(t_i!)^{z_i})^3$.

We note that $\sum_{i=1}^{d} t_i z_i$ is the number of simple factors of the socle, and therefore, by Equation (4.5),

$$\sum_{i=1}^{d} t_i z_i \le \log_{60} n.$$

Proof. Set $\tau = \prod_{i=1}^{d}(t_i!)^{z_i}$. By Lemmas 6.1 and 6.2, the problem reduces to $n^{O(1)}\tau$ instances of $\mathrm{ISOdns}(G,H;\varphi,\psi)$. To find $\mathrm{ISOdns}(G,H;\varphi,\psi)$, we lift G and H to subgroups G^* and H^* of $\prod_{i=1}^{d}\mathrm{Aut}(K_i)^{z_i}$. Using Equation (6.6), it suffices to find $\mathrm{ISO}^*(G^*,H^*)$, which is a code equivalence problem by Lemma 6.3. We can solve the code equivalence problem in time

$$\prod_{i=1}^{d}(c\,|\,\mathrm{Aut}(K_i)|)^{2z_i} \le \prod_{i=1}^{d}(c|T_i|^{2t_i}(t_i!))^{2z_i}$$

$$\le (\prod_{i=1}^{d}|T_i|^{Cz_it_i})\tau^2 \le |\,\mathrm{Soc}(G)|^C\tau \le n^{O(1)}\tau^2.$$

Since our algorithm runs the code equivalence subroutine $n^{O(1)}\tau$ times, our total running time will be $n^{O(1)}\tau^3$. □

We now state some corollaries of Theorem 6.1 that subsume the results from Section 4 (up to constant factors in the exponent of the running time).

Recall that $t(G)$ is the maximum width of the minimal normal subgroups of G. In particular, if $\mathrm{Soc}(G)$ is decomposed as above, $t(G) = \max_i t_i$.

COROLLARY 6.1. *Let* G *and* H *be semisimple groups of order* n. *We can find* $\mathrm{ISO}(G,H)$ *in time* $n^{c\log(t(G))+O(1)}$, *where* $c = 6/\log(60) \approx 1.0158$.

Proof. $\prod_{i=1}^{d}(t_i!)^{z_i} \le t^{\sum 2z_it_i} \le t^{2\log n} \le n^{2\log t}$. The second to last inequality follows because $|G| \ge |\,\mathrm{Soc}(G)| \ge 60^{\sum t_iz_i}$, and hence $\log n \ge \sum t_iz_i$. □

COROLLARY 6.2. *Let* G *and* H *be semisimple groups of order* n, *with* $t(G)$ *bounded by a constant. Then we can find* $\mathrm{ISO}(G,H)$ *in polynomial time.*

7 Appendix

7.1 Algorithm for linear code equivalence. In this section we present the material of an unpublished note by the first author [6].

We give an algorithm that tests the equivalence of linear codes of length n over a field F in time $(2+o(1))^n$, assuming field operations at unit cost. To the best of our knowledge no simply-exponential-time algorithm was previously known.

The set of nonsingular $n \times n$ matrices over a field F is denoted $\mathrm{GL}_n(F)$. A *linear code* of length n is a subspace $U \le F^n$. A $d \times n$ matrix over F *generates* the code U if the rows of A span U.

Note that a *linear code* is a group code (c.f. Definition 1.1) with alphabet $\Gamma = F$, where F is a field.

FACT 7.1. *Let* U, W *be* d*-dimensional codes of length* n *over* F, *generated by* $d \times n$ *matrices* A, B, *respectively. Then* U *and* W *are equivalent if and only if there is a permutation matrix* $P \in \mathrm{GL}_n(F)$ *and a matrix* $T \in \mathrm{GL}_d(F)$ *such that* $B = TAP$.

THEOREM 7.1. *Equivalence of* d*-dimensional linear codes of length* n *(over any field) given by generator matrices can be reduced to* $\binom{n}{d}$ *instances of isomorphism of* $d \times (n-d)$ *bipartite graphs with colored edges. The reduction is polynomial-time, assuming field operations at unit cost.*

Proof. Throughout the proof, all matrices denoted by T, T', T_i belong to $\mathrm{GL}_d(F)$ and all matrices denoted by P, P', P_i are permutation matrices of appropriate dimensions.

Let $A, B \in F^{d\times n}$ be matrices of rank d. We need to find the set of pairs (T^{-1}, P) as in Fact 7.1 such that $B = TAP$ (note that this set is either empty or a coset).

We say that A is in *standard form* if $A = [I_d|A_1]$, i. e., the first d columns of A form the identity matrix. We can perform row operations on A followed by a permutation of the columns to transform A to a standard form, i. e., we can find T', P' such that $T'AP'$ is in standard form. Without loss of generality, $T' = I_d$ and $P' = I_n$, so A itself is in standard form.

At a multiplicative cost of $\binom{n}{d}$, we guess the subset $[d]^\sigma \subseteq [n]$ under a hypothetical $A \mapsto B$ equivalence $\sigma \in S_n$. By applying a single permutation to the columns, we reduce this case to those σ satisfying $[d]^\sigma = [d]$; let us call such σ *basic equivalences*. The corresponding permutation matrix $P(\sigma)$ can be written as a block-diagonal matrix $\mathrm{diag}(P_0, P_1)$ where $P_0 \in F^{d\times d}$ and $P_1 \in F^{(n-d)\times(n-d)}$ are permutation matrices.

We reduce the search for basic equivalences to one instance of finding the isomorphisms of F-colored $d \times (n-d)$ bipartite graphs.

Let $B = [B_0|B_1]$, where $B_0 \in F^{d\times d}$ consists of the first d columns of B. Now we have $B_0 = TP_0$, so if B_0 is singular then there are no basic equivalences.

Assume now that $B_0 \in \mathrm{GL}_d(F)$. Then $B_0^{-1}B$ is in standard form. Again without loss of generality we may therefore assume $B_0 = I_d$, i. e., B itself is in standard form.

Finally, we are now looking for basic equivalences between two codes generated by matrices in standard form $A = [I_d|A_1]$ and $B = [I_d|B_1]$. Suppose $B = TAP$ where $P = P(\sigma) = \mathrm{diag}(P_0, P_1)$ represents a basic equivalence. Then $TP_0 = I_d$, so $T = P_0^{-1}$, and $B_1 = TA_1P_1$, i. e., we are searching for permutations $\sigma_0 \in S_d$ and $\sigma_1 \in S_{n-d}$ such that $B_1 = P(\sigma_0)^{-1}A_1P(\sigma_1)$. This is precisely the isomorphism problem of the F-colored bipartite graphs whose incidence matrices are A_1 and B_1. □

COROLLARY 7.1. *The set of equivalences of two linear codes of length n (over any field) given by generator matrices can be found in $(2 + o(1))^n$ time, assuming field operations at unit cost.*

Proof. Theorem 7.1 and the $\exp(\widetilde{O}(\sqrt{n}))$ algorithm for graph isomorphism [11]. □

REMARK. The isomorphism test in [11] is stated for graphs rather than edge-colored graphs. We note that the edge-colors will only speed up the algorithm.

REMARK. In the opposite direction, Petrank and Roth [28] reduced, in polynomial time, graph isomorphism to equivalence of binary linear codes (of length $O(|V| + |E|)$).

7.2 Reduction of permutational isomorphism of permutation groups to isomorphism of semisimple groups. In this section we give a formal reduction showing that solving permutational isomorphism of permutation groups is indeed necessary for solving the isomorphism problem for semisimple groups.

PROPOSITION 7.1. *If semisimple group isomorphism can be solved in polynomial time, then permutational isomorphism of groups $K, L \le S_k$ can be solved in time polynomial in 2^k and $|K|$.*

Proof. We will show that $G = A_5 \wr K = A_5^k \rtimes K$ and $H = A_5 \wr L$ are isomorphic if and only if K and L are permutationally isomorphic. Note that the multiplication tables of G and H can easily be constructed in time polynomial in 2^k and $|K|$. (Here we assume $|K| = |L|$; if not, then $K \ncong L$ and we can handle this case easily.)

Given an isomorphism $\psi \in \mathrm{ISO}(G, H)$, it is easily verified that $\check{\psi}$ is in $\mathrm{PISO}(K, L)$. Conversely, suppose $\pi \in S_k$ is a permutational isomorphism $K \to L$; define $\psi_\pi \in \mathrm{ISO}(G, H)$ by ψ_π sends elements in the i-th copy of A_5 in G to elements in the $\pi(i)$-th copy of A_5 in H, and ψ_π sends $K \le G$ to $L \le H$ in the same manner as π sends K to L (i. e., by conjugation). □

7.3 Complexity of permutational isomorphism of permutation groups with bounded orbits. We say that a permutation group G has orbit length k if every orbit has length k; and orbit length $\le k$ if all orbits have length $\le k$. We say that a class of permutation groups has bounded orbits if there exists k such that all groups in the class have orbit length $\le k$.

THEOREM 7.2. *Permutational isomorphism of permutation groups with bounded orbits can be solved in time simply exponential in the size of the domain.*

Proof. By Proposition 7.1, this problem reduces to the polynomial-time solvability of isomorphism of semisimple groups with bounded $t(G)$ (Theorem 1.2).

Rather than using this reduction, which involves a blow-up of the problem size, we can apply the idea of our main algorithm directly. We fix an ordering of the elements of each orbit. This can be done in $\prod(k_i!)$ ways, where k_i is the length of the i-th orbit. Now $\prod(k_i!) < \max(k_i)^{\sum k_i} = \max(k_i)^n$, so we split our problem into a simply exponential number of problems where the ordering of each orbit is fixed and the isomorphisms preserve this ordering by definition. But this problem is easily seen to be an instance of the code isomorphism problem, the alphabets being the restriction of the group to each orbit. □

PROPOSITION 7.2. *There is a Karp-reduction from equivalence of linear codes over fields of (variable) prime order p to permutational isomorphism of permutation groups with orbit length $\le p^{O(1)}$. For $p = 2$, orbit length 3 suffices.*

Proof. Let U, W be linear codes of length n over $F = \mathbb{F}_p$. We will reduce the problem of finding the set of equivalences of U and W to that of finding permutational isomorphisms of two permutation groups G and H.

G and H will be permutation groups of degree nq, where q is a prime, $q \equiv 1 \pmod p$. By Linnik's celebrated result, the smallest such q satisfies $q \le p^{O(1)}$ [23].

Consider the group $X = \mathbb{Z}_q \rtimes \mathbb{Z}_p$, defined by

$$X = \langle a, b \,|\, a^p = b^q = 1,\ a^{-1}ba = b^s \rangle,$$

where s is an element of order p modulo q, i. e., p is the smallest number such that $s^p \equiv 1 \pmod q$. We represent X as a permutation group of degree q (subgroup of the group of affine linear transformations $x \to \alpha x + \beta$, where $\alpha, \beta, x \in \mathbb{F}_q$, $\alpha \ne 0$). Let B denote the subgroup generated by b. Note that $B \lhd X$ and B is cyclic of order q. Moreover, $X/B \cong \mathbb{Z}_p$. Let us view U, W as subgroups of the group $L = \mathbb{Z}_p^n$. Now we have

a natural surjective homomorphism $\varphi : X^n \to L$ with kernel B^n.
Claim The codes U and W are equivalent if and only if the permutation groups $G := \varphi^{-1}(U) \leq X^n$ and $H := \varphi^{-1}(W) \leq X^n$ are permutationally isomorphic.

This follows from the following well-known fact (cf. [16, Prop. 1.3]).

FACT 7.2. *If $\chi \in \operatorname{Aut} X$ then $\chi(aB) = aB$.*

If $p = 2$, we can choose $q = 3$ so $X = \mathbb{Z}_q \rtimes \mathbb{Z}_p \cong S_3$. □

COROLLARY 7.2. *There is a Karp-reduction from Graph Isomorphism to permutational isomorphism of permutation groups with orbit length 3.*

Proof. Graph isomorphism reduces to equivalence of binary linear codes [28]. Combine this with the preceding result. □

7.4 Complexity of permutational isomorphism of permutation groups.

PROPOSITION 7.3. *Permutational isomorphism of permutation groups is in NP.*

Proof. This is an immediate corollary to Proposition 3.1. □

We note that isomorphism of permutation groups is also in NP, for analogous reasons, as pointed out by Luks, cf. [5, Cor. 4.1].

PROPOSITION 7.4. *Permutational isomorphism of permutation groups is in coAM.*

Proof. We sketch a private-coin protocol; then the stated result follows by [18] and [3].

Let $G_1 \leq S - n$ and $G_2 \leq S_n$ be the two permutation groups, acting on the set $\{1, \ldots, n\}$. The Verifier flips a fair coin to select $i \in \{1, 2\}$; picks a random permutation $\sigma \in S_n$; and selects $10n$ elements of $\sigma^{-1} G_i \sigma$ uniformly at random. The Verifier reveals these $10n$ elements to the Prover. The Prover guesses i. The Verifier accepts if either the selected $10n$ elements do not generate $\sigma^{-1} G_i \sigma$ or the Prover's guess at i is correct, otherwise rejects.
Claim: If G_1 and G_2 are permutationally isomorphic then any Prover has $1/2 + o(1)$ probability of success; if G_1 and G_2 are not permutationally isomorphic, then an optimal Prove always succeeds.

The proof is based on the observation that with high probability, the $10n$ elements selected generate $\sigma^{-1} G_i \sigma$, a consequence of the fact that S_n has no subgroup chain of length $2n$ [4]. If they do generate $\sigma^{-1} G_i \sigma$ and G_1 and G_2 are permutationally isomorphic then the Prover receives the $10n$ permutations from the exact same distribution. □

COROLLARY 7.3. *Permutational isomorphism of permutation groups is not NP-complete, unless the polynomial-time hierarchy collapses to the second level.*

Proof. This follows along the lines of the the proof that graph isomorphism is not NP-complete unless the polynomial-time hierarchy collapses [17] (cf. [13] for a full proof).

7.5 Relationship to the Babai-Beals filtration.

Our main algorithm was motivated by the following chain of characteristic subgroups, introduced by Babai and Beals [8] and since used extensively in the algorithmic theory of matrix groups and black-box groups (see [9]):

$$1 \leq \operatorname{Rad}(G) \leq \operatorname{Soc}^*(G) \leq \operatorname{Pker}(G) \leq G.$$

We now explain the terms of this chain. Recall that $\operatorname{Rad}(G)$, the *solvable radical,*, is the unique maximal solvable normal subgroup of G. $\operatorname{Soc}^*(G)$ is the preimage of the socle $\operatorname{Soc}(G/\operatorname{Rad}(G))$ under the natural projection $G \to G/\operatorname{Rad}(G)$.

Note that the group $\operatorname{Soc}^*(G)/\operatorname{Rad}(G) = \operatorname{Soc}(G/\operatorname{Rad}(G))$ is the direct product of simple groups $T_1, \ldots T_k$. The group G acts by conjugation on $\operatorname{Soc}(G/\operatorname{Rad}(G))$; this action permutes the k simple groups involved, so we obtain a homomorphism $G \to S_k$. We denote by $\operatorname{Pker}(G)$ the kernel of this homomorphism (permutation representation).

In a sense, this normal structure provides a layering to the group isomorphism problem; the layers are

1st layer $G/\operatorname{Pker}(G)$: a permutation group of logarithmic degree;

2nd layer $\operatorname{Pker}(G)/\operatorname{Soc}^*(G)$, a solvable group satisfying strong structural constraints;

3rd layer $\operatorname{Soc}^*(G)/\operatorname{Rad}(G) = \operatorname{Soc}(G/\operatorname{Rad}(G))$, a direct product of non-abelian simple groups;

4th layer $\operatorname{Rad}(G)$, a solvable group,

While it is by no means the case that solving the isomorphism problem for the layers would automatically solve it for the entire group, solving it for the layers is definitely a prerequisite. (This statement was formalized for the top layer in Proposition 7.1.) Then the task remains to control the "glue" that holds these layers together.

The bottom layer is a solvable group and testing isomorphism in polynomial time for solvable groups remains elusive (they include the notorious class-2 nilpotent groups). In this paper we considered semisimple groups only, i. e., we assumed $\mathrm{Rad}(G)$ is trivial and therefore $\mathrm{Soc}^*(G) = \mathrm{Soc}(G)$ is a direct product of nonabelian simple groups.

Recall that the isomorphism problem for direct products of simple groups (third layer) is easily solved in polynomial time (Proposition 2.1).

The second layer ("outer automorphism layer") is solvable but is no cause for panic; we glue it right to the second layer. So we are considering semisimple groups G satisfying $G = \mathrm{Pker}(G)$.

Observation. $G = \mathrm{Pker}(G)$ if and only if every minimal normal subgroup of G is simple.

In the terminology of Section 1.3, this is equivalent to saying that $t(G) = 1$. So the isomorphism problem for this class of groups is solved in polynomial time by Theorem 1.2.

The following observation shows that the top layer is a permutation group of logarithmic degree.

Fact 7.3. *A semisimple group of order n is the extension of a semisimple group K with $t(K) = 1$ by a permutation group of degree $\leq \log n / \log 60$.*

Proof. Let G be semisimple of order n and let k denote the number of simple factors of $\mathrm{Soc}(G)$. Since every nonabelian simple group has order ≥ 60, we see that $k \leq \log |\mathrm{Soc}(G)| / \log 60$. Moreover, $t(\mathrm{Pker}(G)) = 1$, and $G/\mathrm{Pker}(G) \hookrightarrow S_k$. □

Proposition 7.1 shows that the top layer of a semisimple group is an arbitrary permutation group of logarithmic degree.

7.6 Caveat about the glue. While we believe that solving permutational isomorphism of $K, L \leq S_k$ in time polynomial in $|K|$ and 2^k will be a significant step toward testing isomorphism of semisimple groups, such a result alone will not automatically suffice, at least not for the following straightforward strategy.

For two semisimple groups G and H, suppose $\mathrm{Soc}(G) = \mathrm{Soc}(H) = T^r$, for some nonabelian simple group T. Recall that $G/\mathrm{Pker}(G)$ and $H/\mathrm{Pker}(H)$ can be embedded in S_r, so we can form the set of permutational isomorphisms $\mathrm{PISO} = \mathrm{PISO}(G/\mathrm{Pker}(G), H/\mathrm{Pker}(H))$. On the other hand, by embedding $\mathrm{Pker}(G)$ and $\mathrm{Pker}(H)$ in $\mathrm{Aut}(T)^k$ we can form the set of code isomorphisms $\mathrm{EQ} = \mathrm{EQ}(\overline{\mathrm{Pker}(G)}, \overline{\mathrm{Pker}(H)})$. Note that PISO and EQ are both either empty or cosets of S_r. One may think that to test isomorphism of G and H, it is enough to test if $\mathrm{PISO} \cap \mathrm{EQ}$ is empty or not. If this idea worked, it would imply that the only remaining bottleneck is to solve permutational isomorphism in time, polynomial in $|G/\mathrm{Pker}(G)|$ and 2^r. (Recall that coset intersection can be done in moderately exponential time.) Here we show a counterexample to this idea.

Proposition 7.5. *There exist groups G and H such that $\mathrm{PISO} \cap \mathrm{EQ} \neq \emptyset$ while $G \ncong H$.*

Proof. Let $k \geq 5$ and let $T_1 = T_2 = A_k$, $R_1 = A_k \times C_2$ where C_2 is the cyclic group of order 2; and let $R_2 = S_k$. Form $G_i = (T_1 \times T_2) \rtimes R_i$, for $i = 1, 2$. Note that both R_1 and R_2 have a copy T_3 of A_k as a normal subgroup of index 2. We define the action of the R_i on $T_1 \times T_2$ as follows: T_3 acts on $T_1 \times T_2$ trivially; and the generator of R_i/T_3 switches T_1 and T_2. It follows that $\mathrm{Pker}(G_i) = \mathrm{Soc}(G_i) = T_1 \times T_2 \times T_3$ $(i = 1, 2)$. Now $G_i/\mathrm{Pker}(G_i) = \langle (1,2) \rangle \leq S_3$, so $\mathrm{PISO} = \langle (1,2) \rangle \leq S_3$, and the isomorphisms between $\mathrm{Pker}(G_1)$ and $\mathrm{Pker}(G_2)$ induce S_3 on the set $\{1, 2, 3\}$ of indices. Thus for G_1 and G_2 we have $\mathrm{PISO} \cap \mathrm{EQ} \neq \emptyset$, while it is clear that $G_1 \ncong G_2$. □

7.7 Open questions. Deciding isomorphism of groups (given by Cayley tables) in polynomial time remains elusive. We propose to solve this problem in polynomial time for semisimple groups. The main question along the way is permutational isomorphism of permutation groups of degree k in time, polynomial in 2^k and the order of the groups. In fact, for permutation groups given by lists of generators, a time bound polynomial in 2^k should be achievable, regardless of the order of the permutation groups.

The following goals also seem realistic.

Problem. Decide isomorphism of groups satisfying $\mathrm{Rad}(G) = Z(G)$ in polynomial time.

Theorem 7.1 determines equivalence of linear codes of length n, given by generator matrices, in time $(2 + o(1))^n$ (assuming field operations at unit cost).

Problem. Decide equivalence of linear codes of length n, given by generator matrices, in time $\exp(\widetilde{O}(\sqrt{n}))$ (assuming field operations at unit cost).

Problem. Decide equivalence of group codes in time $(2 + o(1))^n$.

We are unable to achieve this bound even if the group is cyclic or elementary abelian.

References

[1] V. Arvind. Personal communication, 2010.

[2] László Babai. Permutation groups, coherent configurations, and graph isomorphism. April 1983. D.Sc. Thesis, Hungarian Academy of Sci. (Hungarian).

[3] László Babai. Trading group theory for randomness. In *17th STOC*, pages 421–429. ACM Press, 1985.

[4] Lászlo Babai. On the length of subgroup chains in the symmetric group. *Communications in Algebra*, 14:1729–1736, 1986.

[5] László Babai. Bounded round interactive proofs in finite groups. *SIAM J. Discr. Math.*, 5:88–111, 1992.

[6] László Babai. Equivalence of linear codes. 2010. Unpublished manuscript.

[7] László Babai. Coset intersection in moderately exponential time. *Chicago J. Theoret. Comp. Sci.*, to appear.

[8] László Babai and Robert Beals. A polynomial-time theory of black-box groups I. In C. M. Campbell, E. F. Robertson, N. Ruskuc, and G. C. Smith, editors, *Groups St Andrews 1997 in Bath, I*, volume 260 of *London Math. Soc. Lect. Notes*, pages 30–64. Cambr. U. Press, 1999.

[9] László Babai, Robert Beals, and Ákos Seress. Polynomial-time theory of matrix groups. In *41st ACM STOC*, pages 55–64. ACM Press, 2009.

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

[11] László Babai and Eugene M. Luks. Canonical labeling of graphs. In *Proc. 15th ACM STOC*, pages 171–183. ACM Press, 1983.

[12] László Babai, Eugene M. Luks, and Ákos Seress. Permutation groups in NC. In *Proc. 19th ACM STOC*, pages 409–420. ACM Press, 1987.

[13] László Babai and Shlomo Moran. Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity classes. *J. Computer and Sys. Sci.*, 36:254–276, 1988.

[14] John Horton Conway, Robert Turner Curtis, Simon Phillips Norton, Richard A. Parker, and Robert Arnott Wilson. *Atlas of Finite Groups: Maximal Subgroups and Ordinary Characters for Simple Groups*. Oxford University Press, 1985.

[15] Merrick L. Furst, John Hopcroft, and Eugene M. Luks. Polynomial-time algorithms for permutation groups. In *Proc. 21st FOCS*, pages 36–41. IEEE Comp. Soc., 1980.

[16] Marek Golasiński and Daciberg Lima Gonçalves. Spherical space forms - homotopy types and self-equivalences for groups. *Topology and Appl.*, 146-147:451–470, 2005.

[17] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity, or all languages in np have zero-knowledge proof systems. *J. ACM*, 38:690–728, 1991.

[18] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In Silvio Micali, editor, *Randomness and Computation*, 1989.

[19] Telikepalli Kavitha. Linear time algorithms for abelian group isomorphism and related problems. *J. Comput. Syst. Sci.*, 73(6):986–996, 2007.

[20] Neeraj Kayal and Timur Nezhmetdinov. Factoring groups efficiently. In *ICALP '09: Proceedings of the 36th International Colloquium on Automata, Languages and Programming*, pages 585–596. Springer-Verlag, 2009. Also availabe as ECCC Tech Report TR08-074.

[21] Donald E. Knuth. Efficient representation of perm groups. *Combinatorica*, 11:57–68, 1991.

[22] François Le Gall. Efficient isomorphism testing for a class of group extensions. In *STACS*, pages 625–636, 2009.

[23] U. V. Linnik. On the least prime in an arithmetic progression. I. The basic theorem. *Rec. Math. [Mat. Sbornik] N.S.*, 15(57):139–178, 1944.

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

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

[26] Eugene M. Luks. Hypergraph isomorphism and structural equivalence of boolean functions. In *Proc. 31st ACM STOC*, pages 652–658. ACM Press, 1999.

[27] Gary L. Miller. On the $n \log n$ isomorphism technique (a preliminary report). In *Proc. 10th ACM STOC*, pages 51–58, New York, NY, USA, 1978. ACM Press.

[28] Erez Petrank and Ron M. Roth. Is code equivalence easy to decide? *IEEE Transactions on Information Theory*, 43:1602–1604, 1997.

[29] Derek J.S. Robinson. *A Course in the Theory of Groups*. Springer, 2nd edition, 1996.

[30] Carla Savage. An $O(n^2)$ algorithm for abelian group isomorphism. Technical report, North Carolina State University, 1980.

[31] Ákos Seress. *Permutation Group Algorithms*. Cambridge University Press, 2003.

[32] Charles C. Sims. Computation with permutation groups. In S. R. Petrick, editor, *Proc. 2nd Symp. Symb. Algeb. Manip.*, pages 23–28. ACM Press, 1971.

[33] M. Suzuki. *Group Theory I, II*. Springer, 1982, 1986.

[34] Narayan Vikas. An $O(n)$ algorithm for abelian p-group isomorphism and an $O(n \log n)$ algorithm for abelian group isomorphism. *J. Comput. Syst. Sci.*, 53(1):1–9, 1996.

[35] James B. Wilson. Decomposing p-groups via Jordan algebras. *J. Algebra*, 322:2642–2679, 2009.

[36] James B. Wilson. Finding central decompositions of p-groups. *J. Group Theory*, 12:813–830, 2009.

[37] James B. Wilson. Finding direct product decompositions in polynomial time. 2010. Submitted for publication. Available as arXiv e-print 1005.0548.