Representing groups on graphs

Sagarmoy Dutta and Piyush P Kurur
Department of Computer Science and Engineering,
Indian Institute of Technology Kanpur,
Kanpur, Uttar Pradesh, India 208016
{sagarmoy,ppk}@cse.iitk.ac.in

Abstract

In this paper we formulate and study the problem of representing groups on graphs. We show that with respect to polynomial time turing reducibility, both abelian and solvable group representability are all equivalent to graph isomorphism, even when the group is presented as a permutation group via generators. On the other hand, the representability problem for general groups on trees is equivalent to checking, given a group G and n, whether a nontrivial homomorphism from G to S_n exists. There does not seem to be a polynomial time algorithm for this problem, in spite of the fact that tree isomorphism has polynomial time algorithm.

1 Introduction

Representation theory of groups is a vast and successful branch of mathematics with applications ranging from fundamental physics to computer graphics and coding theory [5]. Recently representation theory has seen quite a few applications in computer science as well. In this article, we study some of the questions related to representation of finite groups on graphs.

A representation of a group G usually means a linear representation, i.e. a homomorphism from the group G to the group $\mathrm{GL}(V)$ of invertible linear transformations on a vector space V. Notice that $\mathrm{GL}(V)$ is the set of *symmetries* or *automorphisms* of the vector space V. In general, by a representation of G on an object X, we mean a homomorphism from G to the automorphism group of X. In this article, we study some computational problems that arise in the representation of *finite groups* on graphs. Our interest is the following group representability problem: Given a group G and a graph X, decide whether G has a nontrivial representation on X. As expected this problem is closely connected to graph isomorphism: We show, for example, that the graph isomorphism problem reduces to representability of abelian groups. In the other direction we show that even for solvable groups the representability on graphs is decidable using a graph isomorphism oracle. The reductions hold true even when the groups are presented as permutation groups. One might be tempted to conjecture that the problem is equivalent to Graph Isomorphism. However we conjecture that this might not be the case. The non-solvable version of this problem seems to be harder than graph isomorphism. For example, we were able to show that representability of groups on trees, a class of graphs for which isomorphism is decidable in polynomial time, is as hard as checking whether, given an integer n and a group G, the symmetric group S_n has a nontrivial subgroup homomorphic to G, a problem for which no polynomial time algorithm is known.

2   Background

In this section we review the group theory required for the rest of the article. Any standard text book on group theory, for example the one by Hall [4], will contain the required results.

We use the following standard notation: The identity of a group G is denoted by 1. In addition 1 also stands for the singleton group consisting of only the identity. For groups G and H, $H \leq G$ (or $G \geq H$) means that H is a subgroup of G. Similarly by $H \unlhd G$ (or $G \unrhd H$) we mean H is a *normal subgroup* of G.

Let G be any group and let x and y be any two elements. By the *commutator* of x and y, denoted by $[x, y]$, we mean $xyx^{-1}y^{-1}$. The *commutator subgroup* of G is the group generated by the set $\{[x, y] | x, y \in G\}$. We denote the commutator subgroup of G by G'. The following is a well known result in group theory [4, Theorem 9.2.1]

Theorem 2.1. *The commutator subgroup G' is a normal subgroup of G and G/G' is abelian. Further for any normal subgroup N of G such that G/N is abelian, N contains G' as a subgroup.*

A group is *abelian* if it is commutative, i.e. $gh = hg$ for all group elements g and h. A group G is said to be *solvable* [4, Page 138] if there exists a decreasing chain of groups $G = G_0 \rhd G_1 \ldots \rhd G_t = 1$ such that G_{i+1} is the commutator subgroup of G_i for all $0 \leq i < t$.

An important class of groups that play a crucial role in graph isomorphism and related problems are permutation groups. We follow the notation of Wielandt [11] for permutation groups. Let Ω be a finite set. The *symmetric group* on Ω, denoted by $\text{Sym}(\Omega)$, is the group of all permutations on the set Ω. By a *permutation group* on Ω we mean a subgroup of the symmetric group $\text{Sym}(\Omega)$. For any positive integer n, we will use S_n to denote the symmetric group on $\{1, \ldots, n\}$. Let g be a permutation on Ω and let α be an element of Ω. The image of α under g will be denoted by α^g. For a permutation group G on Ω, the orbit of α is denoted by α^G. Similarly if Δ be a subset of Ω then Δ^g denotes the set $\{\alpha^g | \alpha \in \Delta\}$.

Any permutation group G on n symbols has a generating set of size at most n. Thus for computational tasks involving permutation groups it is assumed that the group is presented to the algorithm via a small generating set. As a result, by efficient algorithms for permutation groups on n symbols we mean algorithms that take time polynomial in the size of the generating set and n.

Let G be a subgroup of S_n and let $G^{(i)}$ denote the subgroup of G that fixes pointwise $j \leq i$, i.e. $G^{(i)} = \{g | j^g = j, 1 \leq j \leq i\}$. Let C_i denote a right *transversal*, i.e. the set of right coset representative, for $G^{(i)}$ in $G^{(i-1)}$. The $\cup_i C_i$ is a generating set for G and is called the *strong generating set* for G. The corner stone for most polynomial time algorithms for permutation group is the Schreier-Sims [9, 10, 3] algorithm for computing the *strong generating set* of a permutation group G given an arbitrary generating set. Once the strong generating set is computed, many natural problems for permutation groups can be solved efficiently. We give a list of them in the next theorem.

Theorem 2.2. *Given a generating set for G there are polynomial time algorithms for the following task.*

1. *Computing the strong generating set.*

2. *Computing the order of G.*

By a graph we mean a *finite undirected graph*. For a graph X, $V(X)$ and $E(X)$ denotes the set of vertices and edges respectively and $\text{Aut}(X)$ denotes the group of all automorphisms of X, i.e. permutations on $V(X)$ that maps edges to edges and non-edges to non-edges.

Definition 2.3 (Representation). *A representation ρ from a group G to a graph X is a homomorphism from G to the automorphism group $\text{Aut}\,(X)$ of X.*

Alternatively we say that G acts on (the right) of X via the representation ρ. When ρ is understood, we use u^g to denoted $u^{\rho(g)}$.

A representation ρ is *trivial* if all the elements of G are mapped to the identity permutation. A representation ρ is said to be *faithful* if it is an injection as well. Under a faithful action G can be thought of as a subgroup of the automorphism group. We say that G is *representable* on X if there is a nontrivial representation from G to X. We now define the following natural computational problem.

Definition 2.4 (Group representability problem). *Given a group G and a graph X decide whether G is representable on X nontrivially.*

We will look at various restrictions of the above problem. For example, we study the abelian (solvable) group representability problem where our input groups are abelian (solvable). We also study the group representability problem on trees, by which we mean group representability where the input graph is a tree.

Depending on how the group is presented to the algorithm, the complexity of the problem changes. One possible way to present G is to present it as a permutation group on m symbols via a generating set. In this case the input size is $m + \#V(X)$. On the other hand, we can make the task of the algorithm easier by presenting the group via a multiplication table. In this paper we mostly assume that the group is in fact presented via its multiplication table. Thus polynomial time means polynomial in $\#G$ and $\#V(X)$. However for solvable representability problem, our results extend to the case when G is a permutation group presented via a set of generators.

We now look at the following closely related problem that occurs when we study the representability of groups on trees.

Definition 2.5 (Permutation representability problem). *Given a group G and an integer n in unary, check whether there is a homomorphism from G to S_n.*

Overview of the results

Our first result is to show that graph isomorphism reduces to abelian representability problem. In fact we show that graph isomorphism reduces to the representability of prime order cyclic groups on graphs. Next we show that solvable group representability problem reduces to graph isomorphism problem. Thus as far as polynomial time Turing reducibility is concerned abelian group representability and solvable group representability are all equivalent to graph isomorphism. As a corollary we have, solvable group representability on say bounded degree graphs or bounded genus graphs are all in polynomial time.

We then show that group representability on trees is equivalent to permutation representability (Definition 2.5). This is in contrast to the corresponding isomorphism problem because for trees, isomorphism testing is in polynomial time whereas permutation representability problem does not appear to have a polynomial time algorithm.

3   Abelian representability

In this section we prove that the graph isomorphism problem reduces to abelian group representability on graph. Given input graphs X and Y of n vertices each and any prime $p > n$, we construct a graph Z of exactly $p \cdot n$ vertices such that X and Y are isomorphic if and only if the

cyclic group of order p is representable on Z. Since for any integer n there is a prime p between n and $2n$ (Bertrand's conjecture), the above constructions gives us a reduction from the graph isomorphism problem to abelian group representability problem.

For the rest of the section, fix the input graphs X and Y. Our task is to decide whether X and Y are isomorphic. Firstly we assume, without loss of generality, that the graphs X and Y are connected, for otherwise we can take their complement graphs X' and Y', which are connected and are isomorphic if and only if X and Y are isomorphic. Let n be the number of vertices in X and Y and let p be any prime greater than n. Consider the graph Z which is the disjoint union of p connected components $Z_1, \ldots, Z_p$ where, for each $1 \leq i < p$, each Z_i is an isomorphic copy of X and Z_p is an isomorphic copy of Y. First we prove the following lemma.

Lemma 3.1. *If X and Y are isomorphic then $\mathbb{Z}/p\mathbb{Z}$ is representable on Z.*

Proof. Clearly it is sufficient to show that there is an order p automorphism for Z. Let h be an isomorphism from X to Y. For every vertex v in X, let v_i denote its copy in Z_i. Consider the bijection g from $V(Z)$ to itself defined as follows: For all vertices v in $V(X)$ and each $1 \leq i < p-2$, let $v_i^g = v_{i+1}$. Further let g map v_{p-1} to v^h and v^h to v_1. It is easy to verify that g is an automorphism of Z and has order p. $\qquad\square$

We now prove the converse

Lemma 3.2. *If $\mathbb{Z}/p\mathbb{Z}$ can be represented on Z then X and Y are isomorphic.*

Proof. If $\mathbb{Z}/p\mathbb{Z}$ can be represented on Z then there exists a nonidentity automorphism g of Z such that order of g is p. We consider the action of the cyclic group H, generated by g, on $V(X)$. Since g is nontrivial, there exists at least one H-orbit Δ of $V(X)$ which is of cardinality greater than 1. However by orbit stabiliser formula [11, Theorem 3.2], $\#\Delta$ divides $\#H = p$. Since p is prime, Δ should be of cardinality p.

We prove that no two vertices of Δ belong to the same connected component. Assume the contrary and let α and β be two elements of Δ which also belong to the same connected component of Z. There is some $0 < t < p$ such that $\alpha^{g^t} = \beta$. We assume further, without loss of generality, that $t = 1$, for otherwise we replace g by the automorphism g^t, which is also of order p, and carry out the argument. Therefore $\alpha^g = \beta$ lie in the same component of Z. It follows then that, for each $0 \leq i \leq p-1$, the element $\alpha_i = \alpha^{g^i}$ is in the same component of Z, as automorphisms preserve edges and hence paths. However this means that there is a component of Z that is of cardinality at least p. This is a contradiction as each component of Z has at most $n < p$ vertices as they are copies of either X or Y.

It follows that there is some $1 \leq i < p$, for which g must map at least one vertex of the component Z_i to some vertex of Z_p. As a result the automorphism g maps the entire component Z_i to Z_p. Therefore the components Z_i and Z_p are isomorphic and so are their isomorphic copies X and Y. $\qquad\square$

Given two graphs X and Y of n vertices we find a prime p such that $n < p < 2n$, construct the graph Z and construct the multiplication table for $\mathbb{Z}/p\mathbb{Z}$. This requires only logarithmic space in n. Using Lemmas 3.1 and 3.2 we have the desired reduction.

Theorem 3.3. *The graph isomorphism problem logspace many-one reduces to abelian group representability problem.*

4 Solvable representability problem

In the previous section we proved that abelian group representability is at least as hard as graph isomorphism. In this section we show that solvable group representability is polynomial time Turing reducible to the graph isomorphism problem. We claim that a solvable group G is representable on X if and only if $\#\mathrm{Aut}\,(X)$ and $\#G/G'$ have a common prime factor, where G' is commutator subgroup of G. We do this in two stages.

Lemma 4.1. *A solvable group G can be represented on a graph X if $\#G/G'$ and $\#\mathrm{Aut}\,(X)$ have a common prime factor.*

Proof. Firstly notice that it suffices to prove that there is a nontrivial homomorphism, say ρ, from G/G' to $\mathrm{Aut}\,(X)$. A nontrivial representation for G can be obtained by composing the natural quotient homomorphism from G *onto* G/G' with ρ.

Recall that the quotient group G/G' is an abelian group and hence can be represented on X if for some prime p that divides $\#G/G'$, there is an order p automorphism for X. However by the assumption of the theorem, there is a common prime factor, say p, of $\#G/G'$ and $\#\mathrm{Aut}\,(X)$. Therefore, by Cayley's theorem there is an order p element in $\mathrm{Aut}\,(X)$. As a result, G/G' and hence G is representable on X. $\qquad\square$

To prove the converse, for the rest of the section fix the input, the solvable group G and the graph X. Consider any nontrivial homomorphism ρ from the group G to $\mathrm{Aut}\,(X)$. Let $H \leq \mathrm{Aut}\,(X)$ denote the image of the group G under ρ. We will from now on consider ρ as an automorphism from G *onto* H. Since the subgroup H is the homomorphic image of G, H itself is a solvable group.

Lemma 4.2. *The homomorphism ρ maps the commutator subgroup G' of G onto the commutator subgroup H'.*

Proof. First we prove that $\rho(G') \leq H'$. For this notice that for all x and y in G, since ρ is a homomorphism, $\rho([x,y]) = [\rho(x), \rho(y)]$ is an element of H'. As G' is generated by the set $\{[x,y]|x,y \in G\}$ of all commutators, $\rho(G') \leq H'$. To prove the converse notice that ρ is a surjection on H. Therefore for any element h of H, we have element x_h of G such that $\rho(x_h) = h$. Consider the commutator $[g,h]$ for any two elements g and h of H. We have $\rho([x_g, x_h]) = [g,h]$. This proves that all the commutators of H are in the image of G' and hence $\rho(G') \geq H'$. $\qquad\square$

We have the following result about solvable groups that directly follows from the definition of solvable groups [4, Page 138].

Lemma 4.3. *Let G be any nontrivial solvable group then its commutator subgroup G' is a strict subgroup of G.*

Proof. By the definition of solvable groups, there exist a chain $G = G_0 \rhd G_1 \ldots \rhd G_t = 1$ such that G_{i+1} is the commutator subgroup of G_i for all $0 \leq i < t$. If $G = G' = G_1$ then $G = G_i$ for all $0 \leq i \leq t$ implying $G = 1$ $\qquad\square$

We are now ready to prove the converse of Lemma 4.1.

Lemma 4.4. *Let G be any solvable group and let X be any graph. The orders $\#G/G'$ and $\#\mathrm{Aut}\,(X)$ have a common prime factor if G is representable on graph X.*

Proof. Let ρ be any nontrivial homomorphism from G to $\text{Aut}(X)$, and let H be the image of group G under this homomorphism. Since the commutator subgroup G' is strictly contained in the group G (Lemma 4.3), order of the quotient group $\#G/G' > 1$. Furthermore, the image group H itself is solvable and nontrivial, as it is the image of a solvable group G under a nontrivial homomorphism. Therefore, the commutator subgroup H' is strictly contained in H implying $\#H/\#H' > 1$.

Consider the homomorphism $\tilde{\rho}$ from G *onto* H/H' defined as $\tilde{\rho}(g) = \rho(g)H'$. Since ρ maps G' onto H', we have that G' is in the kernel of $\tilde{\rho}$. Therefore, $\tilde{\rho}$ can be *refined* to a map from G/G' *onto* H/H'. Clearly the prime factors of $\#H/H'$ are all prime factors of $\#G/G'$. However, any prime factor of $\#H/H'$ is a prime factor of $\text{Aut}(X)$, as both H and H' are subgroups of $\text{Aut}(X)$. Therefore, the orders of G/G' and $\text{Aut}(X)$ have a common prime factor. $\square$

The order of the automorphism group of the input graph X can be computed in polynomial time using an oracle to the graph isomorphism problem [7]. Further since the automorphism group is a subgroup of S_n, where n is the cardinality of $V(X)$, all its prime factors are less than n and hence can be determined. Also since G is given as a table, its commutator subgroup G' can be computed in polynomial time and the prime factors of $\#G/G'$ can also be similarly determined. Therefore we can easily check, given the group G via its multiplication table and the graph X, whether the order of the quotient group G/G' has common factors with the order of $\text{Aut}(X)$. We thus have the following theorem.

Theorem 4.5. *The problem of deciding whether a solvable group can be represented on a given graph reduces to graph isomorphism problem.*

For the reduction in the above theorem to work, it is sufficient to compute the order of G and its commutator subgroup G'. This can be done even when the group G is presented as a permutation group on m symbols via a generating set. To compute $\#G$ we can compute the strong generating set of G and use Theorem 2.2. Further given a generating set for G, a generating set for its commutator subgroup G' can be compute in polynomial time [3, Theorem 4]. Therefore, the order of G/G' can be computed in polynomial time given the generating set for G. Furthermore, G and G' are subgroups of S_m and hence all their prime factors are less than m and can be determined. We can then check whether $\#G/G'$ has any common prime factors with $\#\text{Aut}(X)$ just as before using the graph isomorphism oracle. Thus we have the following theorem.

Theorem 4.6. *The solvable group representability problem, where the group is presented as a permutation group via a generating set, reduces to the graph isomorphism problem via polynomial time Turing reduction.*

5   Representation on tree

In this section we study the representation of groups on trees. It is known that isomorphism of trees can be tested in polynomial time [2]. However we show that the group representability problem over trees is equivalent to permutation representability problem (Definition 2.5), a problem for which, we believe, there is no polynomial time algorithm.

Firstly, to show that permutation representability problem is reducible to group representability problem on trees, it is sufficient to construct, given and integer n, a tree whose automorphism group is S_n. Clearly a tree with n leaves, all of which is connected to the root, gives such a tree (see Figure 1). Therefore we have the following lemma.

Lemma 5.1. *Permutation representability reduces to representability on tree.*

Figure 1: Tree with automorphism group S_n

To prove the converse, we first reduce the group representability problem on an arbitrary tree to the problem of representability on a rooted tree. We then do a divide and conquer on the structure of the rooted tree using the permutation representability oracle. The main idea behind this reduction is Lemma 5.5 where we show that for any tree T, either there is a vertex which is fixed by all automorphism, in which case we can choose this vertex as the root, or there are two vertices α and β connected by an edge which together forms an orbit under the action of $\mathrm{Aut}(T)$, in which case we can add a dummy root (see Figure 2) to make it a rooted tree without changing the automorphism group.
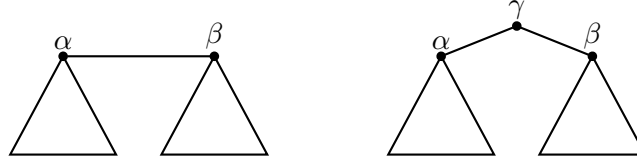

Figure 2: Minimal orbit has two elements

For the rest of the section fix a tree T. Let Δ be an orbit in the action of $\mathrm{Aut}(T)$ on $V(T)$. We define the graph T_Δ as follows: A vertex γ (or edge e) of T belongs to T_Δ if there are two vertices α and β in Δ such that γ (or e) is contained in the path from α to β. It is easy to see that T_Δ contains paths between any two vertices of Δ. Any vertex in T_Δ is connected to some vertex in Δ and all vertices in Δ are connected in T_Δ which implies T_Δ is connected. Furthermore T_Δ has no cycle, as its edge set is a subset of the edge set of T. Therefore T_Δ is a tree.

Lemma 5.2. *Let g be any automorphism of T and consider any vertex γ (or edge e) of T_Δ. Then the vertex γ^g (or edge e^g) is also in T_Δ.*

Proof. Since γ (or e) is present in T_Δ, there exists α and β in Δ such that γ (or e) is in the path between α and β. Also since automorphisms preserve paths, γ^g (or e^g) is in the path from α^g to β^g. $\qquad\square$

Lemma 5.3. *The orbit Δ is precisely the set of leaves of T_Δ.*

Proof. First we show that all leaf nodes of T_Δ are in orbit Δ. Any node α of T_Δ must lie on a path such that the endpoints are in orbit Δ. If α is a leaf of T_Δ, this can only happen when α itself is in Δ.

We will prove the converse by contradiction. If possible let α be a vertex in the orbit Δ which is not a leaf of T_Δ. Vertex α must lie on the path between two leaves β and γ. Also since β and γ are leaves of T_Δ, they are in the orbit Δ.

Let g be an automorphism of T which maps α to β. Such an automorphism exists because α and β are in the same orbit Δ. The image $\alpha^g = \beta$ must lie on the path between β^g and γ^g and neither β^g or γ^g is β. This is impossible because β is a leaf of T_Δ. $\qquad\square$

Lemma 5.4. *Let γ be a vertex in orbit Σ. If γ is a vertex of the subtree T_Δ then subtree T_Σ is a subtree of T_Δ.*

Proof. Assume that Δ is different from Σ, for otherwise the proof is trivial. First we show that all the vertices of Σ are vertices of T_Δ. The vertex γ lies on a path between two vertices of Δ, say α and β. Take any vertex γ' from the orbit Σ. There is an automorphism g of T which maps γ to γ'. Now $\gamma' = \gamma^g$ lies on the path between α^g and β^g and hence is in the tree T_Δ.

Consider any edge e of T_Σ. There exists γ_1 and γ_2 of Σ such that e is on the path from γ_1 to γ_2. By previous argument, T contains γ_1 and γ_2. Since T is a tree, this path is unique and any subgraph of T, in which γ_1 and γ_2 are connected, must contain this path. Hence T_Δ contains e. $\qquad\square$

Lemma 5.5. *Let T be any tree then either there exists a vertex α that is fixed by all the automorphisms of T or there exists two vertices α and β connected via an edge e such that $\{\alpha, \beta\}$ is an orbit of $\mathrm{Aut}\,(T)$. In the latter case every automorphism maps e to itself.*

Proof. Consider the following partial order between orbits of $\mathrm{Aut}\,(T)$: $\Sigma \leq \Delta$ if T_Σ is a subtree of T_Δ. The relation $\leq$ is clearly a partial order because the "subtree" relation is. Since there are finitely many orbits there is always a minimal orbit under the above ordering. From Lemmas 5.3 and 5.4 it follows that for an orbit Δ, if Σ is the orbit containing an internal node γ of T_Δ then Σ is strictly less than Δ. Therefore for any minimal orbit Δ, all the nodes are leaves. This is possible if either T_Δ is a singleton vertex α, or consists of exactly two nodes connected via an edge. In the former case all automorphisms of T have to fix α, whereas in the latter case the two nodes may be flipped but the edge connecting them has to be mapped to itself. $\qquad\square$

It follows from Lemma 5.5 that any tree T can be rooted, either at a vertex or at an edge with out changing the automorphism. Given a tree T, since computing the generating set for $\mathrm{Aut}\,(T)$ can be done in polynomial time, we can determine all the orbits of $\mathrm{Aut}\,(T)$ by a simple transitive closure algorithm. Having computed these orbits, we determine whether T has singleton orbit or an orbit of cardinality 2. For trees with an orbit containing a single vertex α, rooting the tree at α does not change the automorphism group. On the other hand if the tree has an orbit with two elements we can add a dummy root as in Figure 2 without changing the automorphism group. Since by Lemma 5.5 these are the only two possibilities we have the following theorem.

Theorem 5.6. *There is a polynomial time algorithm that, given as input a tree T, outputs a rooted tree T' such that for any group G, G is representable on T if and only if G is representable on the rooted tree T'.*

For the rest of the section by a tree we mean a rooted tree. We will prove the reduction from representability on rooted trees to permutation representability. First we characterise the automorphism group of a tree in terms of wreath product [Theorem 5.9] and then show that we can find a nontrivial homomorphism, if there exists one, from the given group G to this automorphism group by querying a permutation representability oracle.

Definition 5.7 (Semidirect product and wreath product)**.** *Let G and A be any two group and let φ be any homomorphism from G to $\mathrm{Aut}\,(A)$, then the semi-direct product $G \ltimes_\varphi A$ is the group whose underlying set is $G \times A$ and the multiplication is defined as $(g, a)(h, b) = (gh, a^{\varphi(h)} b)$.*

We use $W_n(A)$ to denote the wreath product $S_n \wr A$ which is the semidirect product $S_n \ltimes_\varphi A^n$, where A^n is the n-fold direct product of A and $\varphi(h)$, for each h in S_n, permutes $\mathbf{a} \in A^n$ according to the permutation h, i.e. maps $(\ldots, a_i, \ldots) \in A^n$ to $(\ldots, a_j, \ldots)$ where $j^h = i$.

As the wreath product is a semidirect product, we have the following lemma.

Lemma 5.8. *The wreath product $W_n(A)$ contains (isomorphic copies of) S_n and A^n as subgroups such that A^n is normal and the quotient group $W_n(A)/A^n = S_n$.*

For the rest of the section fix the following: Let T be a tree with root ω with k children. Consider the subtrees of T rooted at each of these k children and partition them such that two subtrees are in the same partition if and only if they are isomorphic. Let t be the number of partitions and let k_i, for $(1 \le i \le t)$, be the number of subtrees in the i-th partition. For each i, pick a representative subtree T_i from the i-th partition and let A_i denote the automorphism group of T_i. The following result is well known but a proof is given for completeness.

Theorem 5.9. *The automorphism group of the tree T is (isomorphic to) the direct product $\prod_{i=1}^{t} W_{k_i}(A_i)$.*

Proof. Let $\omega_1, \ldots, \omega_k$ be the children of the root ω and let X_i denote the subtree rooted at ω_i. We first consider the case when $t = 1$, i.e. all the subtrees X_i are isomorphic. Any automorphism g of T must permute the children ω_i's among themselves and whenever $\omega_i^g = \omega_j$, the entire subtree X_i maps to X_j. As all the subtrees X_i are isomorphic to T_1, the forest $\{X_1, \ldots, X_k\}$ can be thought of as the disjoint union of k copies of the tree T_1 by fixing, for each i, an isomorphism σ_i from T_1 to X_i.

For an automorphism g of T, define the permutation $\tilde{g} \in S_k$ and the automorphisms $a_i(g)$ of T_1 as follows: if $\omega_i^g = \omega_j$ then $i^{\tilde{g}} = j$ and $a_i(g) = \sigma_i g \sigma_j^{-1}$. Consider the map ϕ from $\mathrm{Aut}\,(T)$ to $W_k(A)$ which maps an automorphism g to the group element $(\tilde{g}, a_1(g), \ldots, a_k(g))$ in $W_k(A)$. It is easy to verify that ϕ is the desired isomorphism.

When the number of partitions t is greater than 1, any automorphism of T fixes the root ω and permutes the subtrees in the i-th partition among themselves. Therefore the automorphism group of T is same as the automorphism group of the collection of forests F_i one for each partition i. Each forest is a disjoint union of k_i copies of T_i and we can argue as before that its automorphism group is (isomorphic to) $W_{k_i}(A)$. Therefore $\mathrm{Aut}\,(T)$ should be the direct product $\prod_{i=1}^{t} W_{k_i}(A_i)$. $\square$

Lemma 5.10. *If the group G can be represented on the tree T, then there exists $1 \le i \le t$ such that there is a nontrivial homomorphism from G to $W_{k_i}(A_i)$.*

Proof. If there is a nontrivial homomorphism from a group G to the direct product of groups $H_1, \ldots, H_t$ then for some i, $1 \le i \le t$, there is a nontrivial homomorphism from G to H_i. The lemma then follows from Theorem 5.9. $\square$

Lemma 5.11. *If there is a nontrivial homomorphism ρ from a group G to $W_n(A)$ then there is also a nontrivial homomorphism from G either to S_n or to A.*

Proof. Let ρ be a nontrivial homomorphism G to $W_n(A)$. Since A^n is a normal subgroup of $W_n(A)$ and the quotient group $W_n(A)/A^n$ is S_n, there is a homomorphism ρ' from $W_n(A)$ to S_n with kernel A^n. The composition of ρ and ρ' is a homomorphism from G to S_n.

If $\rho' \cdot \rho$ is trivial then ρ' maps all elements of $\rho(G)$ to identity of S_n. Which imply that $\rho(G)$ is a subgroup of the kernel of ρ', that is A^n. So, ρ is a nontrivial homomorphism from G to A^n. Hence there must be a nontrivial homomorphism from G to A. $\square$

Theorem 5.12. *Given a group G and a rooted tree T with n nodes and an oracle for deciding whether G has a nontrivial homomorphism to S_m for $1 \le m \le n$, it can be decided in polynomial time whether G can be represented on T.*

Proof. If the tree has only one vertex then reject. Otherwise let $t, k_1, \ldots, k_t$ and $A_1, \ldots A_t$ be the quantities as defined in Theorem 5.9. Since there is efficient algorithm to compute tree isomorphism, t and $k_1, \ldots, k_t$ can be computed in polynomial time. If G is representable on T then, by Lemma 5.10 and Lemma 5.11, there is a nontrivial homomorphism from G to either S_{k_i} or A_i for some i. Using the oracle, check whether there is a nontrivial homomorphism to any of the symmetric groups. If found then accept, otherwise for all i, decide whether there is a nontrivial homomorphism to A_i by choosing a subtree T_i from the ith partition and recursively asking whether G is representable on T_i. Total number of recursive calls is bounded by the number of vertices of T. Hence the reduction is polynomial time. $\square$

6 Conclusion

In this paper we studied the group representability problem, a computational problem that is closely related to graph isomorphism. The representability problem could be equivalent to graph isomorphism, but the results of Section 5 give some, albeit weak, evidence that this might not be the case. It would be interesting to know what is the exact complexity of this problem vis a vis the graph isomorphism problem. We know from the work of Mathon [7] that the graph isomorphism problem is equivalent to its functional version where, given two graphs X and Y, we have to compute an isomorphism if there exists one. The functional version of group representability, namely give a group G and a graph X compute a nontrivial representation if it exists, does not appear to be equivalent to the decision version. Also it would be interesting to know if the representability problem shares some of lowness of graph isomorphism [8, 6, 1]. Our hope is that, like the study of group representation in geometry and mathematics, the study of group representability on graphs help us better understand the graph isomorphism problem.

References

[1] V. Arvind and Piyush P Kurur. Graph Isomorphism is in SPP. In 43rd *Annual Symposium of Foundations of Computer Science*, pages 743–750. IEEE, November 2002.

[2] Lázló Babai and Eugene M. Luks. Canonical labeling of graphs. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, pages 171–183, 1983.

[3] Merrick L. Furst, John E. Hopcroft, and Eugene M. Luks. Polynomial-time algorithms for permutation groups. In *IEEE Symposium on Foundations of Computer Science*, pages 36–41, 1980.

[4] Marshall Hall Jr. *The Theory of Groups*. The Macmillan Company, New York, first edition, 1959.

[5] W. D. Joyner. Real world applications of representation theory of non-abelian groups. http://www.usna.edu/Users/math/wdj/repn_thry_appl.htm.

[6] Johannes Köbler, Uwe Schöning, and Jacobo Torán. Graph isomorphism is low for PP. *Computational Complexity*, 2(4):301–330, 1992.

[7] R Mathon. A note on graph isomorphism counting problem. *Information Processing Letters*, 8(3):131–132, 15 March 1979.

[8] Uwe Schöning. Graph isomorphism is in the low hierarchy. In *Symposium on Theoretical Aspects of Computer Science*, pages 114–124, 1987.

[9] C. C. Sims. Computational methods in the study of permutation groups. *Computational problems in Abstract Algebra*, pages 169–183, 1970.

[10] C. C. Sims. Some group theoretic algorithms. *Topics in Algebra*, 697:108–124, 1978.

[11] Helmut Wielandt. *Finite Permutation Groups*. Academic Press, New York, 1964.