ISOMORPHISM TESTING AND CANONICAL FORMS FOR k-CONTRACTABLE GRAPHS
(A Generalization of Bounded Valence and Bounded Genus)

Gary L. Miller*
Department of Mathematics
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract. This paper includes polynomial time isomorphism tests and
canonical forms for graphs called k-contractable graphs for fixed k.
The class of k-contractable graphs includes the graphs of bounded valence
and the graphs of bounded genus. The algorithm uses several new ideas
including: (1) it removes portions of the graph and replaces them with
groups which are used to keep track of the symmetries of these portions;
(2) it maintains with each group a tower of equivalence relation which
allows a decomposition of the group. These towers are called a tower of
Γk actions. It considers the canonical intersection of groups.

1. INTRODUCTION

The author, and independently other researchers [FM80, L180, M180],
have presented polynomial time algorithms for isomorphism testing of
graphs of bounded genus. These algorithms are based on fairly compli-
cated analyses of embeddings of graphs on two dimensional surfaces.
Since then, Luks has presented a polynomial time algorithm for isomorph-
ism testing of graphs of bounded valence [Lu80]. The ideas used in the
bounded valence algorithm are very appealing. They showed relationships
between computational group theory and graph isomorphism. The existence
of a polynomial time algorithm which in a natural way tests isomorphism
of graphs of bounded valence and bounded genus has been an open question
[Ba81]. We show that the class of graphs, called the k-contractable
graphs, contains the graphs of bounded genus. They trivially contain
the graphs of bounded valence. Therefore, these graphs form a common
generalization of the two classes. We give a polynomial time algorithm
for testing isomorphism of these graphs.

More recently, several authors [BL83, FSS83] have shown that graphs
of bounded valence have p-time constructable canonical forms. We also
show how these ideas can be extended to the k-contractable graphs.

We give the first of several definitions of k-contractable graphs.
Later definitions of k-contractable graphs will enlarge the class of
graph which will also be called k-contractable for each k. Consider the
following three operations on a graph, G.

*This work partially supported by NSF Grant MCS 800756-A01.
The first operation, removing leaves, simply removes all leaves and their associated edges. The second operation, removing multiple vertices, identifies vertices of valence two which have the same neighbors. The third operation, contracting edges of valence $\leq k$, simultaneously identifies the end points of edges common to two vertices whose valence is $\leq k + 1$. It then removes selfloops and multiple edges.

Definition 1.1. The graph $G$ is $k$-contractable if the application of the above three operations eventually reduces $G$ to a point.

The definition seems to be dependent on the order in which one applies these three operations. Nevertheless, in the last section we show that any graph which is fixed by all three operations has its genus bounded below by $\Omega(k)$. Diagram 1 describes an infinite periodic planar graph. From this graph we can construct arbitrarily large graphs of genus 1 which are not 10-contractable. These graphs are 6-contractable using a slightly stronger definition of the third operation.

The paper consists of four other sections. Section 2, the preliminaries, contains basic definition plus the notation of a tower of $\Gamma_k$ actions which will be used throughout. Section 3, Canonical ordering and set of orderings, contains the basic group theoretic algorithms used in the isomorphism tests. Section 4 includes the notation of a graph where the symmetries at a vertex are not arbitrary but restricted to a group. This last notion will be used in section 5 to decompose a class of graphs where at intermediate stages the graphs are those with restricted vertex symmetries. The graph for which this contraction procedure works will be called the $k$-contractable graphs. Section 6 shows that the $k$-contractable graph contains the graphs of genus $ck$ for some $c > 0$.

2. PRELIMINARIES

2.1 Graph Theoretic Preliminaries

Throughout this paper graphs will be denoted by $G$, $H$, and $K$; groups by $A$, $B$, and $C$; and sets by $X$, $Y$, and $Z$. Graphs may have multiple edges but no selfloops. It will be important that they are allowed to have multiple edges. The edges and vertices of $G$ will be denoted by $E(G)$ and $V(G)$, respectively. The edges common to some vertex $v$ or set of vertices will be denoted by $E(v)$. Let $\sim$ denote the multiple edge equivalence relation on $G$, i.e. $e \sim e'$ if $e$ and $e'$ are common to the same points. The valence of a given vertex $v$ will be the number of vertices adjacent to $v$, i.e. the number of edges ignoring multiplicity. Let $G$ be a graph
and $Y \subseteq V$. We say two edges $e$ and $e'$ of $G$ are $Y$ equivalent if there exists a path from $e$ to $e'$ avoiding points of $Y$.

Definition 2.1. The graph $Br$ induced from an equivalence class of $Y$-equivalent edges will be called a bridge, or a bridge of the pair $(G,Y)$. The vertex frontier of $Br$ is the vertices of $Br$ in $Y$, while the edge frontier of $Br$ is the edges of $Br$ common to the vertex frontier. A bridge is trivial if it is a single edge.

The main graph theoretic construction we shall use is contracting nonfrontier edges to a point.

Definition 2.2. If $Y \subseteq V$, the vertices of $G$, then $\text{Contract}(G,Y)$ will be the graph obtained from $G$ by identifying the nonfrontier (internal) vertices of $Br$ for each bridge $Br$ of $G$, and removing selfloops. If $Y = \emptyset$ then $\text{Contract}(G,Y)$ is a single vertex.

Let $Y_k$ be the vertices of $G$ with valence greater than $k$. The intuition behind the $k$-contractable graphs is it is those graphs for which the successive application of $\text{Contract}(G,Y_k)$ yields a single point. Here we keep track of the symmetries of the bridges with groups.

We attempt to render this idea.

2.2 Group Theoretic Preliminaries

Let $\text{Sym}(X)$ denote the group of all permutations of $X$. We let $S_n$ denote the symmetric group on a set of size $n$. The group $A$ is a permutation group on $X$ if $A \subseteq \text{Sym}(X)$. The degree of $A$ is $|X|$, while the order is $|A|$. Let $\pi$ be an equivalence relation on $X$. Let $A(\pi)$ denote the subgroup of $A$ which stabilizes the equivalence classes $X/\pi$ of $\pi$, i.e. $A(\pi) = \{a \in A | x \pi a(x) \text{ for all } x \in X\}$. If $Y \subseteq X$, we let $Y$ denote the relation $\{xy | x, y \notin Y \text{ or } x = y\}$. Thus, $A(Y)$ is the subgroup of $A$ which fixes $Y$ pointwise, while $A[Y]$ is the subgroup which stabilizes $Y$ we let $id$ denote the empty relation. We say $A$ preserves $\pi$ if $x \pi y$ implies $a(x) \pi a(y)$ for all $a \in A$. The subgroup of $A$ preserving $\pi$ we will denote by $A(\pi)$. $A$ is primitive if it only preserves the trivial equivalence relations $X$ and id. We say the relation $\pi$ contains $\pi'$ if $x \pi' y$ implies $x \pi y$ for all $x$ and $y$ in $X$, denoted by $\pi' \leq \pi$. If $\pi$ is an equivalence relation then $X/\pi$ denotes the equivalence classes of $\pi$. The restriction of an equivalence relation $\pi$ to $Y$ is denoted by $\pi|Y$. Formally, $\pi|Y$ is defined by $x \pi|Y y$ if $x, y \notin Y$ or $x, y \in Y$ and $x \pi y$. If $Y \subseteq X$ then $Y/\pi$ are the equivalence classes of $\pi$ restricted to $Y$.

The equivalence classes of the relation $\pi$ defined by, $x \pi y$ if for
some $x \in A \cap \langle x \rangle = y$, are called the orbits of $A$. The induced action of $A$ on some orbit $Y$ is the image of $A$ in $\text{Sym}(Y)$ which we identify with the quotient group $A/A(Y)$. In general we shall let $A \triangleright Y$ denote the faithful action of $A[Y]$ on $Y$, i.e. $A[Y]/A(Y)$.

An isomorphism is a surjective map which sends edges to edges and vertices to vertices, and preserves incidence and other possible structure. Groups will be permutation groups and they will act from the left.

### 2.3 Towers of $\Gamma_k$-Actions

Throughout this paper we shall either restrict the groups considered or the way they may act.

**Definition 2.3.** For $k \geq 2$, let $\Gamma_k$ denote the class of groups $A$ such that all the composition factors of $A$ are subgroups of $S_k$.

We shall use the following fact about the primitive actions of $\Gamma_k$ groups.

**Theorem 2.4 [BCP ta].** There is a function $\lambda(k)$ such that any primitive action of $A \in \Gamma_k$ of degree $n$ has order at most $n^{\lambda}$.

The main theorem seems to require that we allow all groups but restrict the way in which they can act. Even in the case of cubic graphs the edge stabilizer is a 2-group but the full group may be arbitrary.

**Definition 2.5.** A group $A$ acting on a set $X$ is a $\Gamma_k$-action if for all $x \in X$ the subgroup $A(x) \in \Gamma_k$.

We extend the notation of a $\Gamma_k$-action to a tower of such actions. The sequence $(\pi_0, \ldots, \pi_t)$ is a tower of equivalence relations on $X$ if $X = \pi_0 \supset \ldots \supset \pi_t = \text{id}$. We shall often write a tower as $(\pi_1, \ldots, \pi_t)$ where it is understood that $\pi_0 = X$. This gives a useful generalization of $\Gamma_k$-actions.

**Definition 2.6.** $(A, \pi_0, \ldots, \pi_t)$ is a tower of $\Gamma_k$-actions if:

1. the sequence $(\pi_0, \ldots, \pi_t)$ is a tower of equivalence relations on $X$;
2. $A \subseteq \text{Sym}(X)$;
3. $A$ preserves $\pi_i$ for $0 \leq i \leq t$;
4. for each $x \in X/\pi_i$, $0 \leq i \leq t$, the action of $A[Y]$ on $X/\pi_{i+1}$ is a $\Gamma_k$-action.

We write (4) more formally.
(4') If $S \in X_{\pi_i}$ and $T \subseteq S_{\pi_{i+1}}$ then $A[T]/A(\pi_{i+1} \setminus S) \in \Gamma_k$.

It is easy to see that (4) and (4') are equivalent since the natural homomorphism from $A[T]$ into $\text{Sym}(S/\pi_{i+1})$ has kernel $A(\pi_{i+1} \setminus S)$. We shall prove some simple closure properties about towers of $\Gamma_k$ actions. We state them as lemmas.

Lemma 2.7. If $(A, \pi_0, \ldots, \pi_t)$ is a tower of $\Gamma_k$ actions and $B \subseteq A$ then $(B, \pi_0, \ldots, \pi_t)$ is a tower of $\Gamma_k$ actions.

Proof. It is clear that $(B, \pi_0, \ldots, \pi_t)$ satisfies the first three conditions. We show it satisfies (4'). Let $S \in X_{\pi_i}$ and $T \in S_{\pi_{i+1}}$ we must show that $B[T]/B(\pi_{i+1} \setminus S) \in \Gamma_k$. But, $B[T] \supseteq A[T]$ and $B(\pi_{i+1} \setminus S) = B[T] \cap A(\pi_{i+1} \setminus S)$. So by the Second Isomorphism Theorem and the Correspondence Theorem $B[T]/A(\pi_{i+1} \setminus S)$ is isomorphic to a subgroup of $A[T]/A(\pi_{i+1} \setminus S)$. The latter quotient group is in $\Gamma_k$. Therefore, the first quotient group is in $\Gamma_k$, since $\Gamma_k$ is closed under taking subgroups.

Lemma 2.8. If $(A, \pi_0, \ldots, \pi_t)$ is a tower of $\Gamma_k$ actions and $X \subseteq A = \pi_0$ then $(A \upharpoonright Y, \pi_0 \upharpoonright Y, \ldots, \pi_t \upharpoonright Y)$ is a tower of $\Gamma_k$ actions.

Proof. Since $A \upharpoonright Y = A[Y] \upharpoonright Y$ and by the previous lemma, towers of $\Gamma_k$ actions are closed under taking subgroup. We may assume that $A = A[Y]$. Let $S' \subseteq Y_{\pi_i}, 0 \leq i \leq t$, and $T' \subseteq S'_{\pi_{i+1}}$. Since $S', T' \neq \emptyset$ and they are subsets of equivalence classes of $\pi_i$ and $\pi_{i+1}$ respectively, there exists unique elements $S \in X_{\pi_i}$ and $T \in S'_{\pi_{i+1}}$ containing $S'$ and $T'$, respectively. We must show that $A[T']/A(\pi_{i+1} \setminus S') \in \Gamma_k$. We have the following chain of inclusions

$$A[T'] = A[T'] \supseteq A(\pi_i \upharpoonright S') \supseteq A(\pi_i \upharpoonright S).$$

So our quotient is a section of a $\Gamma_k$ group and thus $\Gamma_k$.

3. CANONICAL ORDERING OF SETS AND SET OF ORDERINGS

3.1 Canonical Forms

As in [BL83, FSS83] we say a function $\text{CF}: K + K$, where $K$ is a class of graphs, is a canonical form for $K$ if:

1. for $G$ in $K$, $\text{CF}(G) = G$
2. for $G, H$ in $K$, $G \preceq H$ if and only if $\text{CF}(G) = \text{CF}(H)$.

In the previous papers, they considered linearly ordering the vertices as a canonical form or presentation. We shall order edges. For each edge $e$ we shall consider it as oriented, i.e. $e^{-1}$, the reverse of $e$,
is distinct from e. Formally, a canonical form of a graph G will be a linear ordering of the oriented edges of G which satisfies conditions (i) and (ii). We shall often let I, J be maps from \{1, \ldots, |E|\} to E. In general, I will be a map from \{1, \ldots, |X|\} to X, in which case we will call I an ordering of X. If \lambda \in \text{Sym}(X) then the pair (I, \lambda) determines the set of ordering \{\alpha I | \alpha \in \lambda\}. The algorithms considered will return all equivalent ordering, i.e., pairs (I, \lambda). We consider several canonization problems.

Problem 3.1. String canonical forms for towers of \Gamma_k\text{-actions.}

Input: A colored set X with orderings (I, \lambda) where \lambda is given as a tower of \Gamma_k\text{-actions, } (A, \pi_0, \ldots, \pi_t).

Find:

1. \alpha \in A \text{ such that } \alpha I \text{ is canonical};
2. group B \subseteq A \text{ of elements which preserve coloring.}

Formally, an algorithm \text{CF} satisfies condition (1) if whenever \text{CF} on (I, A, \pi_0, \ldots, \pi_t) returns \alpha I and \text{CF} on (J, A, \pi_0, \ldots, \pi_t) returns \gamma I for \beta \in A then \gamma^{-1} \in B. Thus the canonical form may be a function of (\pi_0, \ldots, \pi_t). The ordering I is said to be \leq \text{ the order } J \text{ if } (X \text{ with ordering } I) \text{ is lexigraphically before } (X \text{ with ordering } J).

Problem 3.2. Graph canonical forms restricted to a tower of \Gamma_k\text{-actions.}

Input: A graph G with ordering I of V(G) and a tower of \Gamma_k\text{-actions } (A, \pi_0, \ldots, \pi_t) \text{ on } V(G).

Find:

1. \alpha \in A \text{ such that } \alpha I \text{ is a canonical ordering of } V(G);
2. group B \subseteq A \text{ of automorphisms of } G \text{ in } A.

We list a third problem which we will not use in this paper but whose polynomial time solution will follow easily from the ideas in this paper and the ideas in [M1ta] and may have application elsewhere.

Problem 3.3. Hypergraph canonical forms restricted to tower of \Gamma_k\text{-actions.}

In problem 3.3 the graph in problem 3.2 is replaced by a hypergraph.

Before presenting the fourth problem of this section we give a polynomial time solution to problem 3.1. This algorithm gives:

Theorem 3.4. String canonical forms for towers of \Gamma_k\text{-actions is polynomial time constructable for fixed } k.

Let (A, \pi_0, \ldots, \pi_t) be a tower of \Gamma_k\text{-actions on a colored set } X.

Further, let I be an ordering of X.
We consider two procedures, \( CF \) which returns the canonical ordering and \( C \) which returns the group of symmetries. If \( S \subseteq X \) we define \( C \) and \( CF \) as follows:

(i) \( C_S(A) = \{a \in A|a \text{ preserves color on } S\} \);
(ii) \( CF_S(I,A) = \text{canonical } aI, a \in A, \text{ for coloring on } S \text{ only}. \)

Space does not permit giving the procedure for \( C \) but it follows easily from that of \( CF \).

The algorithm will have a recursive form similar to [Lu80, BL83]. The procedure has two phases. The first phase is used to reduce the group's action on \( S/\pi_1 \) to a \( \Gamma_k \) group. While the second phase just applies an algorithm similar to the color canonical algorithm of [BL83] to an action which is \( \Gamma_k \). In the procedure \( X, I, \pi_0, \ldots, \pi_t \), and \( t \) will be global variables and \( S \) will be an \( A \)-stable subset of \( X \).

Procedure: \( CF_S(I,A,i) \).

Begin: (1) If \( S = \{x\} \) for some \( x \in X \) then return \( I \);

(2) If \( A \) is not transitive on \( S \) then

   (a) "Canonically" pick an \( A \)-stable partition of \( S \text{ w.r.t. } I \),
   say, \( S_1, S_2 \);

   (b) Return \( CF_S(CF_S(I,A,i), C_S(A,i)) \);

(3) If \( S \leq T \leq X/\pi_1 \) for some \( T \) then

   (a) "Canonically" pick \( S' \subseteq S/\pi_{i+1} \text{ w.r.t. } I \);

   (b) Compute \( A' = A[S'] \) and right coset representatives of
   \( A' \text{ in } A \), say, \( \{a_1, \ldots, a_k\} \);

   (c) Return \( CF_S(CF_S(a_{j,i}, A', i+1)) \);

(4) (a) Find a canonical primitive block system w.r.t. \( I \) on \( S \),
   say, \( \pi > \pi_1 \mid S \);

   (b) Compute \( A' = A(\pi) \) and right coset representatives
   \( \{a_1, \ldots, a_k\} \) of \( A' \text{ in } A \);

   (c) Return \( CF_S(CF_S(a_{j,i}, A', i)) \).

End

Since the algorithm parallels earlier algorithms it follows that
\( CF_S(I,A,0) \) will compute a string canonical form.

To see that \( CF \) runs in polynomial time we must bound the size of \( I \)
whenever step (4) is implemented. We state three facts which follow for
any recursive call of \( CF \), say, \( CF(c,A,i) \) by lemmas 2.7 and 2.8.
(i) $(A \uparrow S, \pi_0 \uparrow S, \ldots, \pi_k \uparrow S)$ is a tower of $\Gamma_k$-actions;
(ii) If $|S/\pi_1| > 1$ then $A \uparrow S/\pi_1$ is in $\Gamma_k$;
(iii) If $|S/\pi_1| = 1$ then $A \uparrow S/\pi_{k+1}$ is a $\Gamma_k$-action.

By a straightforward calculation one can show that the number of recursive calls is bounded by $n^{\lambda+2}$, where $\lambda$ is the constant from theorem 2.4. This completes the proof of theorem 3.4.

To apply the solution to the string canonical forms to the graph canonical forms, we simply consider the group as acting on pairs of vertices. Since edges are pairs of vertices we color these pairs according to whether they are edges or nonedges. But this is the string canonical form problem. We need only show that a tower of $\Gamma_k$-actions on $X$ can be “lifted” to a tower of $\Gamma_k$-actions on $X^2$.

We state this lifting in a slightly stronger form which we will need later. If $\pi$ is a relation on $X$ and $\tau$ is a relation of $Y$ then define $\pi \cdot \tau$ on $X \times Y$ by $(x,y) \pi \cdot \tau (x',y')$ if $x \pi x'$ and $y \tau y'$. Thus, a polynomial time algorithm follows for problem 3.2 from the following lemma.

**Lemma 3.5.** If $(A,X,\pi_1,\ldots,\pi_k)$ and $(B,Y,\pi_1,\ldots,\pi_k)$ are towers of $\Gamma_k$-actions then so is $(A \times B, X \times Y, \pi_1 \cdot \tau_1, \ldots, \pi_k \cdot \tau_k)$.

### 3.2 Canonical Group Intersection

In this section we consider the problem of canonically intersecting two permutation groups. The solution again will parallel the work of [Lu80, BL83]. In practice the two groups may not be acting on the same set. Thus, we consider a slight generalization of the problem.

Let $A$ act on $X$ and $B$ act on $Y$. The amalgamated intersection of $A$ and $B$ or simply the intersection of $A$ and $B$, $A \cap B$, is the group of elements in $\text{Sym}(X \cup Y)$ which stabilizes $X$ and $Y$ and whose action on $X$ is in $A$ and on $Y$ is in $B$.

We shall say a function $\text{CIN}$ is canonical form for $A \cap B$ if it returns ordering satisfying condition (1).

1. $\text{CIN}(I,J,A,B) = (aI,bJ)$ and $\text{CIN}(aI,bJ,A,B) = (a'I,b'J)$ implies $a'a^{-1} \in A$ and $b'b^{-1} \in B$ for $a \in A$ and $b \in B$.

This gives the fourth problem.

**Problem 3.6.** Canonical intersection of an arbitrary group with a tower of $\Gamma_k$-actions.

**Input:** Two orderings $(I,A)$ and $(J,B)$ where $A$ is given as a tower of $\Gamma_k$-actions.
Find: $\alpha \in A$ and $\beta \in B$ such that $(\alpha I, \beta J)$ is canonical.

We do not require solutions to canonical intersections to return a linear ordering of $X \cup Y$. But, there are many natural orderings of $X \cup Y$ determined by $\alpha I$ and $\beta J$. For the applications here we find the two orderings more natural.

The algorithm for problem 3.6 will be a combination of the group intersection algorithm in [Lu80] and the canonization algorithms of [BL83, FSS83].

We say $(I, J) \leq (I_1, J_1)$ if $I$ and $I_1$ first enumerate $X \wedge Y$ and $J^{-1} I$ is lexicographically before $J^{-1} I_1$.

The problem of simply computing the intersection of $A$ and $B$ when $A$ is a tower of $\Gamma_k$-actions is also polynomial computable. The algorithm proposed is a natural combination of the algorithm for color symmetries in a tower of $\Gamma_k$-actions and the group intersection algorithm in [Lu80] for $\Gamma_k$ groups. Due to space constraints we will not present it here. The canonization procedure will call this algorithm.

Let $C$ be a subgroup of $A \times B$ acting on $X \times Y$. Let $\Pr_1(C) \Pr_2(C)$ be the projections of $C$ on $X$ and $Y$, respectively. Consider the following function on $C$ for subsets $Z$ of $X \cap Y$.

$$IN_2(Z) = \{(\alpha, \beta) \in C | \alpha \setminus Z = \beta \setminus Z\}.$$  

Similar to $IN_2$ we extend $CIN$. We have $CIN_2(I, J, C)$ return $(\alpha I, \beta J) \in C$ which is canonical w.r.t. $C_2 = \{(a, b) \in C | a \setminus Z = b \setminus Z\}$. We give a list of recursive properties of $CIN_2$ which easily produces the desired polynomial time algorithm.

(3.7). If $\Pr_1(C)$ is not transitive on $Z$ with stable canonical partition $Z_1, Z_2$, then:

$$CIN_2(I, J, C) = \min\{CIN_2(I_1, J_1, C_1), \Pr_1(C)\}.$$  

(3.8). If $A^\ast \subseteq \Pr_1(C)$ with right coset representatives $\{a_1, \ldots, a_k\}$ in $\Pr_1(C)$ then:

$$CIN_2(I, J, C) = \min\{CIN_2(a_1 I, B_1 J, C^\ast)\}$$

where $(a_1, B_1) \in C$ and $C^\ast = \{(\alpha, \beta) \in C | a \in A^\ast\}$.

(3.9). If $Z = \{z\}$ then:

$$CIN_2(I, J, C) = \min\{(a_1 I, B_1 J)\}.$$
where \((a_i, b_i)\) are right coset representatives of \(C((z, z))\) in \(C\).

Using these three identities about \(CIN\) and the recursive control structure of the color canonicalization algorithm we can compute the intersection and the canonical intersection in \(n^{k+2}\) recursive cells of identity (3.9). The stabilizer of \((z, z)\) in \(C\) can be computed using Sim's algorithm as analyzed in [FHL].

In the case that \(B\) is also a tower of \(\Gamma_k\)-actions then we can return not only the intersection but a tower of \(\Gamma_k\)-actions. The following lemma will suffice.

Lemma 3.10. If \((A_1, X, \tau_1, \ldots, \tau_k)\) and \((B_1, Y, \tau_1, \ldots, \tau_k)\) are towers of \(\Gamma_k\)-actions then \((A \cap B_1, X \cap Y, \tau_1 \cap \tau_1, \ldots, \tau_k \cap \tau_k)\) is a tower of \(\Gamma_k\)-actions.

4. SYMMETRIES OF A VERTEX GIVEN BY A GROUP

In [MI79] we discussed gadgets — graphs which were used to denote symmetries or as a data structure for symmetries. Here we reverse those ideas and replace bridges or gadgets of a graph by a group or coset which will represent the symmetries of the frontier of a bridge. We shall apply these ideas to isomorphism testing and canonical forms for \(k\)-contractable graphs. Here, we present an algorithm which under certain conditions tests isomorphism of graphs where the vertices have specified symmetries. We make these notations precise in what follows. It seems crucial that the graphs considered have multiple edges. Throughout this section the graphs are assumed to have multiple edges.

Definition 4.1. A graph with specified symmetry is a graph \(G = (V, E)\), with a set of ordering \((I_V, A_V)\) for each set \(E(v)\), \(v \in V\), and a consistent partition \(P\) of the vertices \(G\). The partition \(P\) is consistent if \(vPw\) implies that \(I_w^{-1}A_vI_v = I_v^{-1}A_vI_v\).

We next consider restriction on the way the group at each vertex can act.

Definition 4.2. A graph \(G\) has its vertex symmetries given by towers of \(\Gamma_k\)-actions if \(G\) has specified symmetry where \(A_V\) is given in the form \((A_V, \tau_1, \ldots, \tau_k)\), a tower of \(\Gamma_k\)-actions, and the partition \(P\) is consistent with this structure. We shall say the symmetries are \(\Gamma_k\) on the multiple edges if for every vertex \(v \in V(G)\) the induced action of \(A_v\) on \(E(v)/KE\) is a \(\Gamma_k\)-action. We shall often refer to these graphs as simply \(\Gamma_k\)-graphs. An isomorphism of \(G\) must preserve this structure.

Theorem 4.3. Isomorphism and canonical form for \(\Gamma_k\) graphs are polynomial
time constructable for fixed \( k \).

The assumption that the symmetries are \( \Gamma_k \)-actions on the multiple edge relation will insure that the induced action on vertices will be tractable. It would be interesting to know if this constraint can be dropped. By identifying multiple edges the symmetries become \( \Gamma_k \)-actions. In this case, the edge stabilizer of a connected graph is a group in \( \Gamma_k \).

We state this well-known fact as a lemma.

Lemma 4.4. If \( G \) is connected with specified symmetries which are \( \Gamma_k \)-actions then the automorphisms of \( G \) which stabilize an edge form a group in \( \Gamma_k \).

We explain the canonical form algorithm here. It will use the leveling idea. By standard techniques we may assume the graph \( G \) is connected. For each edge \( e \) in \( E(G) \) we compute the canonical form from \( e \). We shall level the edges and vertices by their distance from \( e \). Using this leveling we shall inductively construct the partial automorphisms from edges to edges and from vertices to vertices. We begin the formal construction.

Label the edge \( e' \) of \( G \) with the integer which is the distance (the number of vertices in a shortest path) from \( e \) to the edge \( e' \), e.g. \( e \) is labeled 0. Vertices of \( G \) are labeled with the integer which is the number of edges they are from \( e \), the end points of \( e \) are labeled 1. An edge is even if both end points are labeled the same, otherwise it is odd.

Let \( G_1 \) be the induced graph on edges labeled \( \leq i \). That is, \( G_1 \) is the graph on vertices labeled \( \leq i + 1 \) and edges labeled \( \leq i \). The graph \( \bar{G}_1 \) consists of: (1) \( G_1 \); (2) all odd edges labeled \( i + 1 \) where the endpoint labeled \( i + 2 \) has been replaced with a new distinct vertex for each edge; (3) two copies of each even edge labeled \( i + 1 \) where one copy is attached to one end point labeled \( i + 1 \) and the other copy is attached to the other end point labeled \( i + 1 \). Again, the other endpoint of these even edges is a new vertex. The vertex symmetries for vertices labeled \( \leq i \) of \( G_1 \) will be those of \( G \) while the vertices labeled \( i + 1 \) will have no restriction on symmetries. The vertex symmetries of \( \bar{G}_1 \) for vertices labeled \( \leq i + 1 \) will be those of \( G \). Again, the symmetries of vertices labeled \( i + 2 \) will not be constrained.

Let \( \text{Auto}(G_1) \) be the automorphisms of \( G_1 \) which fix \( e \). Similarly, \( \text{Auto}(\bar{G}_1) \) are the automorphisms of \( \bar{G}_1 \) which fix \( e \).

We shall need inductively two conditions or facts concerning the isomorphism. First, that the automorphisms of \( G_1 \) and \( \bar{G}_1 \) which fix \( e \)
acting on the edges are written as a tower of $\Gamma_k$-actions. Second, the
automorphism of $G_i$ leaving e fixed acting on the vertices is in $\Gamma_k$.
We consider the second condition first. If we identify multiple edges
of $\tilde{G}_{i-1}$ but not multiple copies of the vertices and the symmetries are
those induced from the vertices of $\tilde{G}_{i-1}'$, then the new graph satisfies
the hypothesis of lemma 7. Therefore, the automorphism of the graph
fixed the edge e are in $\Gamma_k$. Now, any automorphism of $\tilde{G}_{i-1}$ induces an
automorphism on this graph. Since the action on the vertex is unchanged
by identifying multiple edges we have the second condition for $\tilde{G}_{i-1}$.
If we now also identify the multiple vertices of $\tilde{G}_{i-1}$, the automorphism
fixing e will still be in $\Gamma_k$. But this is the same graph we obtain by
identifying the multiple edges of $G_i$. This proves the second condition.
We shall maintain the first condition throughout the construct.

We need only give a polynomial time algorithm for constructing
$\text{Auto}(G_{i+1})$ from $\text{Auto}(G_i)$ and constructing $\text{Auto}(G_{i+1})$ from $\text{Auto}(G_i)$ where
the groups are given as towers of $\Gamma_k$-actions. We consider the latter
case first.

The elements of $\text{Auto}(G_{i+1})$ are simply those elements of $\text{Auto}(G_i)$
which preserve multiple copies of vertices and edges. That is, they
preserve the relation $a \equiv b$, $a$ and $b$ are copies of the same edge or
vertex. We obtain the canonical form for this relation by applying the
graph canonical form restricted to a tower of $\Gamma_k$-actions algorithm (3.2).
Here the graph has vertices consisting of the vertices and edges of $G_i$
and the edges are the multiple copies relation. Since towers of $\Gamma_k$-actions
are closed under taking subgroups the first condition is inductively
satisfied for $\text{Auto}(G_{i+1})$.

We have left the construction of $\text{Auto}(G_{i+1})$ from $\text{Auto}(G_i)$ which we
must show is a tower of $\Gamma_k$-actions. Let $(A, \pi_1, \ldots, \pi_i) = \text{Auto}(G_i)$, a
tower of $\Gamma_k$-actions. Let $E_i$ be the edges of $G_i$ which are common to a
vertex labeled $i$. The maps on $E_i$ which preserve the symmetry of vertices
labeled $i$ can be written as a direct product of wreath products since
the symmetries are consistent. That is, it will be of form $\Pi A \wr S$. 
Let $B$ be this group. The group $\text{Auto}(G_i)$ will be the amalgamated inter-
section of $A$ and $B$. By (3.6) this intersection and its canonical form
are constructable in polynomial time. To show that the intersection can
be written as a tower of $\Gamma_k$-actions we note the following. Let $\pi_{i,j}$ be
the $i$th equivalence relation of vertex $j$. We claim that $\pi_{i,j} = \overline{s_i} \pi_{i,j}$
where $\overline{s_i}$ has label $s_i + 1$, form a tower for $A \cap B$. By induction,
assume it is true for $A$ where the intersection is taken over vertices
labeled \( i \). In the construction of \( B \) we used the symmetric group in
the wreath product. But we know that the induced action on the verti-
ces labeled \( i + 1 \) given by \( A \) is in \( \Gamma_k \). So we may restrict the wreath
product to this \( \Gamma_k \) group. Let \( B \) be this smaller wreath product. Now,
\((B', \ldots, \tau_1', \ldots, \tau_k')\) is a tower of \( \Gamma_k \)-actions where \( \tau_1' = \frac{c_{i,j}}{j^*} \); \( V_j \) has label
\( i + 1 \). This proves the theorem.

We shall need that the full group of automorphisms of \( G \) can be
written as a tower of \( \Gamma_k \)-actions.

Let \( G \) be a graph whose vertex symmetries are given by towers of
\( \Gamma_k \)-actions and these symmetries are \( \Gamma_k \) on the multiple edges. Suppose
the vertex symmetries of \( G \) are \((A_i, \tau_{1i}, \ldots, \tau_{ki})\) for vertex \( V_i \). Let \( ME \)
be the multiple edge relation on \( G \). In this case the automorphisms of
\( G \) are a tower of \( \Gamma_k \)-actions.

**Lemma 4.5.** If \( G \) and \((A_i, \tau_{1i}, \ldots, \tau_{ti})\), are as above and \( A \) is the group
of automorphisms of \( G \) acting on the edges, then \((A, ME, \tau_1, \ldots, \tau_k)\), where
\( \tau_i = \frac{G_{\tau_{ij}}}{j} \) and \( ME \) for \( V_i \not= V(G) \), is a tower of \( \Gamma_k \)-actions.

5. \( k \)-CONTRACTABLE GRAPHS

In this section we simultaneously define the valence \( k \)-contractable
graphs and present a polynomial time algorithm for testing isomorphisms
of these graphs. We define the valence \( k \)-contractable graphs via a
decomposition algorithm. The definition is unsatisfactory since small
perturbations may result in a different class of graphs. We leave it
as an open problem to find a satisfactory definition. The definition
is sufficient enough so that any perturbation will still contain the
graphs of bounded genus and bounded valence. Let \( G \) be a graph with pre-
scribed symmetries either given by towers of \( \Gamma_k \)-actions or unconstrained.
We shall assume that two unconstrained vertices have at most one edge
between them. We shall call these graphs \( \Gamma_k \)-contractable graphs.

We shall say a vertex \( v \) is \( \Gamma_k \)-contractible if the symmetries of \( v \) induce a \( \Gamma_k \)-
action on the multiple edge relation \( ME \); otherwise it is not \( \Gamma_k \)-contractible.
A \( \Gamma_k \)-bridge is a \( \Gamma_k \)-bridge if it is a bridge of \((G,Y)\) where \( Y \) is the set of verti-
ces which are not \( \Gamma_k \) in \( G \). A \( \Gamma_k \)-bridge is not formally a \( \Gamma_k \)-graph for
two reasons. First, the unconstrained vertices have no associated
tower of relations. This problem is remedied by allowing these uncon-
strained vertices to "inherit" the symmetries from their neighbors.
Note that each neighbor \( v' \) of an unconstrained vertex \( v \) either shares
only a single edge with \( v' \) or else its symmetries are a tower of \( \Gamma_k \)-
actions. Thus, the symmetries of an unconstrained vertex can be re-
stricted to a product of wreath products where the wreath products are
of the form \( AW_{\pi_k} \), where \( \lambda \) is the inherited symmetry from a neighboring
vertex and \( k' \leq k \). Second, we have not specified the symmetry of fron-
tier vertices. We transform \( Br \) into a modified bridge so that it has
the form of a \( \Gamma_k \)-graph. For each vertex \( v \) of \( Br \) common to a frontier
vertex \( v' \) of \( Br \), we introduce a new copy of the frontier vertex \( \bar{v} \)
and have edges between \( v \) and \( v' \) go between \( v \) and \( \bar{v} \). We now view these new
frontier vertices as unconstrained vertices and allow them to inherit
the symmetries of their neighbor. Let \( \bar{Br} \) be the bridge obtained from
\( Br \) by the above construction.

Given any \( \Gamma_k \)-bridge \( Br \), we can compute the automorphisms and canoni-
cal form by first constructing \( \bar{Br} \) and applying the isomorphism test for
\( \Gamma_k \)-graphs. This procedure will return with a group in the form of a
tower of \( \Gamma_k \)-actions. Applying (3.2) we can compute the subgroup and
canonical form of \( \text{Auto}(Br) \) which sends multiple copies of vertices to
corresponding multiple copies of vertices. This will be \( \text{Auto}(Br) \) and
its canonical form.

Lemma 5.1. The automorphism and canonical form for a \( \Gamma_k \)-bridge is poly-
nomial time constructable and the automorphisms can be written as a
tower of \( \Gamma_k \)-actions.

This gives a natural decomposition of \( \Gamma_k \)-contractable graphs, say
\( G \). For each \( \Gamma_k \)-bridge \( Br \) of \( G \) we identify the internal vertices of \( Br \)
and remove selfloops. We denote this graph by \( \text{Contract}(G) \).

The vertex symmetries of this new vertex will be the induced action
of \( \text{Auto}(Br) \) on the frontier edges. This will be a tower of \( \Gamma_k \)-actions.
We call this graph \( \text{Contract}_k(G) \) which is a \( \Gamma_k \)-contractable graph. By
standard argument \( \text{Contract}_k(G) \) is a canonical operation on \( G \).

If we apply \( \text{Contract}_k \) to a tree of valence \( > k + 1 \) then the procedure
will simply return the original graph. Graphs that are sent to them-
selves under a procedure will be called fixed points. We introduce a
decomposition procedure analogous to the tree isomorphism algorithm.
The procedure \( \text{Remove Leaves} \) will remove leaves and let their neighbors
"inherit" their symmetries. The details will appear in the final paper.

We introduce a third reduction which corresponds to a generalized
3-connected decomposition.

Two vertices of valence 2, ignoring multiple edges, are multiple
if their neighboring vertices are the same. This gives a natural equi-
valence relation on valence 2 vertices. An equivalence class will simply be called a set of multiple vertices. Let \( v_1, \ldots, v_k \) be a set of multiple vertices of \( G \). Multiple vertices can be identified with the new vertex "inheriting" the symmetries of the multiple vertices and their common neighbors. The third reduction is called Remove Multiple Vertices. Again, the details will appear in the final paper.

The graph \( G \) is a fixed point of these three reductions if it is a fixed point of each reduction. Note that we may arrive at a different fixed point depending on the order which we apply these reductions. For specificity's sake, suppose we consider fixed points of the following procedure.

**Procedure:** \( \text{Reduction}_k(G) \).

1. If \( G = \text{Remove Leaves}(G) \) then
   \( G = \text{Remove Leaves}(G) \)
2. Else if \( G \neq \text{Remove Multiple Vertices}(G) \) then
   \( G = \text{Remove Multiple Vertices}(G) \)
3. Else \( G = \text{Contract}_k(G) \).

**Definition 5.2.** \( G \) is a \( k \)-contractable graph if successive application of reduction applied to \( G \) yields a singleton.

From the discussion above we get

**Theorem 5.3.** Isomorphism for \( k \)-contractable graphs is polynomial time testable for fixed \( k \).

6. **THE BOUNDED GENUS CASE**

Here we shall show that graphs of bounded genus are \( k \)-contractable graphs. Thus, demonstrating that the \( k \)-contractable graphs form a class of graphs which is a common generalization of the bounded valence and the bounded genus graphs. The containment will follow by showing that the fixed graphs under the reduction operation have the property that their genus grows linearly in \( k \). Throughout the rest of the discussion let \( G \) be a fixed point. Since neither the genus nor the fact that \( G \) is a fixed point is effected by multiple edges, we may assume without loss of generality that \( G \) has no multiple edges. Similarly, we may assume that \( G \) has no vertices of valence 2. Not all vertices of \( G \) will have valence \( \geq k + 1 \). Let \( S \) be the set of vertices of \( G \) with valence \( \leq k \). Then, \( S \) will be an independent set. Let \( v' \) denote the number of vertices of \( G \) in \( V - S \). We first state a relationship between
genus and $k$-contractable graphs as defined in the previous sections. We shall actually prove a slightly stronger result for a stronger notion of $k$-contractable.

Theorem 6.1. If $G$ is a fixed point of Reduction$_{k-1}$ then the genus of $G,g$ satisfies

$$2g \geq \frac{(k/2 - 6)/6}{v'} + 2.$$ 

To see that such a large value of $k$ is necessary consider the infinite tiling of the plane in diagram 1. Since this tiling is periodic we can construct arbitrarily large graphs of genus 1 which are fixed points of Reduction$_{11}$.

Let $v'$ denote the vertices $V-S$. We distinguish two types of edges of $G$. The edges between points of $v'$ will be called type 1 edges and those between $v'$ and $S$ will be called type 2. Since we can easily distinguish these two types we can include in Reduction a procedure which restricts the symmetries at each vertex to the coset which preserves type. Call this new reduction procedure Reduction$_{1}$. For Reduction$_{6}$ the example from diagram 1 will contract to a point. For Reduction$_{1}$ we get the following result.

Theorem 6.2. If $G$ is a fixed point of Reduction$_{k}$ then the genus of $G,g$ satisfies

$$2g \geq \frac{(k - 6)/6}{v'} + 2.$$ 

Note that we are using Reduction$_{k}$ since fixing one edge at a vertex will only in general effect one of the two types of edges at that point. This gives the following corollary for $k$-contractable graphs with respect to Reduction$_{1}$.

Corollary 6.3. If $k > 4g + 2$ and $g \geq 1$ then the $k$-contractable graphs include the graphs of genus $g$. For $g = 0$ (the planar case) $k = 5$ will suffice.

To see the corollary we simply note that $v' \geq 3$ since a fixed point can have no multiple valence 2 vertices.

Proof of Theorem 6.2. The proof uses standard counting argument based on Euler's formula $2g = 3 - f - v + 2$ where $g$ is the genus of some embedding and $e$, $f$, and $v$ are the numbers of edges, faces, and vertices, respectively.

Let $G$ be a graph with some fixed embedding. Further assume $G$ is
fixed by Reduction. Without loss of generality we may add new vertices of type S and new edges of type A as long as the genus is unchanged. If any vertex in V' has two consecutive edges of type A we may add to this face a valence 3 vertex of type S. On the other hand, if a vertex v ∈ V' has two consecutive type B edges, then we may add a type A edge across this face common to v. Here we use the fact that S is an independent set. Without loss of generality we may assume that the edges at each vertex of V' are alternately of type A and then type B. So, if a equals the number of type A edges and b equals the number of type B edges then 2a = b. Using this fact one can show:

**Lemma 6.4.** \( f \leq a + b/2 = 3a. \)

Using the following facts: (1) \(|V'| + |S| = V,\) (2) \(|S| \leq b/3,\) (3) \(e = b + a,\) (4) \(b \geq k|V'|,\) we get theorem 6.2.

**ACKNOWLEDGEMENTS**

I would like to thank Tom Leighton for most helpful discussions especially in the last section. I would also like to thank Laslo Babai, Eugene Luks, and Jeff Kahn. This work was supported by the National Science Foundation.
Bibliography


[FMR 79] I.S. Filotti, G.L. Miller and J. Reif, "On Determining the Genus of a Graph in O(V^2) Steps".


[Mi ta1] G.L. Miller, "Isomorphism of Graphs which are Pairwise k-Separable", submitted to Information and Control.

[Mi ta2] G.L. Miller, "Isomorphism of k-Contractable Graphs (a generalization of bounded valence and bounded genus)", submitted to Information and Control.