Reprinted from INFORMATION AND CONTROL
Vol. 56. Nos. 1–2, January/February 1983

Isomorphism of Graphs Which are Pairwise k-Separable*

GARY L. MILLER

Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

A polynomial time algorithm for testing isomorphism of graphs which are pairwise k-separable for fixed k is given. The pairwise k-separable graphs are those graphs where each pair of distinct vertices are k-separable. This is a natural generalization of the bounded valence test of Luks. The subgroup of automorphisms of a hypergraph whose restriction to the vertices is in a given Γ_k group, for fixed k is constructed in polynomial time.

INTRODUCTION

The computational complexity of testing isomorphism of graphs is one of the outstanding open questions in the theory of computation. The problem of graph isomorphism is not believed to be NP-complete since the counting version of the problem, i.e., the number of isomorphisms, is polynomial time Turing equivalent to graph isomorphism (Babai, 1979; Mathon, 1979). For NP-complete problems their counting version seems to be harder (Angluin, 1980; Valient, 1979). On the other hand, polynomial time algorithms have only been found for special cases. These cases include; graphs of bounded genus (Filotti *et al.*, 1980; Lichtenstein, 1980; Miller, 1980), graphs of bounded valence (Luks, 1980), and graphs of bounded eigenvalue multiplicity (Babai *et al.*, 1982). In this paper and a companion paper (Miller, to appear) we consider two new classes of graphs. Here we consider the pairwise k-separable graphs.

DEFINITION. A graph G is ***pairwise k-separable*** if for each pair of distinct vertices x, y of G there exists a set of size k consisting of vertices and edges disjoint from x and y which separates x from y.

This class of graphs is interesting for two reasons. First, graphs of valence at most k are trivially contained in the graphs which are pairwise *k*-

* Research supported in part by National Science Foundation Grant MCS 80-07756.

0019-9958/83 $3.00

separable, and thus, isomorphism testing in this case is a generalization of the bounded valence case. Second, these graphs arise in an attempt to decompose graphs into k-connected components. Among other things Hopcroft and Tarjan (1972) show that graph isomorphism could be reduced to isomorphism of 3-connected graphs. It is open if graph isomorphism is reducible to isomorphism of 4-connected graphs. We believe that the results in this paper may be useful since we can directly handle those graphs which are nowhere 4-connected, i.e., 3-separable graphs.

The techniques used here combine the group theoretic ideas employed by Luks for testing isomorphism of graphs of bounded valence with the classic connectivity ideas in graph theory. Luks approximated the automorphism of a graph by computing the automorphism of induced subgraphs. The induced subgraphs are obtained by leveling the vertices according to how far they are from some edge. Here, we shall find other characteristic subsets of vertices determined by some edge and consider the induced hypergraph on these vertices. *So,* in a natural way we shall approximate the graph with a sequence of hypergraphs.

One of the interesting subproblems we solve enroute is finding the automorphisms of a hypergraph which induce an action on the vertices in some given group G (where G is in Γ_k).

The paper is divided into three sections: The preliminaries, gives the basic definitions and facts we will need. The second section gives the main group theoretic result we shall need. The third section gives the isomorphism test and the graph theoretic constructions.

1. PRELIMINARIES

Throughout this paper graphs will be denoted by G, H, K, groups by A, B, C, and sets by X, Y, Z. Graphs and hypergraphs may have multiple edges and the edges may be colored. The edges and vertices of G will be denoted by $E(G)$ and $V(G)$, respectively. An isomorphism is a surjective map which sends edges to edges, vertices to vertices, and preserves incidence and color. Groups will normally be permutation groups and they will act from the left. It can easily be shown that the isomorphism of G onto G' can be written as σA if G is isomorphic to G', where σ is an arbitrary isomorphism and A is the group of automorphisms of G. The properties of σA are so similar to a coset of A we shall call σA a ***coset.*** The isomorphism can be represented simply by a plus generators for A. Since any subgroup of S_n can be generated using only n log n elements or n^2 strong generators (Furst, **1980**) we have a compact representation for the isomorphisms, i.e., polynomial space in n.

A few simple remarks are necessary about the relationship between the hypergraph isomorphisms from G to G' and the induced action from vertices V to V' which we will write as $\sigma A \upharpoonright V$ ($\upharpoonright V$ means restricted to V.) Note that two elements $\beta, \gamma \in A$ are the same map on V if and only if $\beta^{-1}\gamma$ fixes V pointwise. If we let A, denote the elements of A which **fix** V, then the elements that act the same on V are left coset of A_V. Now, the group A_V has a simple form; it simply permutes edges with the same points of attachment. This means we can extend elements of $\sigma A \upharpoonright V$ to elements of σA almost arbitrarily. If $\beta \in \sigma A \upharpoonright V$, e is an edge of G, and $e, ,..., e_m$ are the edges of G which have the same points of attachment as e then we can extend as follows: Since β is an isomorphism $\beta(e)$ is equivalent to $m-1$ edges also, say $e'_1,..., e'_m$. **We** now arbitrarily send $e_1,..., e_m$ onto $e'_1,..., e'_m$. To obtain σA from $\sigma A \upharpoonright V$ we extend the generators of $\sigma A \upharpoonright V$ to $V \cup E$ and add generators for A_V. **To** find the isomorphisms we can replace multiple edges of G and G' with labeled single edges which keep track of edge multiplicity. In this case, the cosets σA and $\sigma A \upharpoonright V$ are isomorphic as groups. **A** graph or hypergraph is ***simple*** if it has no multiple edges of the same color. **A** graph is *common* if it has only two point edges.

Besides coloring edges to bookkeep symmetries we shall use cosets of groups which have very special properties, namely,

DEFINITION (Luks, **1980).** For $k \geqslant 2$, let Γ_k denote the class of groups A such that all the composition factors of A are subgroups of S,.

The importance of groups in Γ_k involves the special nature of their primitive actions. Recently Babai–Cameron–Palfy have shown that the primitive groups are of polynomial size in this case.

THEOREM 1 (Babai *et al.,* to appear). ***There is afunction $t(k)$ such that any primitive group $A \in \Gamma_k$ of degree n has order at most n^t.***

In Luks' paper (1980) he does not use this theorem, but instead analyzes the nature of p-Sylow subgroups of primitive groups in Γ_k. Luks' approach is very interesting and any implementation should consider it. Since the contribution of this paper consists of an analysis of the case when the group is not transitive, we shall present all algorithms using Theorem 1.

Any of the algorithms presented here can easily be extended to include Luks' p-Sylow subgroup ideas.

DEFINITION. σA is a coset of X onto Y if A subgroup of Sym(X) and a is a surjective map from X to Y.

Let G **be** a graph or hypergraph, say $G=(V,E)$, **and** $Y \in V$. **We** define the notion of a bridge and an induced hypergraph. We say two edges e and

e' of G are equivalent with respect to Y if there exists a path from e to e' avoiding points in Y.

DEFINITION. The induced graph Br of an equivalence class of edges of G with respect to Y is called a ***bridge,*** or a bridge of the pair (G, Y). The ***frontier*** of Br is the vertices of Br in Y. A bridge is trivial if it is a single edge.

Given the bridges we define the induced hypergraph.

DEFINITION. The ***hypergraph of the pair*** (G, Y) is the hypergraph (Y, E'), where the hyperedges E' are the frontiers of the bridges of (G, Y). Two bridges may have the same frontier and thus introduce multiple hyperedges. We shall denote this graph by $Hyper(G, Y)$.

A hypergraph can be also viewed as a bipartite graph. We introduce a new vertex for each hyperedge and connect an old vertex to a new hyperedge vertex if the edge contains the vertex. We shall call this graph the bipartite graph of (G, Y) denoted by Bipart (G, Y).

2. ISOMORPHISMS OF TWO HYPERGRAPHS IN A COSET

In this section we construct a polynomial time algorithm for the following problem:

Hypergraph Isomorphism in a Γ_k Coset

Input. Hypergraphs G and G', and a coset σA of $V(G)$ onto $V(G')$, where $A \in \Gamma_k$.

Find. Coset of isomorphisms of G onto G' in σA.

One natural application for hypergraph isomorphism in a Γ_k coset is in testing isomorphism of graphs of bounded valence. Here we test for isomorphisms sending some edge e onto some edge e'. The vertices are labeled by how far they are from e and e', respectively. Let G_i and G_i' be the induced graphs on vertices with labels $\leqslant i$. The edges between the vertices on the i and $i+1$ levels form a bipartite graph which we can also view as a hypergraph on vertices labeled $j \leqslant i$. If we have constructed the isomorphisms from G_i onto G_i' which send e to e' we can intersect this coset with the isomorphism of the hypergraph. This is in fact what Luks does but the algorithm presented uses exponential space if the hyperedges are allowed to have edge of arbitrarily high valence even if the coset is constrained to Γ_k. Using our solution to hypergraph isomorphism for Γ_k we will speed up the bounded valence algorithm by approximately the kth root of the previous time.

Luks has observed that combining the new bounded valence algorithm with work of Zemlyachenko and Babai (1981) one gets an $O(\exp(n^{1/2+\epsilon}))$ for $\varepsilon > 0$, algorithm for general graph isomorphism.

For graphs (having edges with only two points of attachment) the problem for Γ_k cosets was known to be polynomial.

Graph Isomorphism in a Γ_k Coset

Input. A coset σA, where $A \in \Gamma_k$, G and G' graphs.

Find. Coset of isomorphsm of G onto G' in σA.

Here one simply constructs the sets $\left[{V \atop 2}\right]$ and $\left[{V' \atop 2}\right]$, all pairs in V and V', respectively, considers σA as acting from $\left[{V \atop 2}\right]$ onto $\left[{V' \atop 2}\right]$ and colors the points according to whether they are edges or nonedges. This reduces graph isomorphism in a Γ_k coset to:

Color Isomorphism in a Γ_k Coset

Input. A Γ_k coset σA from V to V' and a coloring of $V \cup V'$.

Find. Coset of σA which preserves colors.

Luks (1980) gives a polynomial time algorithm for the color isomorphism problem in a Γ_k coset which in turn gives a polynomial time algorithm for graph isomorphism in a Γ_k coset. Let $\mathrm{Iso}(G, G', \sigma A)$ be the isomorphisms from G to G' in σA and let $\mathrm{ISO}(G, G', \sigma A)$ be the proposed algorithm for the graph isomorphism in a Γ_k coset problem.

Given a hypergraph G and a subset of vertices X we define three technical but important derived graphs. Recall that hypergraphs have colored edges, i.e., $G=(V,E)$, where E is disjoint union $\{E_1,\ldots,E_l\}$ and E_i is the set of hyperedges of G with color i.

We can partition the edges of E_i according to whether they are contained in X, contained in $V-X$, or straddle X and $V-X$. We will assume that G has no multiple edges of the same color; G is simple. This gives the partition

(i) $E_{i1} = \{e \in E_i \mid e \cap X = e\}$,

(ii) $E_{i2} = \{e \in E_i \mid e \cap X = \varnothing\}$,

(iii) $E_{i3} = \{e \in E_i \mid e \cap X \neq \varnothing, e\}$.

Abusing notation, let $E_{i3} \cap X = \{e \cap X \mid e \in E_{i3}\}$ for $1 \leqslant i \leqslant l$.

We now define the first two graphs.

DEFINITION. The ***restriction*** of G with respect to X is $R(G,X)=(V',E')$, where $V'=V$ and

$$E'=(E_{,1},\ldots,E_{,,}, E_{13}\cap X,\ldots,E_{,,}\cap X).$$

DEFINITION. The ***partition*** of G with respect to X is $\mathrm{Part}(G,X)=(V'',E'')$, where $V''=V$ and

$$E''=(E_{11},\ldots,E_{l1}, E_{12},\ldots,E_{l2}, E_{13}\cap X,\ldots,E_{l3}\cap X,$$
$$E_{13}\cap\bar{X},\ldots,E_{,,}\cap\bar{X}),$$

where $\bar{X}=V-X$.

Note that the set $E_{i3}\cap X$ as defined does not have multiple edges. Thus the graph's restriction and partition have no multiple edges when G does not. We next define a simple graph.

DEFINITION. The ***join*** of G with respect to X, denoted $\mathit{Join}(G,X)$ will be (V''',E'''), where $V'''=V\cup E''$, i.e., the vertices and edges of $\mathrm{Part}(G,A)$. $E'''=\{(e\cap X, e\cap\bar{X})\mid e\in E_{i3}$ for $1\leqslant i\leqslant l\}$, e.g., $(e\cap X, e\cap\bar{X})$ is the pair of vertices $e\cap X$ and $e\cap\bar{X}$ which are edges of $\mathrm{Part}(G,X)$ and thus vertices of $\mathrm{Join}(G,X)$.

We shall often drop the reference to X in the denotation of these graphs when there is no risk of confusion.

As before, let $\mathrm{Iso}(G,G',\sigma A)$ be the isomorphisms (sending edges to edges and vertices to vertices) which induce elements in σA. Let X be an A-stable subset of V and let $X'=\sigma(X)$. The join, restriction, and partition are all with respect to X and X' for G and G', respectively. Using this idea the coset:

$$\mathrm{Iso}(\mathrm{Join}(G),\mathrm{Join}(G'),\mathrm{Iso}(\mathrm{Part}(G),\mathrm{Part}(G'),\sigma A))$$

is well defined. Note that this coset is not acting on the same sets as $\mathrm{Iso}(G,G',\sigma A)$. But since G has no multiple edges the edges of G which straddle X correspond to $\mathrm{Join}(G)$. While the edges which do not straddle X correspond to edges of $\mathrm{Part}(G)$ with the appropriate color. A similar remark is true for G'. This allows us to state

LEMMA 1. *$\mathrm{Iso}(G,G',\sigma A)$ is equivalent to*

$$\mathrm{Iso}(\mathrm{Join}(G),\mathrm{Join}(G'),\mathrm{Iso}(\mathrm{Part}(G),\mathrm{Part}(G'),\sigma A)).$$

Proof. Suppose $g\in$ LHS. Since $g\in\sigma A$ and X is A-stable, g will preserve the refinement of the color classes. We consider the edges which do not straddle X first. Since these edges lie in $\mathrm{Part}(G)$ and $\mathrm{Part}(G')$ essentially unchanged, we can apply g to them in a natural way. If e is an edge which straddles X, then $e \cap X$ and $e \cap X'$ are in Part(G) and Part(G'), respectively. Now, $g(e) = e'$ for some unique e' in G *so* $g(e \cap X) = g(e) \cap g(X) = e' \cap X'$ and $g(e \cap \bar{X}) = e' \cap \bar{X}'$. Since $(e \cap X, e \cap \bar{X})$ and $(e' \cap X', e' \cap \bar{X}')$ are unique edges, we can in well-defined way iet one be sent to the other by g. Suppose $g \in$ RHS and e is some edge of G. If e does not straddle X then the image of e will simply be the image of e in Part(G). Here we use the fact that the color of e differs from all the edges which straddle X. In the case e straddles X the edge $(e \cap X, e \cap \bar{X})$ is sent to some edge $(e' \cap X', e' \cap \bar{X}')$ by g. Since $g \in$ Iso(Part(G), Part(G'), σA) g must send $e \cap X$ to $e' \cap X'$ and $e \cap \bar{X}$ to $e' \cap \bar{X}'$. *So* we can let $g(e) = e'$.

We need one more easy lemma.

LEMMA 2. (Part(G), Part(G'), σA) *is equivalent to*

$$\mathrm{Iso}(R(G, X), R(G, X')\,)\,\mathrm{Iso}(R(G, \bar{X}), R(G, \bar{X}'), \sigma A)).$$

We define the main procedure for this section Iso(G, G', X, σA), where G, G' are hypergraphs, X is an A-stable subset of V, σA is a coset from V to V', and the edges of $G(G')$ contains only points of $X(X' = \sigma(X))$. The procedure should return with the coset of isomorphism from G to G' which induce elements in σA.

Procedure. ISO(G, G', X, σA).

Begin

(1) *If* $|X| = 1$ *then* $\begin{cases} \sigma A & \text{if } \sigma \text{ is an isomorphism,} \\ \varnothing & \text{otherwise.} \end{cases}$ (*)

(2) *If* A is not transitive on X say X_1, X_2 is a partition of X into stable subsets *then* set

$$\begin{aligned} \sigma' A' &:= \mathrm{Iso}(R(G, X_1), R(G', X_1'), \sigma A); \\ \sigma'' A'' &:= \mathrm{Iso}(R(G, X_2), R(G', X_2'), \sigma' A'); \\ \textbf{\textit{return}}\ & \mathrm{Iso}(\mathrm{Join}(G, X_1), \mathrm{Join}(G', X_1'), \sigma'' A'') \end{aligned} \quad (**)$$

(where Iso is computed using the solution to graph isomorphism in a Γ_k coset).

(3) If A is transitive on X then

(a) Find a block system $X_1, \ldots, X_m$ of X with minimum number of blocks.

(b) Find subgroup B which stabilizes these blocks and coset representatives of B in A, say $\sigma_1,\ldots,\sigma_s$. Compute $\mathrm{ISO}(G, G', X, \sigma\sigma_1 B),\ldots, \mathrm{ISO}(G, G', X, \sigma\sigma_s B)$. Write result as coset.

End.

The correctness of ISO follows from Lemmas 1 and 2, and standard techniques. To analyze the running time of ISO we look at the recursive control structure. Note that we have a tree of recursive calls where the leaves are either (*) or (**). We can view (**) not a recursive call but an added processing cost at the node of the tree. Then, the only leaves are (*). Let $L(n)$ be the number of leaves as a function of $n = |X|$. The recurrence relation for the leaves is $L(n) = m^{t+1} L(n/m)$ and $L(1) = 1$. So $L(n) = n^{t+1}$, where t is from Theorem 1. Since the coset at each node is polynomial the running time of the algorithm is polynomial. We state this as

THEOREM 2. *ISO is a polynomial time algorithm for the hypergraph isomorphism in a Γ_k coset for fixed k.*

3. ISOMORPHISM FOR PAIRWISE k-SEPARABLE GRAPHS

The bounded valence isomorphism algorithm of Luks is based in part on the simple fact that the edge stabilizer of a connected graph of valence k is in Γ_{k-1}. Here we shall need a similar statement about pairwise k-separable graphs. But, for pairwise k-separable graphs we must also assume that they are 2-connected.

THEOREM 3. *If G is 2-connected and pairwise k-separable for $k \geqslant 3$ then the edge stabilizer of G is in Γ_{k-1}.*

This theorem is an interesting exercise in the case $k = 3$. A simple proof of this theorem would be interesting. We prove the theorem via a general discussion of the main result of this section.

THEOREM 4. *Isomorphism for pairwise k-separable graphs is polynomial time testable for fixed k.*

Throughout this section we shall only discuss the problem of finding the stabilizer of an edge, i.e., automorphisms. One can formally reduce the isomorphism problem in this case to the edge stabilizer problem in this case. On the other hand, one can prove what is to follow directly for the isomorphism problem. In any case, the content seems to be in the edge stabilizer problem.

In the bounded valence algorithm, one used the fact that not only is the edge stabilizer in Γ_{k-1} for connected graphs of valence k but any stabilizer of a connected subgraph is also in Γ_{k-1}. For graphs which are 2-connected and pairwise k-separable it seems much harder to decompose the graph such that subgraphs are 2-connected. We circumvent this problem by considering a new approximation technique. The automorphisms of a simple graph G which fix some edge $e=(y_1, y_2)$ of G will be determined by finding a sequence of subsets of V; $\{y_1, y_2\} = Y_1 \subseteq Y_2 \subseteq \cdots \subseteq Y_s = V$ and inductively computing $\mathrm{Auto}_e(\mathrm{Hyper}(G, Y_i))$, where $\mathrm{Auto}_e(H)$ is the group of automorphism of H which **fix** e.

The subsets Y_i must be chosen satisfying two properties. The first of these two properties ensures that the automorphisms at the ith stage are sufficient to compute the automorphisms at the $i+1$ stage. That is, we need that $\mathrm{Auto}_e(\mathrm{Hyper}(G, Y_{i+1})) \upharpoonright Y_i$ are contained in $\mathrm{Auto}_e(\mathrm{Hyper}(G, Y_i))$. In fact, the Y_i satisfy a stronger property which we shall call characteristic.

Definition. Let e be an edge of G and $Y_1 = \{x, y\}$, the end points of e, then the set Y, where $Y_1 \subset Y \subseteq V$ is ***characteristic*** with respect to e if for all $Y \subseteq X \subseteq V$ Y is stabilized by the group $\mathrm{Auto}_e(\mathrm{Hyper}(G, X))$.

If Y is characteristic and $Y \subseteq X \subseteq V$ then $A = \mathrm{Auto}_e(\mathrm{Hyper}(G, X)) \upharpoonright Y \subseteq \mathrm{Auto}_e(\mathrm{Hyper}(G, Y))$. This follows by noting that A preserves the equivalence relation defined on the edges of $\mathrm{Hyper}(G, X)$ by Y. Thus, A sends bridges of Y to bridges of Y and preserves the bridge-frontier relation.

The second property will allow us to extend Y_i, since it will ensure that the groups are in Γ_{k-1}.

Definition. The subset $Y \subseteq V$ is ***consistent*** if the graph $\mathrm{Bipart}(G, Y)$ is 2-connected.

Note that if G is 2-connected then Y is consistent if no hyperedge of $\mathrm{Hyper}(G, Y)$ is critical. If Y is consistent then a bridge Br of (G, Y) will contain at least two points in its frontier, and the frontier points of Br will be $k-1$ separable in Br.

We digress for a moment into a discussion of vertex separators for a pair of distinct and not adjacent vertices x and y of some hypergraph H. Let T and T' be subsets of vertices of H disjoint from x and y which separate x from y. We shall say $T \leqslant T'$, with respect to the pair (x, y) if the size of $T <$ the size of T' or if the size of $T =$ size of T' and the bridge containing x in (H, T) is contained in the corresponding bridge of (H, T'). This defines a partial order on the separators of x and y. We next show the well-known fact that there is a minimum separator for the pair (x, y).

Lemma 3. ***The pair*** (x, y) ***of*** H ***has a unique minimal separator.***

Proof. We give a proof for the sake of completeness. Let T be a minimal separator and T' be an arbitrary separator of the pair (x, y). We show that $T \leqslant T'$. If the size of $T <$ the size of T' we are done. *So,* let the size of T and T' be t. Let Br, and Br, be the bridges of x and y, respectively, in (H, T). If $V(\mathrm{Br}_x) - T$ contains no points of T' we are done. *So* suppose T' contains $k > 0$ points, T_1', in $V(\mathrm{Br}_x) - T$. Now the bridge of X in (Br_x, T_1') must contain at least $t - k + 1$ points of T, say T_1, for otherwise the points $T, \cup T_1'$ form a separator for (x,y) which is less than T. Thus, we can reach $t - k + 1$ points on T from x avoiding T'. On the other hand, the bridge Br,, in (H, T) contains at most $t - k$ points of T'. Thus, by similar arguments we can reach at least k points of T from y avoiding T'. Since $(t - k + 1) + k > t$ we must be able to find a path from x to y avoiding T'. This is a contradiction. Thus $T \leqslant T'$.

We need the previous lemma to apply to intermediate hypergraphs.

LEMMA 4. *If T is a minimum separator of (x,y) in H then T is also the minimum separator of (x,y) in* $\mathrm{Hyper}(H, X)$ *where X contains x, y and T.*

The lemma follows by arguments similar to Lemma 3.

Using Lemma *3* the ***decomposition*** of a pair (G, Y) is Y plus the union over all minimum separators for triples (x, y, Br), where Br is a nontrivial bridge of (G, Y) and x, y are distinct points in its frontier. We shall denote this *set* by ***Decomp(G, Y).*** The Y_i's are defined as follows: (1) $Y_1 = \{y_1, y_2\}$, where y_1 and y_2 are end points of some edge e; (2) $Y_{i+1} = \mathrm{Decomp}(G, Y_i)$ for $i < |V|$.

It follows from the definitions that (C, $Y,$) has the following three properties:

(1) (G, $Y,$) is consistent,

(2) (G, $Y,$) is .characteristic,

(3) $\mathrm{Auto}_e(G, Y,)$ is a Γ_{k-1} group.

We show that these three properties are true inductively for (G, $Y,$).

LEMMA 5. ***If Y is consistent in*** G ***then*** $\mathrm{Decomp}(G, Y)$ ***is consistent.***

Proof: Let $\mathrm{Decomp}(G, Y) = Y'$, Br' be a bridge of (G, Y'), and Br be the bridge in (G, Y) containing Br'. Letting X' be the frontier of Br' we must show that there is path between any two vertices of X' avoiding Br'. It will suffice *to* find a path from any vertex $x' \in X'$ to the frontier of Br avoiding Br$'$, since Y is consistent. Since x' is in Y' it is either in Y, in which case we are done, or x' is in a separator, say T. Suppose $x' \in T$. Now T partitions Br **up** into bridges at least two of which contain vertices of Y. One of these intermediate bridges contains Br'. So by avoiding this intermediate bridge we can find a path to Y.

LEMMA 6. ***The set Y_i is characteristic in G.***

Proof: The proof is by induction on i. For $i = 1$ it is clear. Suppose that Y_i is characteristic we shall show that Y_{i+1} is characteristic. Let X be a set such that $Y_{i+1} \subseteq X \subseteq V$ and let $\sigma \in \mathrm{Auto}_e(\mathrm{Hyper}(G, X))$. We must show that σ stabilizes Y_{i+1}. Let $H = \mathrm{Hyper}(G, X)$. Since σ stabilizes Y_i, σ can be viewed as an automorphism of the graph $\mathrm{Hyper}(H, Y_i)$. Now, the minimum separators for (G, Y_i) will be minimum separators for (H, Y_i) by Lemma 4. So σ will take bridges of (H, Y_i) to bridges and minimum separators to minimum separators. Thus σ will stabilize Y_{i+1}.

To show that $\mathrm{Auto}_e(\mathrm{Hyper}(G, Y_i))$ is in Γ_{k-1}, as well as to construct this group, we will use an intermediatry graph, say In,.

The hypergraph In, equals (V, E) where

$$V = \{(\mathbf{x}, y, \mathrm{Br}) \mid \mathrm{Br} \text{ is a bridge of } (G, Y_i) \text{ and } \mathbf{x}, y \text{ are distinct frontier points of } \mathrm{Br}\},$$

$$E = \{e_z \mid z \in Y_{i+1} - Y_i\},$$

where

$$e_z = \{(\mathbf{x}, y, \mathrm{Br}) \mid z \text{ is in the minimum separator of } \mathbf{x} \text{ and } y \text{ in } \mathrm{Br}\}.$$

Let $A, = \mathrm{Auto}_e(\mathrm{Hyper}(G, Y_i))$. By Lemma 6 there is a natural homomorphism from A_{i+1} to A, which takes minimum separators to minimum separators. On the other hand, A_i acts on the triples (x, y, Br), the vertices of In,. Thus, there is a natural homomorphism of A_{i+1} into $\bar{A}_i = \mathrm{Auto}(\mathrm{In}_i) \cap A_i$ whose kernal fixes Y_{i+1}. So $A,, ,\restriction Y_{i+1}$ can be viewed as a subgroup of $\bar{A}_i$. To show that $A,, ,$ is in Γ_{k-}, we need only show that $\bar{A}_i$ is in Γ_{k-1}, if A_i is in Γ_{k-1}, and that $A_i \restriction Y_i$ is Γ_{k-1} implies that A_i in Γ_{k-1}. Both of these statements reduce to showing that the graphs In, and $\mathrm{Hyper}(G, Y_i)$ have edge multiplicity at most $k - 1$. Each triple $(\mathbf{x}, y, \mathrm{Br})$ of the vertices of In_i has a cut set of size at most $k - 1$. Thus the valence of In_i is at most $k - 1$, and therefore, the multiplicity is at most $k - 1$.

We state the fact that the edge multiplicity is bounded in

LEMMA 7. ***If G is a simple 2-connected graph which is pairwise k-separable, e is an edge of G withfrontier $\{y,, y_2\}$, and $\{y_1, y_2\} \subseteq Y \subseteq V$ then*** $\mathrm{Hyper}(G, Y)$ ***is a k-bond (i.e., $Y = \{y,, y_2\}$ and*** $\mathrm{Hyper}(G, Y)$ ***contains k edges between $y, ,y_2$) or the edge multiplicity is at most $k - 1$.***

Proof: The edge multiplicity in Hyper(G, Y) can be at most k since G is pairwise k-separable. Suppose that the vertices $x_1,..., x_t \in Y$ share k multiple edges in Hyper(G, Y). Since the k edges form k vertex disjoint paths between any two of the x_i's, there may be no other disjoint paths between the x_i's. Thus the vertices are in disjoint components in Hyper(G, Y) minus the k edges. If the component containing x_i contain some other point y then x_i would separate y from x_j, $i \neq j$ in Hyper(G, Y). But x_i would also be a separator in G contradicting the hypothesis that G is 2-connected. So Hyper(G, Y) consists of the t vertices and k multiple edges. Since one of the edges is the two point edge e, Hyper(G, Y) is a k-bond.

By the lemma either Hyper(G, Y_i) has edge multiplicity $k-1$ or $\mathrm{Auto}_e(\mathrm{Hyper}(G, Y_i)) \subseteq S_{k-1}$ and thus trivially in Γ_k.

We give the algorithm explicitly for computing the automorphism stabilizing e in G, where G is 2-connected and pairwise k-separable:

(1) Compute $Y_1,..., Y_t$ for G with respect to e using a maximum flow algorithm.

(2) Inductively compute $\bar{A}_i$ from A_i using ISO algorithm from Section 2.

(3) Inductively compute A_{i+1} from $\bar{A}_i$ using ISO algorithm from Section 2.

(4) output A,.

Since each step takes only polynomial time and $t \leqslant n$ (the number of vertices of G), the algorithm will run for polynomial time.

RECEIVED: September 22, 1982; ACCEPTED: July 28, 1983

REFERENCES

BABAI, L. (1979), On the isomorphism problem, *In* "Proceedings, FCT Conf. Poznan-Kornak."

BABAI, L. (1981), Moderately exponential bound for graph isomorphism, *in* "Proceedings Conf. on Foundation of Computation Theory, Szeget."

BABAI, L., "Monte-Carlo Algorithms in Graph Isomorphism Testing", submitted to SJCOMP.

BOOTH, K. AND COLBOURN, C. (1979), "Problems Polynomially Equivalent to Graph Isomorphism," TRCS-77-04, Department of Computer Science, Univ. of Waterloo.

BABAI, L., CAMERON, P. J., AND PALFY, P. P., "On the Order of Primitive Permutation Groups with Bounded Nonabelian Composition Factors," in preparation.

BABAI, L., ERDOS, P., AND SELKOW, S. (1980), Random graph isomorphism, *SIAM J.* Comput. **9,** 628–635.

BABAI, L. AND GRIGORYEV, D., "Isomorphism Testing for Graphs with Bounded Eigenvalue Multiplicities," submitted for publication.

CAMERON, P. J. (1981), Finite permutation groups and finite simple groups, Bull. London Math. Soc. **13,** 1–22.

FURST, HOPCROFT, J. E., AND LUKS, E. M. (1980), "A Subexponential Algorithm for Trivalent Graph Isomorphism," Tech. Report 80-426, Computer Science, Cornell Univ.

FILOTTI, I. S. AND MAYER, J. N. (1980), A polynomial-time algorithm for determining the isomorphism of graphs of fixed genus, *in* "Proceedings, 12th ACM Symp. Th. Comp," pp. 236–243.

FILOTTI, I. S., MILLER, G. L., AND REIF, J. (1979), "On Determining the Genus of Graph in $O(V^{0(g)})$ Steps."

GAREY, M. AND JOHNSON, D. (1979), "Computer and Intractability, A Guide to the Theory of *NP*-Completeness," San Francisco.

GOLDBERG, M., A nonfactorial algorithm for testing isomorphism of two graphs, ***Discrete Math.*** in press.

HOPCROFT, J. E. AND TARJAN, R. E. (1972), Isomorphism of planer graph (working paper), *in* "Complexity of Computer Computations" (R. E. Miller and J. W. Thatcher, Eds.), pp. 131–152, Plenum, New York.

HOFFMANN, C. M. (1980), Testing isomorphism of cone graphs, *in* "Proceedings 12th Symp. Th. Comp.," pp. 244–251.

KANNAN, R. AND LIPTON, R., "Computational Group Theory," manuscript.

LICHTENSTEIN, D. (1980), Isomorphism for graphs embeddable on the projective plane, *in* "Proceedings 12th ACM Symp. Th. Comp.," pp. 218–224.

LUKS, E. M. (1980), Isomorphism of bounded valence can be tested in polynomial time, *in* "Proceedings 21st Annual Symposium of Foundations of Computer Science," IEEE, New York.

MATHON, R. (1979), A note on the graph isomorphism counting problem, ***Inform.*** Process. Lett. **8** (3), 131–132.

MILLER, G. L. (1979), Graph isomorphism, general remarks. ***J. Comput. System Sci.*** **18**, 128–141.

MILLER, G. L. (1980), Isomorphism testing for graphs of bounded genus (working paper), *in* "Proceedings, 12th ACM Symposium on Theory of Computing," pp. 225–235.

READ, R. AND CORNEIL, D. (1977), The graph isomorphism disease, **J. *Graph. Theory,*** **1**, 339–363.

VALIANT, L. G. (1979), The complexity of enumeration and reliability problems, ***SZAM*** **J.** ***Comput.*** **18**, (3), 410–421.

HOPCROFT, J. E. AND TARJAN, R. E. (1973), Dividing a graph into triconnected components, ***SIAM*** **J.** ***Comput. 2*** (3), 294–303.

LICHTENSTEIN, D. "Decomposing a Graph into Its 4-Connected Components," manuscript.

ANGLUIN, D. (1980), On counting problems and the polynomial time hierarchy, ***Theoret. Comput. Sci.*** **12**, 161–173.

BABAI, L., GRIGORYEV, D. V., AND MOUNT, D. M. (1982), Isomorphism of graphs with bounded eigenvalue multiplicity, in "Proceedings, 14th ACM Symp. Th. Comp."

MILLER, G. L., "Isomorphism Testing of k-Contractible Graphs," to appear.

Printed by the St. Catherine Press Ltd., Tempelhof 41, Bruges, Belgium