GRAPH RIGIDITY*

Richard Cole

TR 82-517
August 1982

Department of Computer Science
Upson Hall
Cornell University
Ithaca, New York 14853

*This research was supported in part by ONR grant N0014-76-C-0018

Abstract

The relationship between graph isomorphism and graph rigidity is studied. Although in general it is not known if these problems are equivalent under polynomial time Turing reductions, equivalence is shown for a subclass of graphs with abelian automorphism groups.

Graph Rigidity

1. *Introduction*

The complexity of graph isomorphism has not been completely determined. It is known to be in NP, is thought not to be NP complete, but is not known to be in P. Recent work has determined several families of graphs for which isomorphism can be decided in deterministic polynomial time. These include graphs of bounded genus [3,8,9,11,16], graphs of bounded valence [12], and graphs of bounded eigenvalue multiplicity [2]. Graph isomorphism is one of the few natural problems thought to lie between P and NP. Thus it is of interest to study its complexity and to consider related problems. Other recent work includes [1,2,6,7,10,13,14,15].

A related problem is finding the automorphism group of a graph, or more precisely, generators for the automorphism group. It is equivalent to testing graph isomorphism under polynomial-time Turing reductions. From generators for the automorphism group, the size of the group can be determined and membership in the group can be tested, both in polynomial time [4]. Another related problem is the graph rigidity problem: determining whether a graph has just one automorphism, namely the identity map. Clearly, this is polynomial-time Turing reducible to the problem of finding the automorphism group of a graph. It is not known whether the converse is true. In this paper we study the relative complexities of testing graph isomorphism, testing graph rigidity, and finding the automorphism group of a graph for a restricted set of graphs.

Our initial aim was to show that testing graph isomorphism and testing graph rigidity were equivalent under polynomial-time Turing reductions. However, this result was proven only for a subclass of the graphs with abelian automorphism groups. The lack of success in proving the stronger result can be taken as further evidence that graph isomorphism is not in P.

In section 2 definitions and notation are given. In section 3 some motivation is provided leading to an outline of the algorithm for finding an automorphism group, and in sections 4

through 7 the algorithm is developed in detail. In section 8 an algorithm for testing isomorphism is described. Both of these algorithms are for restricted sets of graphs.

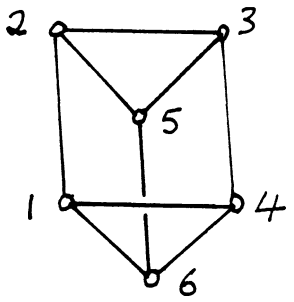
2. *Definitions and Notation*

Throughout this paper, $G = (V,E)$ denotes a graph with edge set E and vertex set V. Some, or all, of the vertices may be colored. A will denote the automorphism group of G. Standard terms from group theory are assumed to be known (Herstein [5] is a useful reference on group theory). The notation $< g_1, g_2, ..., g_k >$ denotes the group generated by the elements $g_1, g_2, ..., g_k$.

Definition 1: Orbit.

Let B be a subgroup of the automorphism group of graph G. The orbit of vertex v under B is the set U of vertices to which v can be mapped by automorphisms in B. If B is the automorphism group of G then U is called an orbit of G.

Example 1



Let $B = <(14)(23),(12)(34)(56)>$. The orbit of 1 under B is the set $\{1234\}$.

Definition 2: Trajectory.

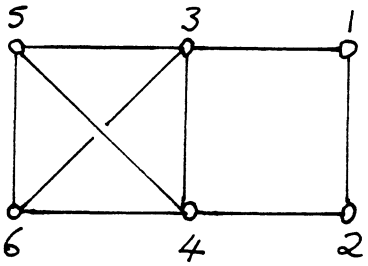
The trajectory of vertex v under automorphism g of graph G is the set of vertices to which v can be mapped by powers of g. That is, it is the orbit of vertex v under group $< g >$.

The definitions of orbit and trajectory are extended in the natural way to groups acting on arbitrary sets.

Definition 3: Refinement.

Let $V_1, V_2, ..., V_k$ be the trajectories under automorphism g in graph G containing more than one vertex (the *nontrivial* trajectories). h, another automorphism of G, is a refinement of g if its nontrivial trajectories are a proper subset of $\{V_1, V_2, ..., V_k\}$.

Example 2



Let $g = (12)(34)(56)$, $h = (12)(34)$, and $k = (56)$. The nontrivial trajectories under g are the sets $\{12\}$, $\{34\}$, and $\{56\}$; the nontrivial trajectories under h are the sets $\{12\}$ and $\{34\}$; the nontrivial trajectory under k is the set $\{56\}$; so h and k are refinements of g.

Definition 4: Minimal.

The automorphism h of graph G is minimal if h has no refinements.

Example 3

In example 2, g is not minimal, while k and h are minimal.

Since the term "self automorphism" is cumbersome the terms rigid and supple are defined instead.

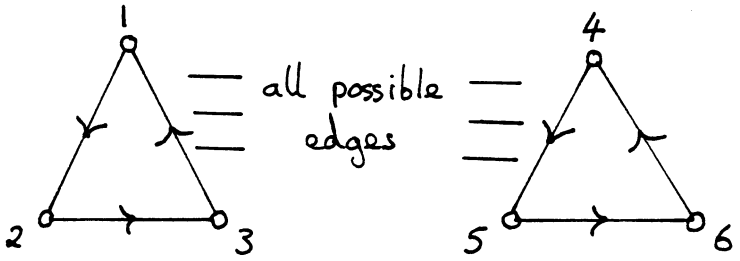
Definition 5: Rigid, supple.

Graph G is rigid if its only automorphism is the identity map. G is supple if it is not rigid.

Definition 6: Full.

Let A be a group acting on a set and let B be a subgroup of A. C, the *closure* of B, is the group of automorphisms in A satisfying $h(V) = V$ for all orbits V under B. B is full if $B = C$.

Example 4



The automorphisms of this graph include $g = (123)(456)$ and $h = (123)$. Let $B = < g >$. Then $h \in C$, the closure of B, since $h(\{123\}) = \{123\}$ and $h(\{456\}) = \{456\}$. But since $h \notin B$, B is not full.

3. *Motivation and the outline of the algorithm.*

Given a rigidity test, it is desired to obtain an isomorphism test. As shown below, given a rigidity test it is possible to obtain an automorphism. This suggests that one try to build the whole of the automorphism group en route to an isomorphism test. An algorithm for finding the automorphism group is developed below.

Let A be the automorphism group of graph G. Suppose subgroup B of A is known. The aim is to find more of A, that is, to find one or more automorphisms in $A - B$. To be able to use the rigidity test to find additional automorphisms the graph G must be changed so that the use of the rigidity test does not return an automorphism in B. The change must be sufficiently minor so that not too many new automorphisms are introduced, and not too many automorphisms outside of B are lost. The change we consider, called *reducing*, is to replace each orbit under B by a single vertex and to add some labelling so that not too much information about the original graph is lost. This process is defined more precisely below.

Some of the automorphisms in $A - B$ may map orbits of G under B to each other. That is, if $g \in A - B$, V is an orbit under B, and $g(V) = U$, then U is also an orbit under B. Call such automorphisms B-orbit preserving. To each B-orbit-preserving automorphism of G there corresponds an automorphism of the reduced graph, G'. The hope is that from the automorphisms of the reduced graph the B-orbit-preserving automorphisms of the original graph can be calculated. For full, normal subgroups B all automorphisms in $A - B$ are B-orbit-preserving, as is proven in lemma 1, section 4. This suggests the following algorithm for finding automorphism group A of graph G.

1) Find B, a full, normal subgroup of A.

2) Reduce G under B to give graph G'.

3) Let A' be the automorphism group of G'. Recursively find A'.

4) From A' and B construct A.

If A is abelian all its subgroups are normal. The following lemma, proven in section 4, guarantees that some of them are full also.

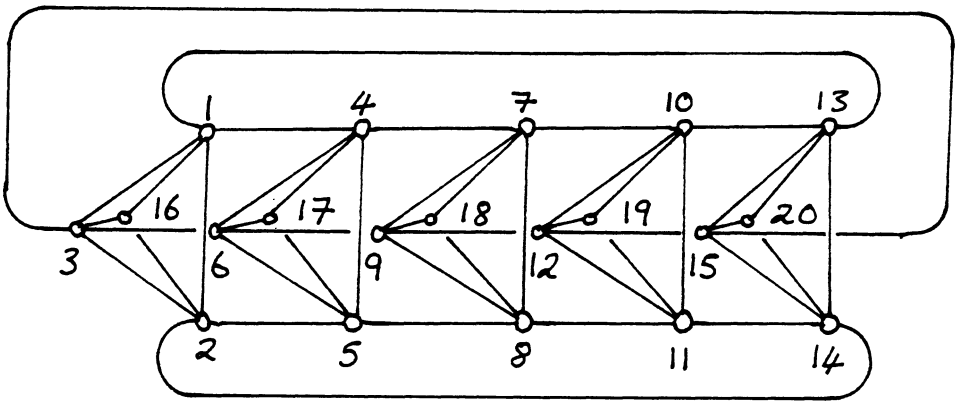
Lemma 2: Supppose automorphism group A of graph G is abelian. Let g be a prime order, minimal automorphism of G and let $B = < g >$. Then B is a full, normal subgroup of A.

For graphs with abelian automorphism groups, this lemma suggests refining step 1 of the above algorithm as follows.

1) Find a prime order, minimal automorphism g of G. Set $B = < g >$.

If the algorithm is only for graphs with abelian automorphism groups step 3 of the algorithm may no longer be valid as A' need not be abelian.

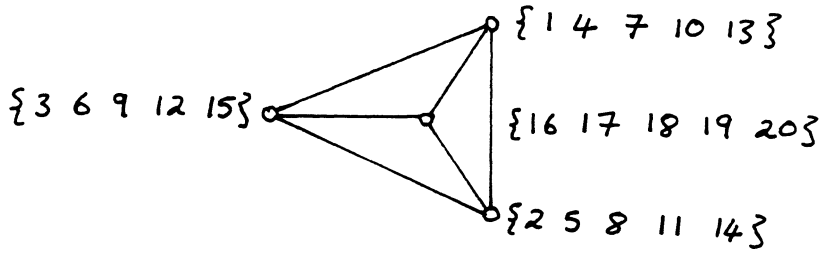
Example 5



The edges, besides those shown, are $(i, i+3 \bmod 15)$ and $(i, i+4 \bmod 15)$ for $1 \leq i \leq 15$. All the edges are undirected. The graph has the abelian automorphism group $<(1\,2\,3)(4\,5\,6)(7\,8\,9)(10\,11\,12)(13\,14\,15), (1\,4\,7\,10\,13)(2\,5\,8\,11\,14)(3\,6\,9\,12\,15)(16\,17\,18\,19\,20)>$. To see that this includes all the automorphisms of the graph consider the graph with vertex 1 fixed; it is rigid. Since the automorphism group moves vertex 1 to every vertex between 2 and 15, and since vertices 16 through 20 are not automorphic to vertex 1, we must have the whole automorphism group. On reducing under the group

$$<(1\,4\,7\,10\,13)(2\,5\,8\,11\,14)(3\,6\,9\,12\,15)(16\,17\,18\,19\,20)>,$$

the following graph is obtained.



It has a non-abelian automorphism group.

To make the algorithm valid it suffices to require that G is a special graph (defined below). Intuitively, a graph is special if whatever sequence of reductions the algorithm makes on the graph, the resulting graph has an abelian automorphism group.

Definition 7: Special.

Graph G is said to be special if its automorphism group A is abelian, and for each prime order, minimal automorphism g in A, G_g, the reduction of G under $< g >$, is special.

In section 4, several lemmas, including the two mentioned above, are proved. In section 5, a method for finding a minimal automorphism is described. In section 6, step 4 of the algorithm is detailed. In section 7, the timing analysis is given.

4. *Several lemmas*

Lemma 1: Let A be a group acting on set S. Let B be a full subgroup of A. Then B is normal in A if and only if each g in A maps orbits under B to orbits under B.

Proof: Assume B is normal in A. Then $gBg^{-1} = B$. Let V_1 and V_2 be orbits of S under B. Let g be in $A - B$ and assume, for a contradiction, that for c and d in V_2, $c = g(a)$ for some a in V_1 and $d = g(b)$ for some b not in V_1. Let h be an element of B such that $h(c) = d$. $g^{-1}hg(a) = b$, so a and b must be in the same orbit under B. This contradicts the assumption that a and b are in different orbits. Thus $g^{-1}(V_2)$ must be contained in a single orbit, V_1 say. Applying the argument to g^{-1}, we deduce that $g(V_1)$ is also contained in a single orbit. That is, $g(V_1) = V_2$. But V_1 was an arbitrary orbit, so it has been shown that any g in A maps orbits under B to orbits under B.

Now assume that every g in A maps orbits under B to orbits under B. Let V_1 be an orbit under B and h an element of B. Then $g^{-1}hg(V_1) = V_1$; that is, $g^{-1}hg$ preserves the orbits under B. Since B is full, $g^{-1}hg$ is in B. That is, B is normal in A. □

Lemma 2: Let A be the automorphism group of graph G. Let g be a prime order, minimal automorphism of G, and let B be $< g >$. Suppose A is abelian. Then B is a full, normal subgroup of A.

Proof: B is normal in A, since A is abelian. Suppose, for a contradiction, that B is not full. Then there is some minimal, nonidentity automorphism h not in B, and thus not a power of g, that maps the trajectories under g to themselves. There is at least one trajectory U, say, on which the action of h is not a power of the action of g. For otherwise, some combination of powers of g and h would be a refinement of g (and of h), contradicting the minimality of g. Let $u \in U$; for some power of g, r say, $g^r(u) = h(u)$. Hence $g^r h^{-1}$ is an automorphism of G fixing some, but not all, of the vertices of U. This contradicts lemma 3. □

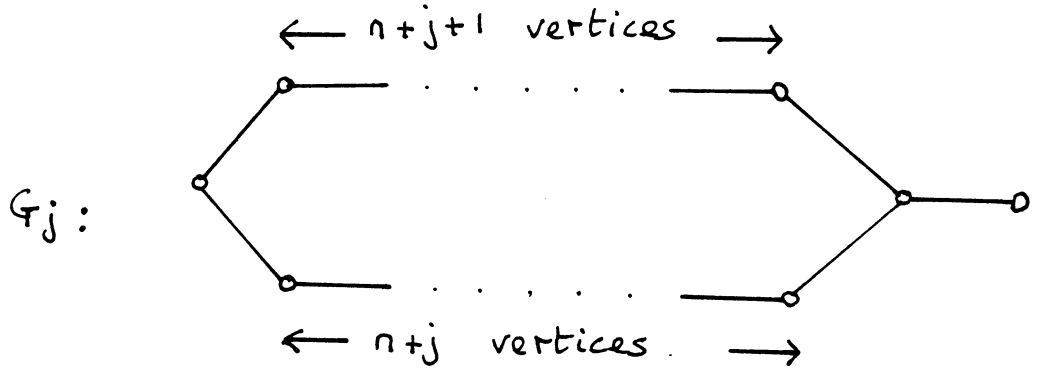
Lemma 3: Let A be the abelian automorphism group of graph G. Let U be an orbit of G and let g be an automorphism of G. Then either g moves every vertex in U or g fixes every vertex in U.

Proof: Suppose g moves some vertex $u \in U$, but fixes some vertex $v \in U$. Since U is an orbit of G there exists an automorphism h of G with $h(u) = v$. Then $gh \neq hg$, contradicting A being abelian. □

5. *Finding an automorphism, given a test for rigidity*

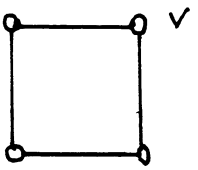
Given a supple graph G, together with a rigidity test, a method is provided for finding a nonidentity automorphism of G. Since vertices of the graph are colored as the construction of an automorphism proceeds, the rigidity test is made color preserving. The idea is to have a distinct graph G_j corresponding to each color C_j. Coloring a vertex with C_j is represented by attaching graph G_j to that vertex. We must ensure that G_j is rigid and that it is so distinct from the remainder of the graph that an automorphism of the graph can map G_j only to copies of itself. It is easiest to show this by diagram.

Diagram 1

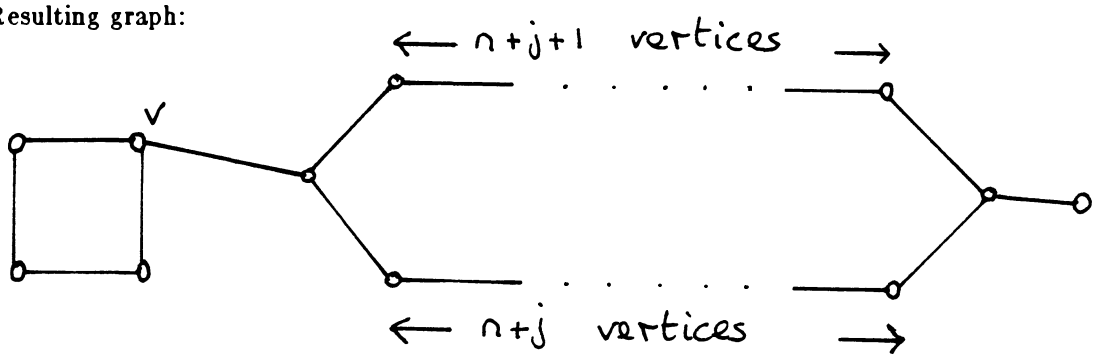


Here n is the number of vertices in G. We observe that G_j is rigid and since each of its arms has more than n vertices an automorphism of a graph colored using this structure can map G_j only to copies of itself.

For example, graph G whose vertex v is to be colored C_j:



Resulting graph:



Lines can be colored in a similar way if desired.

A vertex of graph G is *fixed* if all color preserving automorphisms leave it fixed. A vertex is fixed by coloring it with a new color. As many vertices of G as possible are fixed, until fixing any further vertex would make G rigid. To achieve this, each vertex, in turn, is tested by fixing it; if this makes the graph rigid it is unfixed; otherwise it is left fixed. Each vertex need be tested only

once. The order in which the vertices are tested may affect which ones are fixed, though. The nonidentity, color-preserving automorphisms of G must move all the unfixed vertices. It remains to find such an automorphism.

Let $u_1, u_2, \ldots, u_r$ be the vertices that have been fixed. Let G_1 and G_2 be copies of the graph G with these vertices fixed. This pair of graphs form the graph H.

Consider the order two automorphisms of H that interchange G_1 and G_2. All such automorphisms are derived from automorphisms of G, as follows. Let g be an automorphism of G. The *corresponding* automorphism h of H is given by interchanging G_1 and G_2, with automorphism g applied to the mapping of G_1 onto G_2, and automorphism g^{-1} applied to the mapping of G_2 onto G_1. The aim is to find an automorphism h of H with *corresponding* automorphism g of G not the identity map. From h it is easy to obtain g.

Suppose uncolored vertex v of G_1 is colored with a new color. Then G_1 becomes rigid. However, if $h(v)$ is colored with the same color, then H is supple, h being a nonidentity automorphism of H.

To find $h(v)$, in turn, each vertex in G_2 is colored with the same color as v. If H is supple under this coloring $h(v)$ is the vertex that has been colored. Otherwise the vertex just colored is uncolored and another one is tried. The vertex $i(v)$ is not tried, since the automorphism that corresponds to i is the identity automorphism of G. The pairing is extended to the remaining vertices of G_1, one by one, in the same way. The action of the automorphism h is then known.

Note that automorphism g and the automorphisms it generates are minimal, since as many vertices as possible were fixed, and thus these automorphisms have no refinements. Any prime order automorphism generated by g is suitable for step 1 of the algorithm for finding the automorphism group.

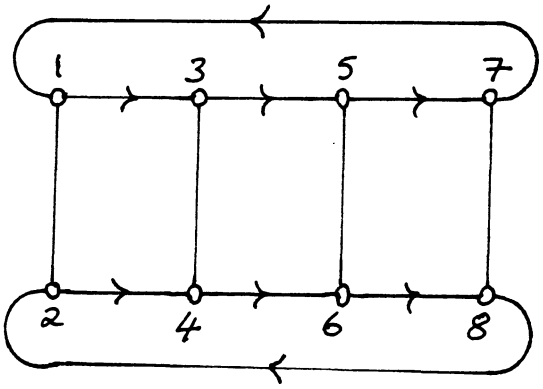
6. *Building* A from A' and g

At this point a precise definition of reduction is needed.

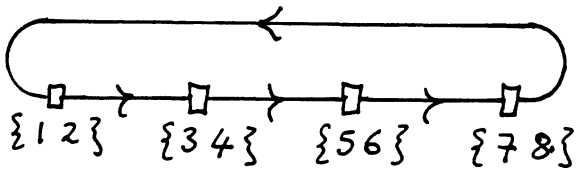
Definition 8: Reduction.

Let $G = (V,E)$ be a graph, possibly with some colored vertices. Let A be the group of color preserving automorphisms of G, and let B be a subgroup of A. Suppose $V_1, V_2, ..., V_k$ are the orbits of G under B. The reduction of G under B is the graph $G' = (V', E')$ built as follows. Each orbit V_i is replaced by a single vertex v_i', for $i = 1, 2, ..., k$. The vertex v_i' is colored with the same color as each of the vertices in V_i. The vertices corresponding to orbits of different sizes are colored with different colors. (This may result in some vertices having two colors, which can be treated as a new color.) An edge is added between vertices v_i' and v_j' if and only if there exists at least one edge between a vertex in orbit V_i and a vertex in orbit V_j, for $1 \leq i < j \leq k$.

Example 6

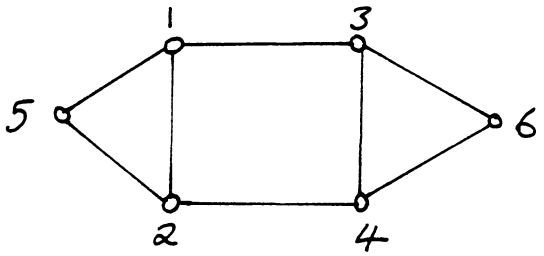


Let $g = (1\,2)(3\,4)(5\,6)(7\,8)$; reducing under $< g >$ gives:

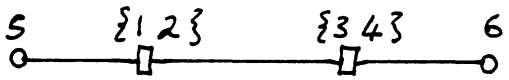


Consider carrying out a reduction, by looking at the following example.

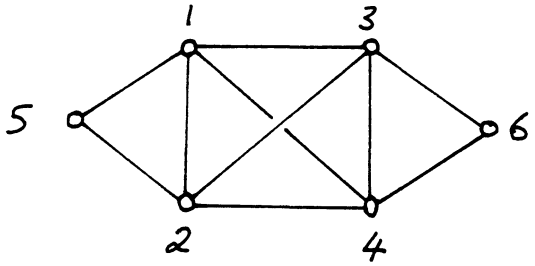
Example 7



Let $g = (1\,2)(3\,4)$; reducing under $< g >$, gives the graph:



This reduction can be thought of as occurring in two steps, the intermediate form being the graph

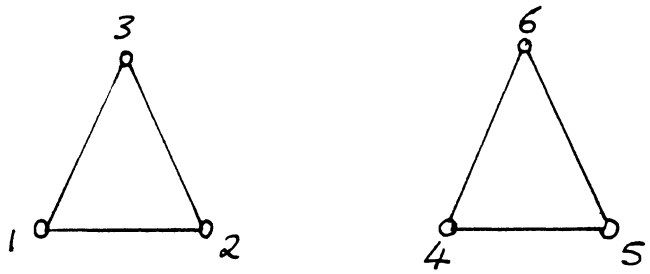


In general, first the edges between pairs of trajectories are filled in (if there were any to begin with) and then the trajectories are shrunk to single vertices. For this algorithm the automorphisms are meant to be orbit preserving. This is ensured by preserving the trajectories under g, as defined below. A reduction can also be broken up by carrying it out on just a few trajectories at a time. These ideas motivate the next four definitions.

Definition 9: Preserving trajectories.

Trajectory S of graph G is preserved under minimal automorphism g by adding directed edges $(u, g(u))$ for all $u \in S$. These edges are colored with the same new color. The trajectories under g are preserved by preserving every trajectory under g. The same new color is used for all the new directed edges.

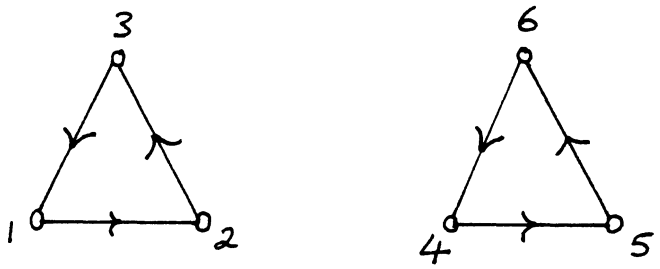
Example 8



Suppose the graph also has edges (14), (25), (36).

Let $g = (123)(456)$.

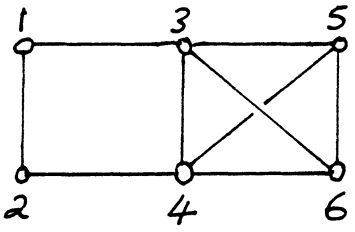
The graph is made trajectory preserving under g by making the edges (12), (23), (31), (45), (56), (64) directed, thus:



Definition 10: Loose link for trajectories.

Let S and T be two trajectories of graph G under minimal automorphism g. S and T are said to be loosely linked if when there is at least one edge between a vertex in S and a vertex in T then every possible edge between the vertices in S and the vertices in T lies in G.

Example 9



Let $g = (12)(34)(56)$. The trajectories under g are $U = \{12\}$, $V = \{34\}$, $S = \{56\}$. The pair of trajectories U and V are not loosely linked, while the pair V and S are loosely linked.

A technical note: To make S and T loosely linked, besides adding edges between S and T, all the trajectories under g are preserved.

Definition 11: Loose link for sets of vertices.

Let U and V be disjoint sets of vertices in graph G, both of them being unions of trajectories under minimal automorphism g. U and V are loosely linked if for all pairs S and T of trajectories under g, $S \subseteq U$, and $T \subseteq V$, S and T are loosely linked.

Definition 12: Reduction on U.

Let g be a minimal automorphism of graph G. Let U be a set of vertices in G, and suppose U is a union of several trajectories under minimal automorphism g. The reduction on U under g preserves the trajectories under g, replaces every trajectory under g in U with a single vertex, leaves the other vertices unchanged, and adds edges as for the standard reduction. These new vertices are colored with a new color to distinguish them from the old vertices.

It is useful to introduce the notion of a tight link. It is the opposite of a loose link.

Definition 13: Tight link for trajectories.

Two trajectories are tightly linked if they are not loosely linked.

The idea is to divide the reduction of G to G' into small steps, obtaining a series of intermediate graphs. The automorphism group A is found from A' by first finding the automorphism groups of these intermediate graphs. Two trajectories under g are said to be equivalent if the corresponding vertices of G' lie in the same orbit of G'. Two vertices of G are said to be equivalent if they lie in equivalent trajectories. Let $U_1, U_2, \ldots, U_r$ be the equivalence classes of the vertices of G. Define G_i to be the graph G with the reduction under g made on just $U_1 \cup U_2 \cup \ldots \cup U_i$, for $i = 0, 1, \ldots, r$. Note that $G = G_0$ and $G' = G_r$. In turn, the automorphism groups of $G_{r-1}, G_{r-2}, \ldots, G_1, G_0$ will be found.

The step between G_{i+1} and G_i is refined into even smaller steps: first, one by one, the sets U_i and U_j are made loosely linked for $j \neq i$, $1 \leq j \leq r$; then, secondly, a reduction is made on the set U_i under the projection of g onto G_i. Let the sequence of graphs be $G_{i+1} = H_0, H_1, ..., H_r = G_i$. In turn, the automorphism groups of $H_r, H_{r-1}, ..., H_0$ will be found. Note that the sets U_i are known only after A' is found, so the graphs G_i can only be found once the automorphism group of G' is known.

Lemma 4: Let h be an automorphism of H_j, and let h' be the projection of h onto G'. Then h' is an automorphism of G'.

Proof: h is $< g >$-orbit-preserving, and in G' the orbits under g are replaced by single vertices. Hence the action of h' on G' is an automorphism. □

Now, the method for finding the automorphism groups of the graphs H_j is described. There are two types of reductions that are carried out. First, on graph H two sets of vertices U and V are made loosely linked, giving graph K; U and V are sets of vertices corresponding to orbits of G'. Second, on graph H a reduction is made on set U of vertices, giving graph K; U is a set of vertices corresponding to an orbit of G'. In each case, given the automorphism groups of K and G', we show how to obtain the automorphism group of H using the rigidity test.

The method for the first case is described. A set of generating automorphisms for the automorphism group of H can be divided into two parts. One, generating automorphisms for the automorphism group of H with the vertices in $U \cup V$ fixed; this group is exactly the automorphism group of K with the vertices in $U \cup V$ fixed. Two, generating automorphisms for the automorphism group of H restricted to $U \cup V$. Together these provide generating automorphisms for the automorphism group of H. The second set of generating automophisms is described in more detail.

Let g_H be the projection of g onto H; g_H need not be minimal. For some of the trajectories under g which are tightly linked in G may be loosely linked in H. If g_H is minimal write

$g_H = g_1$; otherwise let $g_H = g_1 g_2 \ldots g_t$, where each g_i is minimal and a refinement of g_H.

Let U' and V' be the projections onto G' of U and V, respectively. Let $\{k_1, k_2, \ldots, k_y\}$ be a maximal set of automorphisms of K such that their projections onto G' restricted to $U' \cup V'$ are unequal and let k_j' be the projection of k_j onto G', for $1 \leq j \leq y$. Let $\{k_{i_1}', k_{i_2}', \ldots, k_{i_s}'\}$ be the subset of $\{k_1', k_2', \ldots, k_y'\}$ that are projections of automorphisms of H, and let h_j be an automorphism of H whose projection onto G' is k_{i_j}', for $1 \leq j \leq s$. As is shown below, in lemma 5, the automorphism group of H is generated by:

a) generators for the automorphism group of H with the vertices in $U \cup V$ fixed.

b) $g_1, g_2, \ldots, g_t$.

c) $h_1, h_2, \ldots, h_s$.

The automorphisms $g_1, g_2, \ldots, g_t$ are found by inspecting the graph H and determining which trajectories under g are tightly linked. Since the automorphisms g_i move distinct sets of vertices t is bounded by n. Generators for the automorphism group of H with the vertices in $U \cup V$ fixed are just generators for the automorphism group of K with the vertices in $U \cup V$ fixed. A polynomial sized set of these generators is found using the methods of [4].

The set $\{k_1', k_2', \ldots, k_y'\}$ is found by the methods of [4] also, by obtaining generators for the automorphism group of K restricted to $U \cup V$, taking projections of the generators onto G', and generating the y possible automorphisms. This is polynomial as $y \leq n^2$. For, by lemma 3, each automorphism of G' either moves every vertex in U' or no vertex in U'; likewise with respect to V'. Thus the action of an automorphism of G' on $U' \cup V'$ is fully determined by its action on one vertex of U' and one vertex of V'. Hence $y \leq n^2$.

Now, a method is described for testing whether k', an automorphism of G', is the projection of an automorphism h of H, and if it is, for finding h. We note that k' determines the action of h on the trajectories under g: whatever k' does to its vertices, h, if it exists, does to the corresponding trajectories. The method for finding h resembles the method given in section 5 for finding an automorphism.

A graph F is built consisting of two copies of H; a similar graph F' is built consisting of two copies of G'. Let f' be the automorphism of F' corresponding to k' (see section 5), and let f be the automorphism of F corresponding to h, if it exists. f' pairs the vertices in F'. The correponding pairs of trajectories in F are assigned distinct colors. Each vertex is colored with the color assigned to its trajectory.

Any automorphism of F that interchanges the copies of H must have f' as its projection onto F'. The other automorphisms of F are those generated by $< g_1, g_2, ..., g_t >$ on each copy of H. By further coloring the vertices, the automorphisms in $< g_1, g_2, ..., g_t >$ are made noncolor preserving while leaving some color preserving automorphism whose projection onto F' is f', if there was one to begin with.

Let T_{1j} be a trajectory in the first copy of H, T_{1j} being in the set S_j, the set of trajectories containing vertices moved by g_j. One vertex v_{1j} is chosen from each set T_{1j}. Each vertex v_{1j} is colored with a distinct new color. All the maps in $< g_1, g_2, ..., g_t >$ acting on the first copy of H are no longer color preserving. For the sets S_j are disjoint, so on coloring one vertex in T_{1j}, g_j ceases to be color preserving, but g_i is unaffected, for $i \neq j$.

let T_{2j} be the trajectory in the second copy of H with which T_{1j} has been paired. One vertex v_{2j} is chosen from each T_{2j}. Vertex v_{2j} is colored with the same color as v_{1j}. All the maps in $< g_1, g_2, ..., g_t >$ acting on the second copy of H are no longer color preserving. For consider the set S_i, the trajectories under g moved by g_i. The trajectories in S_i are tightly linked. So any automorphism of H must map set S_i to some other set S_k. Let S_{1i} and S_{2i} be set S_i in the first and second copy of H, respectively. Any nonidentity automorphism of F maps sets S_{1i} to sets S_{2k}, one-one and onto. Thus each vertex v_{2j}, for $1 \leq j \leq t$, lies in a trajectory from a distinct set S_{2k}, for $1 \leq k \leq t$. That is, for each set S_{2k} there is one trajectory in which a vertex is colored, so the automorphisms in $< g_1, g_2, ..., g_t >$ acting on the second copy of H are not color preserving.

However, if originally there was an automorphism f of F whose projection onto F' was f', then there still is one. It suffices to multiply f by appropriate powers of $g_1, g_2, ..., g_t$, acting on

the first copy of H. This automorphism is found by a method similar to that of section 5.

The second case is very similar. Let $\{k_1, k_2, ..., k_y\}$ be a maximal set of automorphisms of K such that their projections onto G' restricted to U' are unequal. As before, let k_j' be the projection of k_j onto G', for $1 \leq j \leq y$, let $\{k_{i_1}', k_{i_2}', ..., k_{i_s}'\}$ be the subset of $\{k_1', k_2', ..., k_y'\}$ that are projections of automorphisms of H, and let h_j be an automorphism of H whose projection onto G' is k_j', for $1 \leq j \leq s$. If g_H is minimal set $g_H = g_1$; otherwise let $g_H = g_1 g_2 ... g_t$ with each g_i being minimal and a refinement of g_H. Finally let U' be the projection of U onto K. The automorphism group of H with the vertices in U fixed is exactly the automorphism group of K with the vertices in U' fixed. In this case $y \leq n$. By a similar argument to that given above, we deduce the automorphism group of H is generated by $g_1, g_2, ..., g_t$, plus generators for the automorphism group of H with the vertices in U fixed, together with $h_1, h_2, ..., h_s$. Each of these is found by a method similar to that used in the first case.

Lemma 5: Let K be the graph H with sets U and V made loosely linked with respect to the prime order automorphism g. If g is minimal write $g = g_1$; otherwise let $g = g_1 g_2 ... g_t$ where each g_i is minimal and a refinement of g. Let G' be the reduction of H under g, and let U' and V' be the projections of U and V onto G'. Let $\{k_1, k_2, ..., k_y\}$ be a maximal set of automorphisms of K such that their projections onto G' restricted to $U' \cup V'$ are unequal. Let k_j' be the projection of k_j onto G', for $1 \leq j \leq y$, and let $\{k_{i_1}', k_{i_2}', ..., k_{i_s}'\}$, be the subset of $\{k_1, k_2, ..., k_y\}$ that are projections of automorphisms of H. Finally let h_j be an automorphism of H whose projection onto G' is k_{i_j}', for $1 \leq j \leq s$.

Suppose:

1) the trajectories under g are preserved on H, and

2) U' and V' are orbits of G'.

Then the automorphism group of H is generated by:

a) generators for the automorphism group of H with the vertices in $U \cup V$ fixed. Call this group B.

b) $g_1, g_2, ..., g_t$.

c) $h_1, h_2, ..., h_s$.

Proof: By (1) each automorphism of H has a projection onto G', and so from (3) each automorphism of K maps U to U and V to V. It follows that every automorphism of H is an automorphism of K. Let h be an automorphism of H and let h' be the projection of h onto G'. By the maximality of the set $\{k_1, k_2, ..., k_y\}$, h' is equal to some k_{i_j}' on $U' \cup V'$. As shown below, in lemma 6, there is an automorphism h_j of H whose projection onto G' is equal to k_{i_j}'.

Consider the automorphism $h_j h^{-1}$ of H. On set $U \cup V$ it maps each trajectory under g to itself, and so there is some automorphism $f \in < g_1, g_2, ..., g_t >$ with the same action on $U \cup V$, since the trajectories under each of $g_1, g_2, ..., g_t$ are tightly linked. Thus $(h_j h^{-1})f^{-1}$ fixes the vertices in $U \cup V$, and so is an automorphism of H fixing the vertices in $U \cup V$. Hence $h \in < h_j, g_1, g_2, ..., g_t, B >$. □

Lemma 6: Let H, K, and G' be as in lemma 5. Let h be an automorphism of H and let h' be its projection onto G'. Let k be an automorphism of K whose projection onto G', k', satisfies $k' = h'$ on $U' \cup V'$. Then there is an automorphism d of H whose projection onto G' equals k'.

Proof: Let g_K be the projection of g onto K, and let $g_K = g_1 g_2 ... g_t$, with each g_i being minimal and a refinement of g_K. Consider the automorphism $h k^{-1}$ of K. On $U \cup V$ it maps the trajectories under g to themselves. As above, in the proof of lemma 5, there is some automorphism $f \in < g_1, g_2, ..., g_t >$ with the same action on $U \cup V$. Then $h k^{-1} f^{-1}$ is an automorphism of K fixing the vertices in $U \cup V$, and thus is an automorphism of H. Set $d = (hk^{-1}f^{-1})^{-1}h$ $(= f k)$; it is an automorphism of H and its projection onto G' is k'. □

7. *Timing*

Consider the outline algorithm given in section 3. Step 2, the reduction, clearly takes polynomial time. Now, step 1, finding a minimal automorphism, is analyzed. To find a maximal set of vertices, whose fixing leaves the graph supple, requires at most n calls to the rigidity test. The pairing procedure, which obtains a nonidentity automorphism, needs at most n^2 calls to the rigidity test. So step 1 is polynomial in the rigidity test.

Next, step 4, building A from A' and g, is analyzed. As argued in section 6, it is sufficient to obtain the automorphism group of H from the automorphism group of K, for polynomially many such graphs. Each such construction takes polynomial time, plus, possibly, polynomially many calls to the rigidity test. Thus step 4 is polynomial in the rigidity test.

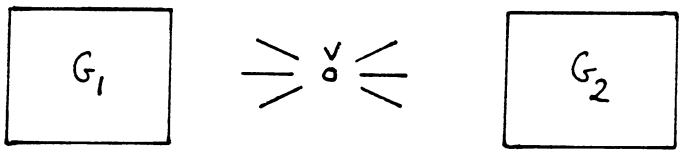
From step 3, the recursive call, this leads to the following recurrence for the running time $T(n)$ of the algorithm:

$$T(n) \leq T(n-1) + p(n) \leq n\,p(n),$$

where $p(n)$ is a polynomial in n. So the problem of finding the automorphism group for special graphs is polynomial time Turing reducible to the rigidity problem.

8. *Isomorphism testing*

The standard construction for going from a method for finding an automorphism group of graphs to an isomorphism test does not work for special graphs. The standard method to test if two graphs G_1 and G_2 are isomorphic follows. The graph H pictured below is built.



all possible edges between v and G_1 and between v and G_2 are added to H.

The automorphism groups A_1, A_2 and A are found, where A_1, A_2, and A are the automorphism groups of G_1, G_2, and H, respectively. G_1 is isomorphic to G_2 if and only if $|A| = 2|A_1||A_2|$.

The problem is that even if G_1 and G_2 are special, H need not be, and it may not be possible to find the automorphism group A. For let g be an automorphism of H that fixes the vertices in G_2 but moves the vertices in G_1, and let h be an automorphism of H that interchanges G_1 and G_2. Then $hg \neq gh$, so A is not abelian, and H is not special.

When testing two special graphs for isomorphism a construction similar to the one above for finding the automorphism group of a graph cannot be used since isomorphic reductions must be made on both graphs, and it is not clear how to determine that two reductions are isomorphic. This leads us to make reductions by all isomorphisms of a given type (eg. of order p) in both graphs, and thus to develop an algorithm for a slightly smaller class of graphs.

In the construction below, the graphs will be reduced under various different groups. It is necessary that the resulting graphs have abelian automorphism groups. This leads to a new definition of special.

Definition 14: Special.

Let A be the automorphism group of graph G, and let G_B be the reduction of G under subgroup B of A. G is special if A is abelian and for all subgroups B of A, G_B is also special.

Definition 15: Maximal reduction.

Let A be the automorphism group of graph G. G', the maximal reduction of G is the reduction of G under A.

Definition 16: Full reduction.

The intuitive idea of a full reduction is to make maximal reductions until no further change to the graph occurs under a maximal reduction. Now, for the formal definition. Let $G = G_1, G_2, \ldots, G_k$ be a sequence of graphs with G_{i+1} being the maximal reduction of G_i, for $i = 1, 2, \ldots, k\text{-}1$. G_k is called the full reduction of G if G_k is rigid.

Definition 17: Maximal reduction except on U.

Let U be a set of vertices of graph G, and let A be the automorphism group of G with the vertices in U fixed. G', the maximal reduction of G except on U, is the reduction of G under A.

Definition 18: Full reduction except on U.

The intuitive idea of a full reduction except on a set U of vertices is to make maximal reductions except on U until no further change occurs to the graph under such a maximal reduction. Now, for the formal definition. Let U be a set of vertices of graph G. Let $G = G_1, G_2, ..., G_k$ be a sequence of graphs with G_{i+1} being the maximal reduction of G_i except on U, for $i = 1, 2, ..., k$-1. If, on fixing the vertices in U, G_k becomes rigid, then G_k is the full reduction of G except on U.

Let G_1 and G_2 be the special graphs being tested for isomorphism. In the discussion that follows, G_1 and G_2 are assumed to be isomorphic; if a step does not proceed as described then either one, at least, of G_1 and G_2 is not special or G_1 and G_2 are not isomorphic. If G_1 and G_2 are both rigid then an isomorphism can be sought by a method similar to that of section 5. Otherwise, if G_1 and G_2 are not rigid, let g_1 be a prime order, minimal automorphism of G_1. A method is given below for reducing G_1 by g_1 and for reducing G_2 by an isomorphic automorphism g_2, giving the graphs G_1' and G_2'. An isomorphism of G_1' and G_2' is found recursively. The sequence of graphs built in section 6 is constructed between G_1 and G_1', and an isomorphic sequence of graphs is built between G_2 and G_2'. Let H_1 and K_1 be a pair of graphs in the sequence between G_1 and G_1', and let H_2 and K_2 be the isomorphic graphs in the sequence between G_2 and G_2'. We show how to obtain an isomorphism of H_1 and H_2 from an isomorphism of K_1 and K_2. After polynomially many iterations this leads to an isomorphism of G_1 and G_2.

To build the sequence of graphs, the automorphism groups of G_1' and of G_2' are found. From these the orbits of G_1' and of G_2' are found, and paired isomorphically. Since the

trajectories under g_1 and g_2 are known too, it is straightforward to build the isomorphic sequences of graphs between G_1 and G_1', and between G_2 and G_2'.

Now, the method for finding an isomorphism of H_1 and H_2 from an isomorphism of K_1 and K_2 is given. First, the automorphism groups of K_1 and K_2 are found. Suppose K_1 is the graph H_1 made loose on U_1 and V_1, and K_2 is the graph H_2 made loose on U_2 and V_2, where U_1 and U_2 are isomorphic, as are V_1 and V_2. Then, consider the actions of the isomorphisms of K_1 and K_2 restricted to $U_1 \cup V_1$ and to $U_2 \cup V_2$. There at most n^2 of them, neglecting multiplication by minimal parts of the projections of g_1 or g_2. One of these, at least, must be an isomorphism of H_1 and H_2, again neglecting the minimal parts of the projections of g_1 and g_2. This isomorphism can be found by a method similar to that of section 6. A similar argument handles the case that K_1 is the reduction of H_1 on U_1, and K_2 is the reduction of H_2 on U_2. (See the argument given in section 6.)

The isomorphism test either returns an isomorphism of the two graphs, or tells us that at least one of the following is true

1) the two graphs are not isomorphic

2) one, at least, of the two graphs is not special.

Making the same reduction on two graphs

The prime order, minimal automorphism g_1 of G_1 is known. A method is described for finding the isomorphic automorphism g_2 of G_2. First isomorphic sets of vertices in G_1 and G_2 are found. Each such set is a union of orbits in G_1 or G_2. These sets are refined until they have the following property.

> Let U_1 and U_2 be an isomorphic pair of the refined sets, and let H_1 and H_2 be the full reductions of G_1 except on U_1 and of G_2 except on U_2, respectively. Then H_1 is made rigid by fixing any vertex in U_1 and H_2 is made rigid by fixing any vertex in U_2.

By a method similar to that of section 5 an isomorphism of H_1 and H_2 is found. This isomorphism gives the action of g_2 on U_2: simply map the action of g_1 on U_1 onto U_2 under the

isomorphism. On repeating this for all pairs of isomorphic orbits, the action of g_2 on G_2 is found.

Initially, the sets of vertices U_1 and U_2 are just the vertices of G_1 and G_2, respectively. Now, an initial refinement is found. Let K_1 and K_2 be the full reductions of G_1 and G_2, respectively. Using a method similar to that of section 5, an isomorphism of K_1 and K_2 can be found, since the graphs K_1 and K_2 are both rigid. This isomorphism pairs corresponding vertices in K_1 and K_2. The corresponding sets of vertices in G_1 and G_2 are identified.

Next, the general method for refining the vertex sets is described. Let U_1 and U_2 be a pair of isomorphic sets of vertices, U_1 in G_1, and U_2 in G_2. Let H_1 and H_2 be the full reductions of G_1 except on U_1, and of G_2 except on U_2, respectively. Suppose it takes k maximal reductions to fully reduce H_1 (or H_2).

If $k=0$ the sets U_1 and U_2 are in the desired form. If $k=1$, then let K_1 and K_2 be the maximal reductions of H_1 and H_2, respectively. An isomorphism of K_1 and K_2 is easily found as they are rigid. The corresponding sets of vertices in H_1 and H_2 are paired; these pairs of sets of vertices satisfy the desired property.

If $k \geq 2$, then $k-1$ maximal reductions on H_1 and H_2 are made giving the graphs L_1 and L_2. If the projections of some pair of vertex sets onto L_1 and L_2 are not orbits of L_1 and L_2 then an isomorphism of L_1 and L_2 is found using method 1, below. This isomorphism allows the orbits of L_1 and L_2 to be paired isomorphically, providing a refinement of the vertex sets. On the other hand, if the projections of the vertex sets are orbits of the graphs L_1 and L_2, then the graphs F_1 and F_2 are obtained by making $k-2$ maximal reductions of H_1 and H_2, respectively. From F_1 and F_2 isomorphic graphs F_1' and F_2' are found, F_1' and F_2' being isomorphic reductions of F_1 and F_2, respectively. F_1' and F_2' are chosen so that they satisfy the following property.

Let A_1 and A_2 be isomorphic subgroups of the automorphism groups of F_1' and F_2', and let T_1 and T_2 be isomorphic sets of vertices in F_1' and F_2', respectively. Let E_1 and E_2 be the reductions of F_1' and F_2' under A_1 and A_2, let D_1 and D_2 be the maximal reductions of F_1' and F_2', and let R_1 and R_2 be the maximal reductions of E_1 and E_2, respectively. Then E_1 and E_2 are supple, D_1 and D_2 are supple, while R_1 and R_2 are rigid. Also every

automorphism in A_1 and A_2 moves some vertex in the sets T_1 and T_2, and the projections of T_1 and T_2 onto E_1 and E_2 are orbits of E_1 and E_2, respectively.

The procedure for finding F_1' and F_2', A_1 and A_2, and T_1 and T_2 is given below in method 2. Finally, an isomorphism of F_1' and F_2' is found. The procedure is given below in method 3. From this isomorphism a refinement of the vertex sets can be found by pairing isomorphic orbits of F_1' and F_2'. For if a refinement could not be found, under one maximal reduction each of F_1' and F_2' would become rigid, contradicting D_1 and D_2 being supple.

Method 1: Let E_1 and E_2 be two isomorphic special graphs such that under a maximal reduction each of E_1 and E_2 becomes rigid. A method is given for finding an isomorphism of E_1 and E_2.

If E_1 and E_2 are rigid then it is straightforward to find an isomorphism of E_1 and E_2 by a method similar to that of section 5. If not, two isomorphic vertices, v_1 in E_1 and v_2 in E_2 are found; v_1 and v_2 are colored with the same new color. The resulting pair of graphs are still isomorphic, so the procedure is applied recursively. Eventually, E_1 and E_2 are made rigid by this coloring. It remains to show how to find a pair v_1, v_2 of isomorphic vertices.

First, a method is given for pairing isomorphic orbits of E_1 and E_2. These are found by making maximal reductions of E_1 and E_2 giving graphs R_1 and R_2, respectively. It is straightforward to find an isomorphism of R_1 and R_2 as they are rigid. But the vertices of R_1 and R_2 correspond to the orbits of E_1 and E_2, so this allows the orbits of E_1 and E_2 to be paired isomorphically. Let O_1 and O_2 be one such pair of orbits, O_1 in E_1 and O_2 in E_2. Then any pair (v_1,v_2) of vertices, with $v_1 \in O_1$ and $v_2 \in O_2$, are isomorphic, as required.

Method 2: Let G_1 and G_2 be isomorphic special graphs, let H_1 and H_2 be the maximal reductions of G_1 and G_2, and let S_1 and S_2 be the maximal reductions of H_1 and H_2, respectively. Suppose that the following two properties hold.

1) S_1 and S_2 are rigid.

2) H_1 and H_2 are supple.

A method is given for finding graphs F_1 and F_2, isomorphic reductions of G_1 and G_2, for finding isomorphic automorphism groups A_1 and A_2, subgroups of the automorphism groups of F_1 and F_2, and for finding isomorphic sets of vertices T_1 and T_2, in F_1 and F_2, respectively, satisfying the following four properties.

1) Let E_1 and E_2 be the reductions of F_1 and F_2 under A_1 and A_2, respectively. Then E_1 and E_2 are supple.

2) Let R_1 and R_2 be the maximal reductions of E_1 and E_2, respectively. Then R_1 and R_2 are rigid.

3) Let D_1 and D_2 be the maximal reductions of F_1 and F_2, respectively. Then D_1 and D_2 are supple.

4) All the automorphisms in A_1 and A_2 move at least one vertex in T_1 or T_2, and the projections V_1 and V_2, of T_1 onto E_1 and T_2 onto E_2, are orbits of E_1 and E_2, respectively.

Let $V_{11}, V_{12}, \ldots, V_{1k}$ and $V_{21}, V_{22}, \ldots, V_{2k}$ be the isomorphic orbits of H_1 and H_2, respectively. They correspond to isomorphic vertices of S_1 and S_2; the isomorphism of S_1 and S_2 can easily be found since they are rigid graphs. Let $U_{11}, U_{12}, \ldots, U_{1k}$ and $U_{21}, U_{22}, \ldots, U_{2k}$ be the corresponding sets of vertices in G_1 and G_2. Consider the graphs $H_i = G_{i0}, G_{i1}, \ldots, G_{ik} = G_i$, for $i = 1, 2$, where G_{ij} is the maximal reduction of G_i except on $U_{i1} \cup U_{i2} \cup \ldots \cup U_{ij}$. We find the value of j satisfying

1) $G_{i,j-1}$ under a maximal reduction is rigid.

2) G_{ij} under a maximal reduction is supple.

There is such a value of j since under a maximal reduction G_{i0} $(= H_i)$ is rigid and under a maximal reduction G_{ik} $(= G_i)$ is supple.

G_{1j} and G_{2j} are the required graphs F_1 and F_2, U_{1j} and U_{2j} are the required sets T_1 and T_2, and the projections onto G_{1j} and G_{2j} of the automorphism groups of G_1 and G_2 with $U_{11} \cup U_{12} \cup \ldots \cup U_{i,j-1}$ and $U_{21} \cup U_{22} \cup \ldots \cup U_{2,j-1}$ fixed are the required automorphism groups A_1 and A_2, respectively. (The reductions of F_1 and F_2 under A_1 and A_2 are the graphs $G_{1,j-1}$ and

$G_{2,j-1}$, respectively.)

Method 3: Let F_1 and F_2 be two isomorphic special graphs, and let T_1 and T_2 be isomorphic sets of vertices in F_1 and F_2, respectively. Let A_1 and A_2 be isomorphic subgroups of the automorphism groups of F_1 and F_2, let E_1 and E_2 be the reductions of F_1 and F_2 under A_1 and A_2, let D_1 and D_2 be the maximal reductions of F_1 and F_2, and let R_1 and R_2 be the maximal reductions of E_1 and E_2, respectively. A method is given for finding an isomorphism of F_1 and F_2 when the following four conditions hold.

1) E_1 and E_2 are supple.

2). R_1 and R_2 are rigid.

3) D_1 and D_2 are supple.

4) All the automorphisms in A_1 and A_2 move at least one vertex in the vertex sets T_1 and T_2, and the projections V_1 and V_2, of T_1 onto E_1 and of T_2 onto E_2, are orbits of E_1 and E_2, respectively.

First, an isomorphism of E_1 and E_2 is found by method 1. Consider any isomorphism of F_1 and F_2; its projection onto E_1 and E_2 is an isomorphism of these graphs. If the isomorphism is restricted to $V_1 \cup V_2$, by lemma 3, there are at most n possible isomorphisms, $i_1, i_2, \ldots, i_k$, say, for some $k \leq n$. In turn, an attempt is made to extend each of $i_1, i_2, \ldots, i_k$ to F_1 and F_2, until an isomorphism of F_1 and F_2 is found.

Consider trying to extend i_j to F_1 and F_2. i_j pairs vertices in V_1 and V_2. These vertices correspond to the orbits of F_1 and F_2 under A_1 and A_2 contained in T_1 and T_2, respectively. This pairing is shown by assigning a separate color to the vertices in each pair of isomorphic orbits. The aim is to isomorphically pair all the vertices in T_1 and T_2. By coloring each such pair of vertices with a distinct color the graphs F_1 and F_2 become easier to test for isomorphism.

Next, the method for pairing the vertices in T_1 and T_2 is described. Suppose T_1 and T_2 fall into orbits $W_{11}, W_{12}, \ldots, W_{1t}$ and $W_{21}, W_{22}, \ldots, W_{2t}$, respectively, with W_{1i} isomorphic to W_{2i}, for $1 \leq i \leq t$. Any pair of vertices (v_{11}, v_{21}), with $v_{11} \in W_{11}$ and $v_{21} \in W_{21}$ are isomorphic.

They provide the first pair of isomorphic vertices, and they are colored with a new color. On making a full reduction of the graphs F_1 and F_2 (with the latest coloring), a pair of rigid graphs is obtained. Their isomorphism is easy to obtain; it enables the new, refined orbits of F_1 and F_2 contained in T_1 and T_2 to be paired isomorphically. The above procedure is then iterated until all the vertices in T_1 and T_2 are paired.

Finally, consider the graphs F_1 and F_2 with all the vertices in T_1 and T_2 paired isomorphically, and colored accordingly. Let S_1 and S_2 be the full reductions of F_1 and F_2, respectively. The isomorphism of S_1 and S_2, which can easily be found since these graphs are rigid, pairs sets of vertices in F_1 and F_2; call them superorbits. The crucial observation is that if an automorphism of F_1 or F_2 moves one vertex in a superorbit then it moves every vertex in the superorbit. For suppose an automorphism h_1 of F_1 moves some, but not all , of the vertices in the superorbit O_1. Then, consider O_1' and h_1', the projections of O_1 and h_1 onto E_1. h_1' moves some, but not all, of the vertices of O_1', and a contradiction is obtained from lemma 3 by showing that O_1' is contained in an orbit of E_1. An orbit of E_1 corresponds to a vertex of R_1, while O_1' corresponds to an orbit of S_1, and R_1 and S_1 are both full reductions of F_1, but S_1 comes from a more refined coloring of F_1 and hence its vertices correspond to smaller sets of vertices in F_1 than do the orbits in E_1. Contradiction. The method for finding an isomorphism of F_1 and F_2 is given below in method 4.

Method 4: Let F_1 and F_2 be isomorphic special graphs, and let R_1 and R_2 be their full reductions. The set of vertices in F_1 and F_2 corresponding to single vertices in R_1 and R_2 will be called superorbits. The method for finding an isomorphism of F_1 and F_2 is described in the case that the following property holds.

> If an automorphism of F_1 (or F_2) moves one vertex in a superorbit then it moves every vertex in that superorbit.

If F_1 and F_2 are rigid then an isomorphism is found by a method similar to that of section 5. Otherwise, F_1 and F_2 are reduced by isomorphic automorphisms h_1 and h_2 giving graphs E_1 and E_2, respectively. An isomorphism of E_1 and E_2 is found recursively. This isomorphism is

extended to F_1 and F_2, in the same way as in the general case. In contrast to the general case, finding isomorphic automorphisms is fairly simple. Let h_1 be a minimal automorphism of F_1, and let W_1 and W_2 be a pair of isomorphic superorbits of F_1 and F_2, respectively. Let C_1 and C_2 be the full reductions of F_1 except on W_1 and of F_2 except on W_2, respectively. On fixing any vertex in W_1 or W_2, the graph C_1 or C_2, respectively, is made rigid. Thus it is straightforward to find an isomorphism of C_1 and C_2. The action of h_2 on W_2 is just the image of the action of h_1 on W_1 under the isomorphism of C_1 and C_2. On applying this to every pair (W_1, W_2) of superorbits the action of h_2 on F_2 is found.

Timing

The four methods given above, as will be argued below, are all polynomial in the rigidity test. Thus, in the general method, making a refinement of the vertex sets takes time polynomial in the rigidity test. Since at most n such refinements can be made, given a minimal automorphism g_1 of G_1, it takes time polynomial in the rigidity test to find an isomorphic automorphism g_2 of G_2. From an isomorphism of G_1' and G_2', the reductions of G_1 and G_2 under g_1 and g_2, respectively, it takes time polynomial in the rigidity test to find an isomorphism of G_1 and G_2. This leads to a reccurrence of the form

$$T(n) \leq p(n) + T(n-1),$$

for the running time, where $p(n)$ is a polynomial in n. This has the solution

$$T(n) \leq n\, p(n),$$

so the running time of the algorithm is polynomial in the rigidity test.

Methods 1 and 2 are clearly polynomial in the rigidity test. By an argument similar to that given above, so is method 4. Now, method 3 is discussed. An attempt is made to extend at most n isomorphisms to F_1 and F_2. For each such attempt, first, a pairing of the vertices in T_1 and T_2 is made, taking time polynomial in the rigidity test; then, second, method 4 is used once. Thus method 2 takes time polynomial in the rigidity test. Hence, for special graphs, testing

isomorphism is polynomial in the rigidity test.

Acknowledgements

My most sincere thanks go to John Hopcroft. His help with the initial stages of this work and with the final writing were invaluable.

References

[1] L. Babai, Monte Carlo algorithms in graph isomorphism testing, submitted to SIAM Journal on Computing, 1979.

[2] L. Babai, D. Yu. Grigoriev, D. M. Mount, Isomorphism of graphs with bounded eigenvalue multiplicity, 14th Annl. STOC symp., 1982, pp. 310-324.

[3] I. Filotti and J. Mayer, A polynomial time algorithm for determining isomorphism of graphs of fixed genus, 12th Annl STOC Symp., 1980, pp. 236-243.

[4] M. Furst, J. Hopcroft, and E. Luks, Polynomial time algorithms for permutation groups, 21st Annl. FOCS symp., 1980, pp. 36-41.

[5] I.N. Herstein, Topics in Algebra, Blaisdell.

[6] C. M. Hoffmann, Subcomplete generalizations of graph isomorphism, submitted to JCSS, 1981.

[7] C. M. Hoffmann, An $O(n^4)$ isomorphism test for trivalent graphs, unpublished manusript.

[8] J. Hopcroft and R. Tarjan, A V logV algorithm for isomorphism of triconnected planar graphs, JCSS 7(1973), pp. 323-331.

[9] J. Hopcroft and J. Wong, A linear time algorithm for isomorphism of planar graphs, 6th Annl. STOC Symp., 1974, pp. 172-184.

[10] A. Lubiw, Some NP-Complete problems similar to graph isomorphism, SIAM Journal on Computing, 1(1981), pp. 11-21.

[11] D. Lichtenstein, Isomorphism for graphs embeddable on the projective plane, 12th Annl. STOC Symp., 1980, pp. 218-224.

[12] E. Luks, Isomorphism of graphs of bounded valence can be tested in polynomial time, 21st Annl. FOCS Symp., 1980, pp. 42-49.

[13] R. Mathon, A note on the graph isomorphism counting problem, IPL, 8(1979), pp. 131-132.

[14] G. Miller, Graph isomorphism, general remarks, 9th Annl. STOC Symp., 1977, pp. 143-150.

[15] G. Miller, On the $n^{\log n}$ isomorphism technique, 10th Annl. STOC Symp., 1978, pp. 51-58.

[16] G. Miller, Isomorphism testing for graphs of bounded genus, 12th Annl. STOC Symp., 1980, pp. 218-224.