NORMAL FORMS FOR TRIVALENT GRAPHS AND GRAPHS OF BOUNDED VALENCE

Martin Fürer
Institut für
Angewandte Mathematik
Universität Zürich
CH - 8001 Zürich

Walter Schnyder
Mathematisches Seminar
ETH Zürich
CH - 8092 Zürich

Ernst Specker
Mathematisches Seminar
ETH Zürich
CH - 8092 Zürich

Abstract

A function f is defined, mapping graphs with n vertices onto graphs with vertex set $\{1,\ldots,n\}$. $f(X)$ is isomorphic to X and X is isomorphic to Y iff $f(X) = f(Y)$. For each d, the restriction of f to graphs of valence d is computable in time $O(n^{\tau(d)})$ for a suitable integer $\tau(d)$.

For $d > 3$, the proof uses a recent result of L. Babai, P.J. Cameron and P.P. Pálfy on the order of primitive groups with bounded composition factors; for the trivalent case a more elementary proof is presented.

1. Introduction

The graph isomorphism problem (GIP) is the problem of deciding, whether two graphs are isomorphic; it is clearly in NP. GIP is neither known to be in P nor to be NP-complete.

Traditionally, the solution of an isomorphism problem is attempted via the problem of determining a complete system of invariants; for the special case of graph theory see [11,12] (invariants are sometimes called certificates). In spite of the discovery of many interesting invariants, the solution of the isomorphism problem has been attained in this way only for rather special classes of graphs.

Recently, a breakthrough has been achieved in the original isomorphism problem by introducing group theoretic methods [1,6,10]. Taking into consideration that a complete system of invariants is much more useful for applications (e.g. chemical documentation) than a mere test for isomorphism, it is natural to try to apply these new methods to the problem of invariants.

In [10], Luks mentions that the problem of a complete system of invariants can be reduced to the following problem:

Input: Generators for $G \subset S_n$, an n-digit binary number m.
Find: The greatest number in the G-orbit of m (S_n acts on n-digit numbers by permuting digits).

However, this problem has turned out to be NP-hard [10,13] even for abelian 2-groups G. On the other hand even for Γ_d-groups G which appear in the d-valent case, our method implies that a slightly changed problem is feasible. We can find a rearrangement (depending on G) of the n digital positions such that for every m the maximum in the G-orbit of m can be found in polynomial time.

The algorithm presented in this paper assigns to every graph a normal form, i.e. an isomorphic copy. It should perhaps be remarked that the problems of canonical labelings and normal forms are equivalent for graphs of bounded valence.

The paper is divided in 2 parts. In the first part the trivalent case is treated. In section 2, for trivalent graphs with a distinguished edge a binary tree is defined in a canonical way; the leaves of the tree are the vertices of the given graph. In section 3, normal forms of these structures are computed by solving a sequence of embedding problems.

In the second part (sections 4,5), the general case of graphs of bounded valence is treated. The combinatorial method of section 2 cannot be used here directly. The binary trees are replaced in section 4 by "imprimitivity trees"; an analysis of the action of edge stabilizers in automorphism groups permits to generalize the methods of section 3 for computing normal forms of graphs of

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1983 ACM 0-89791-099-0/83/004/0161 $00.75

bounded valence by solving embedding problems (section 5).

The main result of this paper has been presented to the AMS [5]; in the meantime, we have learned that it has also been obtained by Babai and Luks [4].

2. Construction of a binary tree in a trivalent graph

We want to construct a binary tree whose leaves are the vertices of the given trivalent graph. Binary trees are formalized in the following manner.

Definition: *Binary trees* include the empty tree $\emptyset$. All other trees have a root. Every vertex is either a leaf or has two successors. For pairs of disjoint trees $\{T_L, T_R\}$ the tree $T_L T_R$ $(= T_R T_L)$ is defined as follows:

If $T_L = \emptyset$ then $T_L T_R = T_R$;
if $T_R = \emptyset$ then $T_L T_R = T_L$.

Otherwise the *product* $T_L T_R$ is the tree whose root has as successors the roots of T_L, T_R. We call $T_L (T_R)$ the left (right) subtree of T. This is, however, not a canonical concept, as left and right are defined only by the presentation of the trees. We have therefore to make sure that the final result of our algorithm does not depend on the presentation of the input.

Let X be a connected graph with distinguished edge e.

Definition: [10] X_r is the subgraph of X which appears in paths of length $\leq r$ through e. The *level* of a vertex v of X is the smallest r such that v is a vertex of X_r.

Definition: A *bipartite graph* X is a triple $(U(X), V(X), E(X))$ with $E(X) \subset U(X) \times V(X)$.

A bipartite graph is called "special", if vertices of U have degree at most 2, vertices of V at most 3 and at least 1.

Lemma 1: *There exists a function* T' *canonically defined (by a fast algorithm) on pairs* (T,X), *where* X *is a special bipartite graph and* T *a binary tree with leaf set* $U(X)$ *such that the value of* T' *is a binary tree with leaf set* $V(X)$.

Proof: In the computation of T' for a pair (X_o, T_o), we recursively compute T' for subgraphs X of X_o and trees T contained in T_o. $U(X)$ is abbreviated by U, $V(X)$ by V; the tree defined by T' is called $T'(T,V)$.

We procede by recursion on $|U|$, the size of U. The case $|U| = 1$ is trivial.

Otherwise

$$T = T_L T_R .$$

Let U_ρ be the leaf set of T_ρ $(\rho \in \{L,R\})$. For all $v \in V$ with two edges from U_ρ and one edge from U_σ, we omit the two edges from U_ρ. (Either $\rho = L$ and $\sigma = R$ or vice versa.)

Now

$$V = V_L \cup V_R \cup V_B ,$$

where the vertices of V_ρ have edges only from U_ρ, and the vertices of V_B have one edge from both U_L and U_R.

Now $T'(T,V)$ has the form

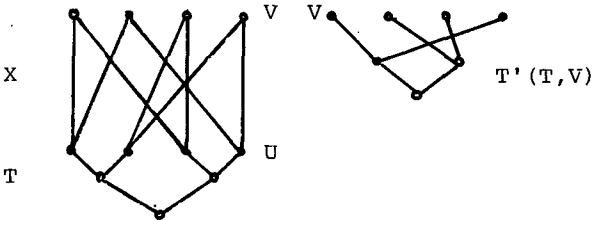
$$T'(T,V) = (T'(T_L,V_L)\ T'(T_R,V_R))\ T" ,$$

where $T"$ is defined as follows:
If $|U_\rho| = 1$ then $T" = T'(T_\sigma, V_B)$.
Otherwise (non trivial case)

$$T_L = T_{LL}\ T_{LR} ,\quad T_R = T_{RL} T_{RR} ;$$

we then define

$$T" = (T'(T_{LL}T_{RR}, V_B)T'(T_{LR}T_{RL}, V_B))(T'(T_{LL}T_{RL}, V_B)$$
$$T'(T_{LR}T_{RR}, V_B)) .$$



This is the crucial point of the tree construction; it cannot be generalized to higher valences.

Left and right subtrees have always been handled symmetrically, i.e. $T'(T,V)$ is constructed in a canonical way. Therefore the claim of the Lemma holds.

Theorem 1: *There is a polynomially time bounded (deterministic) algorithm accepting as input connected trivalent graphs with distinguished edge and producing for such an* (X,e) *a binary tree* T *whose leaves are the vertices of* X . T *is produced in a canonical way; therefore, every automorphism of* (X,e) *induces an automorphism of* T .

Proof: Let B_r be the following bipartite special graph: parts of B_r are level r and level $r-1$ of X ; edges of B_r are edges of X connecting points of the two parts.

T_1 is the binary tree with two leaves which are the vertices of the distinguished edge. T_r is defined by T_{r-1} and B_r as in Lemma 1. T is defined as the product $T_1(T_2(T_3...))$. Clearly all this can be done in polynomial time.

3. Normal forms for binary trees with a relation on the leaves

Canonical labeling of trees

We use a well known canonical labeling of (the vertices of) binary trees. Labels are defined and ordered as follows:

0 is a label and

$0 \leqslant \ell$ for all labels ℓ ;

if ℓ_1 and ℓ_2 are labels with $\ell_1 \leqslant \ell_2$ then (ℓ_1,ℓ_2) is a label and

$(\ell_1,\ell_2) \leqslant (\ell_1',\ell_2')$ if $\ell_1 < \ell_1'$ or $\ell_1 = \ell_1'$ and $\ell_2 \leqslant \ell_2'$.

Definition: The *canonical labeling* of a binary tree is given by

label $(v) = 0$ for all leaves v ;

if u,v,w are the roots of T,T',TT' and label $(u) \leqslant$ label (v) then

label $(w) =$ (label (u), label (v)) .

Two vertices are the roots of isomorphic subtrees iff they have the same labels.

A binary tree with a relation on the leaves is a triple (V_o,T_o,R_o) where T_o is a binary tree with leaf set V_o , and R_o is a binary relation on V_o .

Definition: The *normal form* of (V_o,T_o,R_o) is the isomorphic structure (V,T,R) with the following properties:

a) $V = \{1,...,n\}$ with $n = |V| = |V_o|$.

b) If T is arranged in the plane such that the leaves are $1,2,...,n$ from left to right, then

label (left successor of v)
$\leqslant$ label (right successor of v)

for every vertex v of the tree.

c) For all images S of R_o under isomorphisms from (V_o,T_o) to (V,T) the binary number matrix(S)

$= S(1,1)\ S(1,2)\ ...\ S(1,n)\ S(2,1)\ ...$
$...\ S(2,n)...\ S(i,j)\ ...\ S(n,n)$

assumes its maximal value for $S = R$.

Theorem 2: *The normal form* (V,T,R) *can be computed in time polynomial in* $n = |V_o|$.

Proof: R is built gradually, by adding more and more pairs to the relation, starting from the relation $\emptyset$.

For $1 \leqslant i \leqslant n$ and $0 \leqslant j \leqslant n$ we define

$$R_{ij} = R \cap \{(u,v) : u < i \text{ or } u = i \text{ and } v \leqslant j\}$$

Note that R is not yet known but well defined.

Then $R_{10} = \emptyset$, $R_{i+1,0} = R_{in}$, $R = R_{nn}$ and the maximality property of matrix(R) implies

$$R_{i,j+1} = \begin{cases} R_{ij} \cup \{(i,j+1)\} \text{ if } (V,T,R_{ij} \cup \{(i,j+1)\}) \\ \quad \text{is embeddable in } (V_o,T_o,R_o) \\ R_{ij} \text{ otherwise} \end{cases}$$

The difficult task is now to use this condition for computing $R_{i,j+1}$ from R_{ij} by doing an embedding test.

In general, graph embedding is a hard problem; in our case however the binary tree structure permits to convert it into a polynomially bounded number of isomorphism problems.

We use a basic property of trees of bounded degree:

Under all possible isomorphisms $f: T \to T_o$,

only a small number of images $\{f(1),\ldots f(j)\}$ of initial segments of leaves are possible. For binary trees, the situation is especially simple, because

$f(j)$ already defines $\{f(1),\ldots f(j)\}$ for all j .

Now assume that there is an embedding f from $(V,T,R_{ij} \cup \{(i,j+1)\})$ to (V_o,T_o,R_o) , and we know $f(i) = u_o$ and $f(j+1) = v_o$.

Then we also know

$$R_o(i,j+1,u_o,v_o) = R_o \cap \{(f(u),f(v)) : u < i \text{ or } u = i \text{ and } v \leq j+1\} .$$

And the maximality property of matrix(R) implies

$$f(R_{i,j+1}) = R_o(i,j+1,u_o,v_o) .$$

Actually we do not know u_o and v_o in advance. But we can just test for all u_o and v_o in V_o , whether there exists an isomorphism f from

$(V,T,R_{ij} \cup \{(i,j+1)\})$ to $(V_o,T_o,R_o(i,j+1,u_o,v_o))$.

Hence we have transformed an embedding problem into isomorphism problems.

For all $(u_o,v_o) \in R_o$, we wanted to solve

Problem 1: Does there exist an embedding f of $(V,T,R_{ij} \cup \{(i,j+1)\})$ into (V_o,T_o,R_o) with $f(i) = u_o$ and $f(j+1) = v_o$?

We have reduced Problem 1 to

Problem 2: Does there exist an isomorphism f between $(V,T,R_{ij} \cup \{(i,j+1)\})$ and $(V_o,T_o,R_o(i,j+1,u_o,v_o))$?

Clearly Problem 2 can be solved by the method of Luks [10]. In fact, it is easy to give generators of the automorphism group of $(V \cup V_o, TT_o, \emptyset)$. This is a 2-group. We let it act on $(V \cup V_o) \times (V \cup V_o)$ and ask for generators stabilizing $R_{ij} \cup \{(i,j+1)\} \cup R_o(i,j+1,u_o,v_o)$. Luks [10, section 2.2] has given a polynomial algorithm for constructing such a set of generators. Now we have just to see whether (at least) one generator interchanges the two trees, and Problem 2 is solved. This proves Theorem 2.

Remark: Naturally Theorem 2 holds for a bounded number of relations of bounded arity on the vertices of trees of bounded degrees.

Corollary: *Normal forms of trivalent graphs can be computed in polynomial time.*

Proof: For graphs with a distinguished edge e , we just compose the algorithms of Theorem 1 and 2. Otherwise, we try all possibilities for e and choose one with maximal matrix(R) .

4. Interval numberings

In section 2, starting from a trivalent graph X and a distinguished edge e of X , a binary tree T has been defined in a canonical way. The leaves of T are the vertices of X and the automorphism group $Aut(T)$ of T contains the stabilizer of e in $Aut(X)$.

In section 3, special embedding problems have been solved using the fact that orderings of the leaves of T exist whose initial segments have few images under $Aut(T)$.

This last property generalizes to graphs of bounded valence:

If G is the stabilizer of an edge e in the automorphism group of a d-valent graph X , then forests exist whose leaves are the vertices of X and can be ordered in such a way that initial segments of the orderings have few images under G (d being a constant).

We call some of these forests "imprimitivity forests", the corresponding orderings of their leaves "interval orderings".

Imprimitivity forests may have unbounded degrees. This is an essential difference to the trivalent case, as initial segments of an interval ordering of the leaves of an imprimitivity forest may have an exponential number of images under the automorphism group of this forest (even when all trees of the forest are stabilized). In particular, this implies that a two-phased process (1. constructing a forest, 2. solving embedding problems by use of the automorphism group of the forest) is no more possible. Forests and normal forms will have to be computed simultaneously.

In section 4, we define the notions of imprimitivity forest and interval ordering; we prove the above mentioned property of initial segments of these orderings.

In section 5, we generalize the embedding method, show how imprimitivity forests can be constructed in a canonical way and finally define polynomially computable normal forms for graphs of bounded valence.

Terminology

For terminology on permutation groups, the reader is refered to [10]; further information can be found in [6,8,10,14]. We shall use the following notation:

$Sym(V)$ for the group of permutations of V ; $[G,V]$ for an action of the group G on V ; $\mathcal{V}(X)$ for the set of vertices, $\mathcal{E}(X)$ for the set of edges of a graph X .

A *numbering* of a structure is a bijection from the underlying set V to $\{1,2,\ldots,|V|\}$. If a and b are numberings of the structure S , then $aS = bS$ iff $b \in a\,Aut(S)$. If $G \subset Sym(V)$, a *numbering class* for (G,V) is a coset aG , where a is a numbering of V . In computational context, numbering classes for (G,V) will be represented by a numbering and less than $|V|^2$ generators of G . A numbering class for a structure $S = (V,R_1,\ldots,R_m)$ is a numbering class for $(Aut(S),V)$. Thus a numbering class for a graph X with distinguished edge e is of the form $a\,Aut_e(X)$, where $Aut_e(X)$ is the stabilizer of the edge e in $Aut(X)$.

The type of a structure $S = (V,R_1,\ldots,R_m)$ is $(k_1,\ldots,k_m)$, where k_i is the arity of R_i . Subsets of an ordered set are ordered by the lexicographic ordering of their bit vector representations. If V is an ordered set, V^k is lexicographically ordered. k-ary relations on V are subsets of V^k . Structures of the same type on an ordered set V are ordered by the lexicographic ordering of their relation sequences.

The efficiency of our algorithm for computing normal forms relies on properties of the edge stabilizers in the automorphism group of graphs of bounded valence.

Abstracting from the graphs, Luks has characterized essential properties of these edge stabilizers [10]:

Definition: For $d \geq 2$, Γ_d is the class of groups G such that all composition factors of G are subgroups of S_d .

Subgroups and homomorphic images of Γ_d-groups are again Γ_d-groups.

For each d-valent graph X with a distinguished edge e , $Aut_e(X) \in \Gamma_{d-1}$.

A consequence of the rich imprimitivity structure of Γ_d-groups is

Theorem 3 (E.Luks) [10]: *Isomorphism of structures consisting of polynomially many relations of bounded arity on the vertices of graphs of bounded valence can be tested in polynomial time.*

The next theorem is crucial for the extension to graphs of bounded valence of the embedding method of part I.

Theorem 4 (L.Babai, P.J. Cameron, P.P. Pálfy) [2]: *There is a function $t(d)$ such that any primitive Γ_d-group $G \subset S_n$ has order at most $n^{t(d)}$.*

We now define numbering classes of a kind suitable to reflect the properties of imprimitivity structures of Γ_d-groups.

Definition: Let G be a group acting transitively on a set V . An *imprimitivity tree* for $[G,V]$ is a sequence $R_1,\ldots,R_m$ of G-invariant equivalence relations on V satisfying:

a) R_1 is the relation with only one class. R_m is the equality.

b) $R_{i+1} \subsetneq R_i$ $(1 \leq i < m)$.

c) For all $B \in V/R_i$, B/R_{i+1} is a minimal blocksystem for $[G_B,B]$.

The classes of $R_1,\ldots,R_m$ are called blocks of the tree.

Note that condition (c) is satisfied by all $B \in V/R_i$, if it is satisfied by some $B \in V/R_i$.

Definition: If G is a group acting on a set V , an *imprimitivity forest* for $[G,V]$ is a set consiting of an imprimitivity tree for $[G,O]$, for each orbit O .

Definition: Let $G \subset Sym(V)$. A numbering a of (G,V) is an *interval numbering* if an imprimitivity forest for $[G,V]$ exists whose blocks are intervals of the ordering induced by a on V .

An *interval numbering class* is a numbering class consisting of interval

numberings.

Lemma 2: *If* $G \subset \mathrm{Sym}(V)$ *is a* Γ_d*-group and* $\bar{a}$ *is an interval numbering of* (G,V)*, then for each initial segment* I *of* V *the following holds:*

(1) $|G/G_I| \leq |V|^{t(d)}$
(I *has at most* $|V|^{t(d)}$ *distinct images under* G *.)*

(2) The images of I *under* G *are computable in polynomial time from generators of* G *.*

Proof:

(1) By the definition of interval numberings $I = O_1 \cup O_2 \cup \ldots \cup O_{r-1} \cup U$, $U \subset O_r$, where $O_1,\ldots,O_r$ are orbits of G. $G_I = G_U$.
U is an initial segment of O_r, thus $U = B_1 \cup B_2 \cup \ldots \cup B_{s-1} \cup V$, $V \subset B_s$ for blocks $B_1,\ldots,B_s$ of a minimal blocksystem S of $[G,O_r]$.

Let K denote the stabilizer in G of all blocks of S. $G_U \supset K_V$ and

$$|G/G_I| = |G/G_U| \leq |G/K_V| = |G/K| \cdot |K/K_V| \leq |G/K| \cdot |G_{B_s}/G_V| .$$

G/K is isomorphic to the group of permutations of S induced by G. This is a primitive Γ_d-group of degree $|S| = \frac{|O_r|}{|B_s|}$.

By Theorem 4

$$|G/G_I| \leq \left(\frac{|O_r|}{|B_s|}\right)^{t(d)} \cdot |G_{B_s}/G_V| .$$

Furthermore the restriction of the imprimitivity tree of $[G,O_r]$ to B_s is an imprimitivity tree for $[G_{B_s},B_s]$, V is an initial segment of B_s, and G_V is the stabilizer of V in the Γ_d-group G_{B_s}.

Let $\mathrm{Ind}(n)$ be the largest possible index of stabilizers of initial segments in Γ_d-groups on n points (the orderings being induced by interval numberings).

Setting $n = |V|$, $b = |B_s|$, we obtain the recurrence

$$\mathrm{Ind}(n) \leq \left(\frac{n}{b}\right)^{t(d)} \cdot \mathrm{Ind}(b)$$

with the consequence $\mathrm{Ind}(n) \leq n^{t(d)}$.

(2) The index of G_I in G is polynomial in $|V|$ and we have a membership test for G_I.

Representatives of G/G_I are thus computable in polynomial time [6].

5. Normal forms for graphs of bounded valence

Lemma 3: *There exists a deterministic algorithm with*

Input: d*-valent graph* X *with distinguished edge* e ;
interval numbering a *for* (X,e) ;
relations $R_1,R_2,\ldots,R_m$ *of arity* $\leq d$ *on the vertices of* X .

Output: $b \in a\,\mathrm{Aut}_e(X)$ *such that* $b(R_1,\ldots,R_m) = \max a\,\mathrm{Aut}_e(X)\,(R_1,\ldots,R_n)$

Cost: *For each* d *a polynomial* $p_d(|X|,m)$.

Remark: The resulting numbering class $b\,\mathrm{Aut}_e(X,R_1,\ldots,R_m)$ is thus independent of the representative a chosen from the interval numbering class $a\,\mathrm{Aut}_e(X)$.

Proof of Lemma 3 (reproduces the proof of Theorem 2)

We only consider the case where $m = 1$, R is a binary relation.

Let $n = |X|$ and $Y = aX$ (the isomorphic image of X in $\{1,\ldots,n\}$ under a, with the distinguished edge ae).
$a\,\mathrm{Aut}_e(X)$ is precisely the set of all isomorphisms from (X,e) to (Y,ae).

Define $S = \max a\,\mathrm{Aut}_e(X)\ R$ and
$S_{ij} = S \cap [\{1,\ldots,i-1\} \times \{1,\ldots,n\} \cup \{i\} \times \{1,\ldots,j\}]$.

S will be constructed in n^2 steps by successively building the relations $S_{11},S_{12},\ldots,S_{1n},S_{21},\ldots,S_{2n},\ldots,S_{ij},\ldots,S_{nn}$ in this order.

Determining S_{ij}

Define $T = \begin{cases} S_{ij-1} \cup \{(i,j)\} & \text{if } j \geq 2 \\ S_{i-1\,n} \cup \{(i,j)\} & \text{if } j = 1 \end{cases}$

($S_{on} = \emptyset$).

The maximality property of S implies:

$$S_{ij} = T \quad \text{or} \quad S_{ij} = T - \{(i,j)\} ,$$

and

$S_{ij} = T$ iff there exists an isomorphism $\sigma : (Y,ae) \to (X,e)$ embedding T into R.

The properties of interval numberings of graphs of bounded valence enable us to convert this embedding problem into instances of isomorphism problems.

By the maximality property of S each isomorphism $\sigma : (Y,ae) \to (X,e)$ embedding T into R is an isomorphism from (Y,ae,T) to (X,e,R'), where

$$R' = R \cap [\sigma\{1,\ldots,i-1\} \times V \cup \{\sigma i\} \times \sigma\{1,\ldots,j\}] .$$

The images $\sigma\{1,\ldots,k\}$ are of the form $ga^{-1}\{1,\ldots,k\}$ ($g \in \mathrm{Aut}_e(X)$). For every k there are at most $n^{t(d-1)}$ such images, all of them being computable in polynomial time (Lemma 2).

We are thus left with at most $n^{1+2t(d-1)}$ possible images R' of T, and determining S_{ij} is reduced to the following:

Determine for each of the possible images R' of T whether (Y,ae,T) is isomorphic to (X,e,R').

Finally, a last isomorphism test on (Y,ae,S) and (X,e,R) will produce a numbering b with the desired property.

The algorithm of Lemma 3 expects interval numbering classes as input. Lemma 4 shows that we can provide such classes.

Lemma 4: *There is a polynomially time bounded deterministic algorithm with*

Input: *numbering class for* (G,V) *where* $G \subset \mathrm{Sym}(V)$.

Output: *imprimitivity forest and interval numbering class for* (G,V).

The resulting imprimitivity forest and interval numbering class are independent of the presentation of the initial numbering class.

Proof:

Imprimitivity forest for $[G,V]$

Let $a : V \to \{1,\ldots,|V|\}$ be a representative of the initial numbering class.

a induces in a canonical way an ordering $<$ of V.

Case 1: G is intransitive on V.

Partition V into orbits of G. For each orbit O construct an imprimitivity tree for $[G,O]$ by application of Case 2.

Case 2: G is transitive on V.

Construct an imprimitivity tree $R_1,\ldots,R_m$ for $[G,V]$.

Step 1: R_1 is the equivalence relation on V with the only class V.

Step i+1:
Define $v = \min V$ and let B be the equivalence class of v modulo R_i.

The subsets of B are linearly ordered by the lexicographic ordering on their bit vector representations.

The block $C \neq B$ of $[G_B,B]$ which is maximal with respect to this ordering, can be determined by repeatedly testing the existence of blocks $D \neq B$ containing subsets of B.

But C is also maximal in the sense that no block D of $[G_B,B]$ exists with $C \subsetneq D \subsetneq B$. Therefore, $G_B C = \{gC : g \in G_B\}$ is a minimal blocksystem for $[G_B,B]$, and we can define R_{i+1} to be the relation with the equivalence classes GC (computable in polynomial time).

If $|G| = 1$ then we set $m = i+1$ else we continue the process.

Interval numbering class for (G,V)

A new linear ordering $<'$ of V is constructed such that orbits and blocks

of the imprimitivity forest are intervals of $<'$.

(1) Definition of $<'$ on an orbit O of G .

Let $R_1,\ldots,R_m$ be the above constructed imprimitivity tree for $[G,O]$.

We successively define orderings $<_i$ of O/R_i $(i=1,\ldots,m)$, then we set $<' = <_m$.

Step $i+1$: Let $A,B \in O/R_{i+1}$ and $\bar{A},\bar{B} \in O/R_i$ with $A \subset \bar{A}$, $B \subset \bar{B}$. Define $A <_{i+1} B \iff \bar{A} <_i \bar{B}$ or $\bar{A} = \bar{B}$ and $\min A < \min B$ (the minimum being taken with respect to the initial ordering $<$ of V).

(2) For $v_1 \in O_1$, $v_2 \in O_2$, $O_1 \neq O_2$ orbits of G , define $v_1 <' v_2 \iff \min O_1 < \min O_2$.

$<'$ determines an interval numbering $a' : V \to \{1,\ldots,|V|\}$. The resulting interval numbering class is $a'G$.

Uniqueness of the output

Let a and $b = ag$ $(g \in G)$ be representatives of the same initial numbering class, inducing the orderings $<_a$ and $<_b$ of V .

Define $t_g : v \in V \mapsto gv \in V$ and
$\alpha_g : h \in G \mapsto ghg^{-1} \in G$.

$T_g = (t_g,\alpha_g)$ is an isomorphism from $(V,<_b,G)$ to $(V,<_a,G)$:

- $t_g : (V,<_b) \to (V,<_a)$ and $\alpha_g : G \to G$ are isomorphisms,
- for all $v \in V$ and $h \in G$, $t_g(hv) = \alpha_g(h)\, t_g(v)$.

Clearly t_g must be an isomorphism from $(V,<'_b)$ to $(V,<'_a)$, and the induced interval numberings define the same interval numbering class.

Furthermore t_g is an isomorphism of the imprimitivity forests. As these forests are G-invariant, they are identical.

Theorem 5: There is a function f , mapping graphs with n vertices to graphs with vertex set $\{1,\ldots,n\}$ such that $f(X)$ is isomorphic to X and X is isomorphic to Y iff $f(X) = f(Y)$.

For each d , the restriction of f to d-valent graphs is computable in time $O(n^{\tau(d)})$ for a suitable integer $\tau(d)$.

Proof: It is clearly sufficient to define f on the class of connected graphs.

A function ϕ mapping a connected graph with distinguished edge (X,e) to a graph with vertex set $\{1,\ldots,|X|\}$ will be defined such that:

(1) $\phi(X,e) \cong X$

(2) $(X,e) \cong (X',e') \Rightarrow \phi(X,e) = \phi(X',e')$

The function f defined by

$$f(X) = \max_{e \in \mathcal{E}(X)} \phi(X,e)$$

will then satisfy all conditions.

Defining ϕ

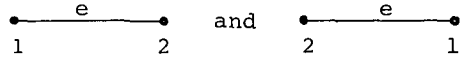
Let X be a d-valent graph with distinguished edge e .

As in [10], subgraphs X_r $(r=1,2,\ldots)$ of X are successively considered. X_r consists of all vertices and edges of X appearing in paths of length $\leq r$ through e .

A unique numbering class for X_r is computed for each r .

The process ends as soon as $X_r = X$ by letting $\phi(X,e)$ be the image of (X,e) under its numbering class.

X_1 has only the edge e . The numbering class for X_1 consists of two numberings:



Numbering class for X_{r+1}

Let a $\mathrm{Aut}_e(X_r)$ be the numbering class for X_r resulting from the last step.

This class may split into several cosets of $\pi_r(\mathrm{Aut}_e(X_{r+1}))$ (the restriction of $\mathrm{Aut}_e(X_{r+1})$ to X_r) . A direct extension to a numbering class for X_{r+1} is thus

impossible.

Therefore, we will proceed in two phases:

Selection of a numbering class for $(\pi_r(Aut_e(X_{r+1})),X_r)$,

extension of this class to a numbering class for X_{r+1} .

Selection

Let A be the set of all subsets of $\mathcal{V}(X_r)$ and define (as in [10])

$$F : \mathcal{V}(X_{r+1}) - \mathcal{V}(X_r) \to A$$

by $F(v) = \{w \in \mathcal{V}(X_r) : \{v,w\} \in \mathcal{E}(X)\}$

Notice that $1 \leq |F(v)| \leq d$ and that for each subset B of $\mathcal{V}(X_r)$: $|\{v : F(v) = B\}| \leq d-1$.

Define the d-ary relations $R_1,\ldots,R_{d-1}$ on $\mathcal{V}(X_r)$ by

$R_i(w_1,\ldots,w_d) \iff i = |\{v \in \mathcal{V}(X_{r+1}) - \mathcal{V}(X_r) :$

$F(v) = \{w_1,\ldots,w_d\}\}|$

Furthermore, define the binary relation R_d on $\mathcal{V}(X_r)$ by

$R_d(w_1,w_2) \iff \{w_1,w_2\} \in \mathcal{E}(X_{r+1}) - \mathcal{E}(X_r)$.

As shown in [10]:
$Aut_e(X_r,R_1,\ldots,R_d)$ is precisely $\pi_r(Aut_e(X_{r+1}))$.

We now apply the earlier lemmas:

(i) Using Lemma 4, we transform the numbering class a $Aut_e(X_r)$ into an interval numbering class b $Aut_e(X_r)$.

(ii) Using Lemma 3, we obtain the numbering class c $Aut_e(X,R_1,\ldots,R_d)$ maximizing $(R_1,\ldots,R_d)$ under b $Aut_e(X)$.

Extension

Each numbering ν of $\mathcal{V}(X_r)$ induces a linear ordering of $\mathcal{V}(X_r)$ and thus an ordering $<_\nu$ of the subsets of $\mathcal{V}(X_r)$ in a canonical way.

An extension $\bar{\nu}$ of ν to a numbering of $\mathcal{V}(X_{r+1})$ will be called *admissible* if for all $v_1,v_2 \in \mathcal{V}(X_{r+1}) - \mathcal{V}(X_r)$

$$f(v_1) <_\nu f(v_2) \Rightarrow \bar{\nu}(v_1) < \bar{\nu}(v_2) .$$

One easily verifies:

- Any two admissible extensions of ν only differ by some element of $Aut_e(X_{r+1})$ stabilizing all vertices of X_r .
- If $\bar{\nu}$ is an admissible extension of ν and $\bar{g} \in Aut_e(X_{r+1})$ is an extension of $g \in \pi_r(Aut_e(X_{r+1}))$ then $\bar{\nu}\,\bar{g}$ is an admissible extension of νg .

The admissible extensions of the numberings in c $Aut_e(X,R_1,\ldots,R_m)$ therefore form an (easily computable) coset of $Aut_e(X_{r+1})$. We choose this coset as numbering class for X_{r+1} .

This proves that normal forms for graphs of bounded valence are computable in deterministic polynomial time.

With one more application of the Lemmas 4, 3 we obtain:

Corollary

Normal forms for structures consisting of polynomially many relations of bounded arity on the vertices of graphs of bounded valence are computable in polynomial time.

6. Remarks

1. Our algorithm for computing normal forms of trivalent graphs uses $O(n^3)$ isomorphism tests, each of which can be done in time $O(n^3)$, giving a total time (for fixed edge) of $O(n^6)$. It seems clear however that this bound can be improved drastically, because the isomorphism problems differ only slightly. Possibly also ideas from [7] can be used.

2. When applied to graphs with distinct eigenvalues, our method allows to reproduce a result of F.T. Leighton and G.L. Miller [9]. We hope that it can be generalized to the case of graphs with bounded eigenvalue multiplicity.

References

[1] L. Babai, Monte Carlo algorithms in graph isomorphism testing, manuscript, 1979

[2] L. Babai, P.J. Cameron, P.P. Pálfi, On the order of primitive permutation groups with bounded nonabelian composition factors, to appear

[3] L. Babai, D.Yu. Grigoryev, D.M. Mount, Isomorphism of graphs with bounded eigenvalue multiplicity, *Proc. 14th ACM Symp. Theory of Computing* (1982), 310-324

[4] L. Babai, E.M. Luks, Canonical labeling of graphs, this volume

[5] M. Fürer, W. Schnyder, E. Specker, Canonical labeling for graphs of bounded degree, *Abstract Amer. Math. Soc. 4, No. 2* (1983)

[6] M. Furst, J. Hopcroft, E. Luks, Polynomial-time algorithms for permutation groups, *21st IEEE Symp. on Foundations of Comp. Sci.* (1980), 36-41

[7] Z. Galil, C.M. Hoffmann, E.M. Luks, C.P. Schnorr and A. Weber, An $O(n^3 \log n)$ deterministic and an $O(n^3)$ probabilistic isomorphism test for trivalent graphs, *23rd IEEE Symp. on Foundations of Comp. Sci.* (1982)

[8] C.M. Hoffmann, Group-Theoretic Algorithms and Graph Isomorphism, Lecture Notes in Computer Science 136, Springer, Berlin, Heidelberg, New York, 1982

[9] F.T. Leighton and G. Miller, Certificates for graphs with distinct eigenvalues, in preparation

[10] E. Luks, Isomorphism of graphs of bounded valence can be tested in polynomial time, *J. Comp. Sys. Sci.* 25 (1982), 42-65

[11] G. Miller, Graph isomorphism, general remarks, *J. Comp. Sys. Sci.* 18 (1979), 128-142

[12] R.C. Read, D.G. Corneil, The graph isomorphism disease, *J. Graph Theory* 1 (1977), 339-363

[13] W. Schnyder, in preparation, 1982

[14] H. Wielandt, Finite permutation groups, Academic press, New York, 1964