# TESTING ISOMORPHISM OF CONE GRAPHS*

(Extended Abstract)

Christoph M. Hoffmann

Dept. of Computer Science
Purdue University
West Lafayette, Indiana 47907

presently

Gastprofessor
Vertretung des Lehrstuhls für Informatik
Christian-Albrechts-Universität
23 Kiel 1
West Germany

Summary

We give algorithms to decide graph isomorphism in a subclass of graphs which we call cone graphs. A cone graph is an undirected graph for which there exists a vertex r which uniquely determines a breadth-first search (BFS) tree. Equivalently, all shortest paths from r to any other graph vertex are unique.

Our algorithms may be used either nondeterministically or probabilistically. Used as probabilistic algorithms, they return always a correct answer, but with an expected running time only.

## 1. Introduction

It is not known whether graph isomorphism can be decided in nondeterministic subexponential time. A number of classes of graphs are known for which there exist polynomial time or subexponential time algorithms. These include Steiner graphs [6], planar graphs [2,3], interval graphs [5], cubic edge transitive graphs [4], graphs of bounded genus [9], and graphs of bounded color multiplicity [1].

We give algorithms for a new class of graphs which we call cone graphs. These are undirected, connected graphs in which there exists a vertex r such that a breadth-first search from r uniquely partitions the graph edges into BFS tree edges and nontree edges. Cone graphs do not fall into any of the other graph classes mentioned above.

The intuition behind this definition is that any automorphism which leaves r fixed must preserve the distance from r in the BFS tree, thus must respect the edge partition

*This work was supported in part by NSF grant MCS 78-01812, and a travel grant from the Deutsche Forschungsgemeinschaft.

into tree and nontree edges. We thus can determine all automorphisms by determining the automorphisms which leave r fixed, for each of the possible images of r. Note that the group of automorphisms which leave r fixed is a subgroup of the automorphism group of the BFS tree.

We use a group theoretic approach which generalizes a technique discovered by Babai [1], and overcomes one of its limitations. His method counts automorphisms which permute vertices only within blocks of a fixed vertex partition. The blocks must be of constant size. We overcome this restriction by a recursion which takes advantage of the structure of the BFS tree. If h denotes the height of the BFS tree, k the largest number of sons of any interior vertex, then we obtain an $O((k!\cdot n)^h)$ algorithm. Note that h is O(log n) if the BFS tree satisfies minimal regularity conditions.

Section 3 summarizes the counting technique. We develop the nondeterministic version of our algorithms in Section 4. In Section 5 we prove bounds on the running time of the probabilistic version.

Certain cone graphs have much additional structure leading to stronger properties. It seems as if these properties should admit faster algorithms. We discuss this possibility briefly in Section 6.

## 2. Definitions and Terminology

We consider undirected graphs G = (V,E), where V denotes the set of <u>vertices</u> of G, E the <u>edges</u>, i.e. unordered pairs (v,w) of vertices. Two graphs G = (V,E) and G'=(V', E') are <u>isomorphic</u> if there exists a bijective mapping $p: V \leftrightarrow V'$ such that (p(v),p(w)) is an edge of G' whenever (v,w) is an edge of G. An <u>automorphism</u> of G is a vertex permutation p such that (p(v),p(w)) is an edge of G whenever (v,w) is. The automorphisms of a graph form a group composition. A <u>group</u> $A = (S,*)$ is a set S closed under an associative binary operation $*$, such that in the equation $x*y = z$ any two elements of S determine the third uniquely (in S).

We study graph automorphisms because of their close relationship to the graph isomorphism problem. We will see that the algorithms developed here for determining the automorphism group of a graph, with minor modifications, can also decide presence or absence of an isomorphism between two graphs.

We use elementary facts from group theory, such as can be understood from the first few chapters of Hall's book [8].

The <u>order</u> of a group $A = (S,*)$ is the number of elements in S, and is written $|A|$. $B = (S',*)$ is a <u>subgroup</u> of A, $B < A$, if B is a group under $*$, and S' is a subset of S. Given a <u>subgroup</u> B of A, the <u>quotient</u> A:B is the set of equivalence classes of elements in A, with x equivalent y iff $x^{-1}*y$ is an element of B. It is elementary from group theory that all equivalence classes are of equal size, and that B is one of them. For x not element of B, the set $x*B$ is a <u>left</u> <u>coset</u> of A:B, and is the equivalence class of x. We make use of the following

<u>Theorem</u> (Lagrange) If B is a subgroup of A, then the order of B divides the order of A.

We solve the isomorphism problem for a specific class of graphs which we call <u>cone graphs</u>.

Definition A graph $G = (V,E)$ is a cone if there exists a vertex $r$ in $V$ such that a breadth-first search (BFS), beginning with $r$, results in a unique and balanced BFS tree.

Once we have fixed $r$, we can assign to each vertex of $G$ a rank $i$, with leaves assigned rank 0, and assigning to fathers of vertices of rank $i$ the rank $i+1$.

Definition A graph $G$ is a regular cone if it is a cone and all interior vertices of the BFS tree have the same number of sons. It is a semiregular cone if it is a cone and all interior vertices of the BFS tree of the same rank have the same number of sons.

It is decidable in polynomial time whether a graph is a cone, a regular cone, or a semiregular cone. Note that we do not, with our definitions, constrain the subgraph consisting of nontree edges in any way.

Observe that it suffices to solve the isomorphism problem for cone graphs subject to mapping roots into each other, for we can separately solve it for each of the root choices.

## 3. Babai's Algorithm

In [1], Babai gave a $\Delta^R$ algorithm for counting the number of automorphisms of a graph $G = (V,E)$ subject to the following constraint. Partition the set of vertices $V$ of $G$ into blocks $B_1, B_2, .., B_p$, such that each block consists of at most $k$ vertices, where $k$ is a constant independent of the size $n$ of $V$. The automorphisms counted are precisely those which permute vertices within each partition only, that is, a vertex $v$ in $B_i$ can be mapped only into other vertices in $B_i$.

Let $A$ be the group of all such automorphisms of $G$. The key idea for determining $A$ is this. Beginning with a group $H_0$ whose structure is known a priori, construct a tower of subgroups

$$H_m < H_{m-1} \quad \cdots \quad H_1 < H_0$$

such that $A = H_j$ for a known value $j$, $H_m$ consists of the identity alone, and $m$ is bounded by a polynomial in $n$ (the number of vertices of the graph). Membership in $H_i$ must be testable in polynomial time, and the size of the quotients $H_i:H_{i+1}$ must also be uniformly bounded by a polynomial in $n$.

This can be done under the restrictions on the block size and the automorphisms. The algorithm guesses all (left) coset representatives for each of the $H_i:H_{i+1}$. We verify that $x$ and $y$, guessed as distinct coset representatives for $H_i:H_{i+1}$, indeed represent distinct cosets by verifying that $x$ and $y$ are in $H_i$ but not in $H_{i+1}$, and that $x^{-1}y$ is not in $H_{i+1}$, i.e. $x$ and $y$ are not equivalent.

Since the order of $H_0$ is known a priori, it follows from Lagrange's Theorem that we have found all coset representatives if and only if the sizes for each of the quotients, i.e. the number of distinct coset representatives for $H_i:H_{i+1}$, plus 1, multiply to the size of $H_0$. The size of $A$ will be the product of the sizes of $H_i:H_{i+1}$, $i \geq j$, since $H_m$ consists of the identity only.

Note that the group $A$ is generated by the coset representatives for the factors $H_i:H_{i+1}$, $i \geq j$. If we have a device for generating elements in $H_0$ with uniform distribution, then the (this far nondeterministic) algorithm can be made a probabilistic algorithm, with the coset representatives guessed randomly. We give the necessary details of this in Section 5.

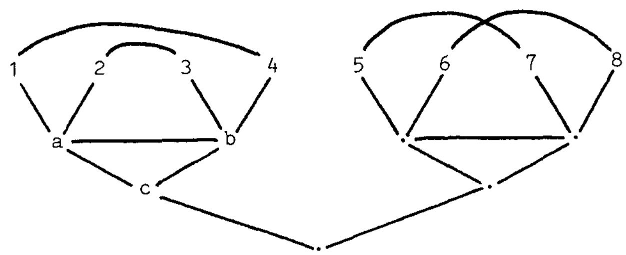
## 4. A Nondeterministic Algorithm

For the sake of a simple presentation, we develop our algorithms for regular cone graphs of degree 2, i.e. graphs for which there is a vertex r such that a BFS from r results in a unique full binary tree which is balanced.

Consider such a cone of height h. Inductively we determine a tower of height h of automorphism subgroups. Let A be the group of all automorphisms of G which leave the root r fixed. The group $A_0(G)$ consist of the identity automorphism only. For $1 \leq i \leq h$, the group $A_i(G)$ consists of all those automorphisms of G which permute only nodes within subtrees of the BFS tree of G which are of height i. Observe that

$$I = A_0(G) < A_1(G) < \dots < A_h(G) = A$$

<u>Example</u> Consider the graph below




The following four leaf permutations (completed appropriately) make up $A_1(G)$: $\langle(12)(34)\rangle$, $\langle(12)(34)(56)(78)\rangle$, $\langle(56)(78)\rangle$, $\langle\rangle$. The permutation $\langle(13)(24)\rangle$ can also be completed to an automorphism, but it requires permuting nodes a and b which are in the subtree rooted in c. This subtree is of height 2. Thus this automorphism is in $A_2(G)$, but not in $A_1(G)$.

We determine A by determining the number of cosets $a_i$ of $A_i(G):A_{i-1}(G)$, $1 \leq i \leq h$. The inductive basis of our recursive algorithm is determining $a_1$. This we do using the method of Babai observing that the automorphisms in $A_1(G)$ respect a vertex partition in which two leaves which have the same (rank 1) father are in the same block. Thus, each block contains either just two leaves or one interior vertex of G.

Let $U_1$ be the group of all permutations which permute only nodes in subtrees of height 1 (i.e. leaves only), without necessarily respecting nontree edges. Clearly $U_1 > A_1(G)$, and $|U_1| = 2^m$, where $m = (n+1)/4$ is the number of vertices of rank 1 in G. Note that we can represent elements of $U_1$ as a 0/1 assignment to the rank 1 vertices, 1 meaning that the two sons are to be exchanged, 0 that they stay fixed.

Define subgraphs $W_{ij}(G)$, $1 \leq i,j \leq m$, such that $W_{ij}(G)$ consists of the vertices of G plus all those nontree edges (v,w) where v is son of the i-th rank 1 vertex, w son of the j-th rank 1 vertex. Construct a sequence of graphs $X_i(G)$, $0 \leq i \leq q = \binom{m}{2}$, where $X_0(G)$ consists of the BFS tree of G only, $X_1(G) = X_0(G) \cup W_{11}(G)$, $X_2(G) = X_1(G) \cup W_{12}(G)$, ..., $X_m(G) = X_{m-1}(G) \cup W_{1m}(G)$, $X_{m+1}(G) = X_m(G) \cup W_{22}(G)$, ..., $X_q(G) = X_{q-1}(G) \cup W_{mm}(G)$.

Let $H_i(G) = A_1(X_i(G))$ be the 1-automorphism group of $X_i(G)$. Observe that $H_i > H_{i+1}$, and that $H_q = A_1(G)$. Define groups $H_{q+1}, \dots, H_{q+m}$ by

$$H_{q+i} = \{x \text{ in } H_q \mid x \text{ does not permute the leaves } 1 \dots 2i\}.$$

Then $H_{q+m} = I$, i.e. consists of the identity alone. We determine the tower

$$I = H_{q+m} < \dots < H_q = A_1 < \dots < H_1 < H_0 = U_1$$

From it we obtain $a_1 = A_1(G)$ , and a set of generators for $A_1$.

<u>Lemma 4.1</u> $|H_i : H_{i+1}| \leq 4$.

The lemma is proved by analyzing the structure of coset representatives. Let $X_{i+1} = X_i \cup W_{jp}$. If b and b' are in different cosets of $H_i:H_{i+1}$, they cannot permute the leaves numbered 2j, 2j+1, 2p, 2p+1 in the same way. The argument for $i \geq q$ is similar.

Theorem 4.2 $A_1(G)$ can be determined in $O(m*e)$ nondeterministic steps, where e is the number of edges of G, m the number of vertices of rank 1.

The algorithm guesses a table of coset representatives for the quotients in the tower $H_{q+m} \cdots H_0$. Because $|H_0| = 2^m$, and by Lemma 4.1, we must guess O(m) coset representatives in total. Testing that x represents a coset of $H_i:H_{i+1}$ means testing x in $H_i$, x not in $H_{i+1}$ and costs O(e) steps for applying x and verifying which edges are preserved. Similarly we test whether x and y represent different cosets.

Definition Graphs G and G' are 1-isomorphic if G' can be mapped onto G by a permutation in $U_1$, and vice versa.

Note that G and G' must have BFS trees of equal height. Let G+G' denote the cone obtained by making G and G' the subtrees of the root of G+G'. Denote with $B_1(G+G')$ those automorphisms of G+G' which possibly exchange G and G', but otherwise permute only nodes within subtrees of height 1. We construct a tower of subgroups to determine $B_1$ by an obvious variant of the above algorithm, by introducing the $W_{ij}(G)$ and $W_{ij}(G')$ edges simultaneously. This effectively doubles the number of vertices in each partition block. Complete the resulting tower by requiring that the leaves 1..2i and 2m+1..2m+2i remain fixed, except, possibly, being exchanged as blocks. Finally, the G and G' components are to remain fixed, thus arriving at a group consisting of the identity alone. The height of this tower is now q+m+1, $q=\binom{m}{2}$, and the group $H_0$ has the order $2^{2m+1}$. We can bound $|H_i:H_{i+1}|$ by 32, since we constrain at most four 0/1 assignments plus the ability to exchange G and G'.

Having constructed the tower for $B_1$, observe that G and G' are 1-isomorphic iff at least one of the generators of $B_1(G+G')$ exchanges the G and G' components. Thus we obtain

Corollary 4.3 1-isomorphism can be tested in $O(n*e)$ nondeterministic steps.

For determining $a_2$, $a_3$, ... and the groups $A_2$, $A_3$, ... we have to generalize the notion of 1-isomorphism. Define $U_i$ in exact analogy to $U_1$ as those permutations which are automorphisms of the BFS tree but do not permute nodes of rank i or higher.

Definition G and G' are k-isomorphic if G' can be mapped onto G by a permutation in $U_k$, and vice versa.

If G is a cone of height h, then h-isomorphism means isomorphism subject to mapping the root of G into the root of G'. We wish to determine k-isomorphism from (k-1)-isomorphism. Crucial for this is the following result.

Theorem 4.4 $A_{k+1}:A_k \cong A_{k+1}U_k:U_k$

It is clear that $A_{k+1}U_k$ is a group. Write $A_{k+1}:A_k$ in terms of its left cosets

$$A_{k+1} = A_k + b_1A_k + \cdots + b_rA_k$$

Clearly $b_j$ must permute some vertices of rank k, but no vertex of rank greater k. Thus $b_j$ is in $U_{k+1}$ but not in $U_k$. We show that the $b_j$ are all distinct coset representatives of $A_{k+1}U_k:U_k$. Assume then that $b_j^{-1}b_i$ is in $U_k$. This means that $b_j$ and $b_i$ must permute the rank k vertices of G in the same way. Since $b_j^{-1}b_i$ is an automorphism, it must also be in $A_k$, contrary to assumption. By a similar argument we can show that the representatives for distinct cosets of $A_{k+1}U_k:U_k$ also represent distinct cosets of $A_{k+1}:A_k$.

Note that we can factor the coset representatives $b_j$ into a permutation of the subtrees rooted in the rank k vertices (in toto), followed by a permutation of vertices **within** these subtrees:

$$b_j = c_j c_j'$$

i.e. $c_j \in U_{k+1}$ and $c_j' \in U_k$. We can now interpret the $c_j$ geometrically. Consider an assignment of 0 or 1 to the rank k+1 vertices of G, 1 specifying that the two subtrees rooted in the sons of the rank k+1 vertex are to be exchanged, 0 that they remain unexchanged. Such an assignment x is one of the $c_j$ permutations iff, after applying the permutation to G, the resulting graph G' is k-isomorphic to G. We exploit this in

Theorem 4.5 If k-isomorphism of G (regular cone of degree 2) can be tested in T(k,n) nondeterministic steps, where n is the number of vertices of G, then (k+1)-isomorphism can be tested in $c*m*T(k,n)$ nondeterministic steps, where m is the number of vertices of rank k+1, and c is a constant independent of k and n.

Observe that $U_{k+1}:U_k$ is a group of order $2^m$ which contains $A_{k+1}U_k:U_k$ as subgroup. We construct a tower

$$I = H_{q+m} < \dots < H_q = A_{k+1}U_k:U_k < H_{q-1} < \dots < H_0 = U_{k+1}:U_k$$

In analogy to the graphs $X_j$ and $W_{ip}$ from above, we define

$$W_{ip} = \{(v,w) \mid (v,w) \text{ a nontree edge, v a vertex in the subtree rooted in the i-th rank k+1 vertex, w vertex in the subtree rooted in the p-th rank k+1 vertex}\}.$$

Define the graphs $X_j$ now as before, using the new definition for $W_{ip}$. Let $H_i$ be the group $A_{k+1}(X_i)U_k:U_k$, $i \le q$. For i=q+j we require that the assignment to the first j vertices of rank k+1 is identically 0.

It is clear that Lemma 4.1 holds for this new tower of subgroups, and that we can use the approach of guessing O(m) distinct coset representatives using the procedure for k-isomorphism to verify that distinct cosets are represented. This establishes the theorem.

Corollary 4.6 (k+1)-isomorphism can be tested in $c*m*T(k,n)$ nondeterministic steps, where m, n and T(k,n) are as above, and c is a constant independent of n and k.

We test (k+1)-isomorphism by the obvious variant of the procedure for determining the (k+1)-automorphisms (modulo k-automorphism). Cf. Corollary 4.3.

Corollary 4.7 If G and G' are two regular cone graphs of degree 2 with n vertices, then isomorphism of G and G' can be decided in $O(n^{\log(n)+c})$ steps, c some constant.

We have to test log(n)-isomorphism for at most n choices of r for G' separately. The timing of the log(n)-isomorphism procedure follows from Corollary 4.3 and Corollary 4.6 by a simple induction.

We now explain how these results generalize to regular cones of higher degree, semi-regular cones and cones in general. For regular cones of degree d, i.e. cones with a BFS tree in which every interior vertex has d sons, the order of $U_{k+1}:U_k$ is $(d!)^m$, since we can permute the d descending subtrees of any rank k+1 vertex in d! ways. We must guess at most $(d!)^2$ distinct coset representatives for each quotient in the subgroup tower, generalizing Lemma 4.1. The bound of Corollary 4.3 becomes now $O(m*(d!)^2*e)$. Since Theorem 4.4 is true for the general case, we use the same techniques for determining the higher order automorphisms. This results in a procedure requiring $O((n*d!)^{2*h+c})$ nondeterministic steps, where h is the height of G which is the logarithm of n base d, i.e. O(log(n)).

For semiregular cones we observe that $A_{k+1}(G) = A_k(G)$ if all rank k+1 vertices have exactly one son. Thus, such plies can be skipped in the recursion, and the recursive depth is still O(log n), as in the case of regular cones. For general cones we can bound the recursion depth only with $O(\sqrt{n})$, arriving at the less attractive running time bound of $O((n*d!)^{c*\sqrt{n}+c'})$.

## 5. The Probabilistic Variant

Given a method to generate a set X of uniformly distributed independent random elements of the groups $H_0$, we can modify the algorithms of the previous section to be probabilistic. Since there is a method for verifying whether all coset representatives have been found, the modified algorithm always gives a correct answer, but only with an expected running time. Note that the algorithm includes coin tossing, which intuitively serves to make wasteful computations improbable. The ideas on which the modifications are based are due largely to Babai [1], and we only sketch them here.

Let $\{x_1, .., x_r\}$ be a uniformly distributed set of independent random elements of a group G which has a subgroup H. If $\{b_1, .., b_s\}$ is a complete set of coset representatives for G:H, i.e. $G = b_1H + \ldots + b_sH$, then we can derive a set $\{y_1, .., y_r\}$ of uniformly distributed, independent random elements. This is done by considering the elements $x_i$: if $x_i$ belongs to the j-th coset of G:H, i.e. $b_j^{-1}x_i \in H$, then define $y_i = b_j^{-1}x_i$. The resulting set will have the desired properties. We can use this device to "push" a set $\{x_i\}$ of uniform random elements of G into a subgroup of G. Of course, this process can be iterated.

Observe that the groups $U_{k+1}:U_k$, for fixed degree d, have an especially nice structure, and that $x \in U_{k+1}:U_k$ can be represented as vector of elements of $S_d$, the symmetric group of d elements. It is not hard to generate elements in $S_d$ with a uniform distribution, thus we can easily do so for the elements of $U_{k+1}:U_k$. These are first used to determine the coset representatives of $H_0:H_1$, then pushed down to the $H_1$ level and used to determine coset representatives of $H_1:H_2$, pushed to the $H_2$ level, and so on.

The running time of this procedure must be estimated by determining the cost of generating a random element in $H_0$, the cost for pushing it across a subgroup level, and the number of elements to be generated so that the probability for having all coset representatives for each of the quotients exceeds 1/2. The details of this will be given in the full paper. For example, for regular cones of degree 2, $O(n^{3\log(n)+\log\log(n)+c})$ steps is the expected running time for deciding isomorphism.

## 6 Further Remarks

As Gary Miller observed [7], for regular cones of degree 2 the groups $U_{k+1}:U_k$ form a vector space over the field of integers mod 2. It is known that $A_1(G)$ can be found in $O(n^2)$ deterministic steps as linear subspace of $U_1$. The idea is to find, by inspecting nontree edges connecting leaves, a set of linear equations constraining the 0/1 assignment to the rank 1 vertices of the graph.

The procedure just sketched is of limited value in testing 1-isomorphism because the converse of the following result is not true.

Theorem 6.1 If G is a regular cone of degree 2, x a 0/1 assignment to its rank 1 vertices specifying a leaf permutation, then $A_1(G) = A_1(G_x)$, $G_x$ the graph obtained from G by performing x.

Note also, that the group $B_1(G+G')$, formed in Section 4 to test 1-isomorphism is not a linear subspace of some vector space, for it is not Abelian. But there is a deterministic algorithm for deciding 1-isomorphism of regular cone graphs of degree 2. For this we can derive a system of equalities and inequalities for the 0/1 assignment to the rank 1 vertices establishing an isomorphism from a suitably colored superposition of the two graphs.

Theorem 6.2 1-isomorphism of a regular cone of degree 2 can be tested in $O(n^2)$ deterministic steps.

Note that this result enables us to test membership in $A_2U_1:U_1$ deterministically, but we were not able to find a basis for this subspace in efficient deterministic time bounds.

Acknowledgements Gary Miller contributed through many discussions to the germination of the ideas. Thanks are due to P. Young, H. Rubin, M. O'Donnell and M. Machtey who were drafted as audience in earlier stages of this work and found a number of mistakes.

## References

1. L. Babai, Monte Carlo Algorithms in Graph Isomorphism Testing, submitted to SIAM J. on Computing (1979)

2. J. Hopcroft and R. Tarjan, A VlogV Algorithm for Isomorphism of Triconnected Planar Graphs, J. of Comp. and Sys.Sci. 7(1973) 323-331

3. J. Hopcroft and J. Wong, A Linear Time Algorithm for Isomorphism of Planar Graphs, 6th STOC (1974) 172-184

4. R. Lipton, The Beacon Set Approach to Graph Isomorphism, SIAM J. on Comp. (to appear)

5. G. Lueker and K. Booth, A Linear Time Algorithm for Deciding Interval Graph Isomorphism, JACM 26 (1979) 183-195

6. G. Miller, On the $n^{\log(n)}$ Isomorphism Technique, 10th STOC (1978) 51-58

7. G. Miller, Personal Communication

8. M. Hall Jr., The Theory of Groups, MacMillan, New York 1959

9. G. Miller, Isomorphism Testing For Graphs of Bounded Genus, Manuscript (1979)