CONSTRUCTIVE GRAPH LABELING USING DOUBLE COSETS

BY

HAROLD BROWN
LARRY MASINTER
LARRY HJELMELAND

STAN-CS-72-318
OCTOBER 1972

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

CONSTRUCTIVE GRAPH LABELING USING DOUBLE COSETS

Harold Brown
Larry Masinter
Larry Hjelmeland

Abstract

Two efficient computer implemented algorithms are presented for explicitly constructing all distinct labelings of a graph G with a set of (not necessarily distinct) labels L , given the symmetry group B of G . Two recursive reductions of the problem and a precomputation involving certain orbits of stabilizer subgroups are the techniques used by the algorithm. Moreover, for each labeling, the subgroup of B which preserves that labeling is calculated.

This research was supported by the Advanced Research Projects Agency SD-183.

CONSTRUCTIVE GRAPH LABELING USING DOUBLE COSETS[1]

By Harold Brown, Larry Masinter and Larry Hjelmeland

1. Introduction. We consider in this paper the following graph theoretical problem: Given a graph G with n nodes and topological symmetry group B and a set L of n not necessarily distinct labels, construct all topologically distinct labelings of the nodes of G with the elements of L. This problem arises in numerous contexts, and it has been investigated by Pólya [7], DeBruijn [4] and others. In particular, the number of such distinct labelings is given by the generalized Polya enumeration formula.[2] We present here two efficient computer implemented algorithms for explicitly constructing all topologically distinct labelings of G by L. Moreover, for each distinct labeling, the algorithms determine the subgroup of B which preserves that labeling.

Our interest in the graph labeling problem initially arose in the context of the DENDRAL project [2]. This project includes among its objectives the application of computer implemented artificial intelligence techniques to the analysis and classification of organic compounds. Necessary to this work are algorithms to systematically generate all the distinct valence isomers of a given set of atoms. Routines to perform this task in the special case where the isomers form only topologically tree-like structures have been described in [3] and [5]. For the general case, algorithms are required which generate all distinct cyclic structures formed from a given set of atoms with pre-assigned free valences. The graph labeling problem is central to these cyclic structure generation algorithms.[3]

We now describe a group theoretic approach to the graph labeling problem.

2. Algebraic formulation and notation. The graph labeling problem admits a completely algebraic formulation as follows:

We index from 1 to n the nodes of the graph G in some fixed order and index also from 1 to n the n labels in the set L where, for notational convenience, we index equal labels in sequence, i.e., if there are n_1 labels of the first type, n_2 labels of the second type, etc., then we index the labels of the first type with $1, \ldots, n_1$, the labels of the second type with $n_1+1, \ldots, n_1+n_2$, etc. With this indexing, any labeling of G by L can be considered as a bijective map from the integral interval $[1,n]$ (the node indices) to $[1,n]$ (the label indices). (Throughout, $[a.b]$ will always denote the interval of integers from a through b inclusive if $a \leq b$, and $[a,b] = \emptyset$ if $a > b$). Thus, the indexed labelings of G by L can be bijectively identified with S_n , the full permutation group on $[1,n]$.[4]

Any topological symmetry of G in the symmetry group B can be considered as a permutation of the node indices, i.e., B can be isomorphically identified with a subgroup B of S_n , and for $\alpha \in S_n$ and $\beta \in B$, the labelings α and $\alpha\beta$ correspond to topologically equivalent labeled graphs.

The indexed set of labels also admits a symmetry group. If there are n_1 labels of the first type, n_2 labels of the second type, . . . n_k labels of the k-th type, $n_1 + n_2 + . . . + n_k = n$, then the labels with indices in the intervals

$$I_j = [(\sum_{i=1}^{j-1} n_i) + 1,\ \sum_{i=1}^{j} n_i],\ j = 1,2, \ldots, k,$$

are indistinguishable as unindexed labels. These labels, therefore, may be freely permuted in any indexed labeling without changing the corresponding labeled graph. Hence, the indices of the labels admit the symmetry group $A = S_{(n_1)} \times S_{(n_2)} \times \ldots \times S_{(n_k)}$ where "X" denotes the (internal) direct product of subgroups in S_n and $S_{(n_j)}$ denotes the full group of permutations on the interval I_j naturally embedded in S_n. Explicitly, for $\alpha \in S_n$, α is in $S_{(n_j)}$ if and only if $\alpha(t) = t$ for $t \notin I_j$. Note that this latter condition implies that $\alpha(I_j) = I_j$ since α is bijective and $\{ I_j , [1,n]/I_j \}$ partitions $[1,n]$. The subgroup A will be called the <u>label subgroup</u> of S_n corresponding to the the (ordered) partition $n_1 + n_2 + . . . + n_k = n$ of n.

We now define a relation Δ on S_n by $\gamma_1 \Delta \gamma_2$ if and only if there exist $\alpha \in A$ and $\beta \in B$ such that $\gamma_1 = \alpha\gamma_2\beta$. Since A and B are subgroups of S_n, Δ is an equivalence relation on S_n. In terms of the graph G, γ_1 and γ_2 determine topologically equivalent labelings of the nodes of G with the labels in L if and only if $\gamma_1 \Delta \gamma_2$. Since Δ is an equivalence relation on S_n, the equivalence classes of Δ partition S_n. Hence, we can determine all topologically distinct labelings of G by L by selecting precisely one element from each distinct Δ-equivalence class, i.e., by selecting a representative set for the partition of S_n induced by A .

For any $\gamma \in S_n$, the Δ-equivalence class determined by γ is the set $C_\gamma = \{\alpha\gamma\beta \mid \alpha \in A, \beta \in B\}$, i.e., C_γ is the set product $A\gamma B$. This set product is called the double coset of A and B in S_n determined by γ . Thus our graph labeling problem can be algebraically formulated as follows:

Given a label subgroup A of S_n and a subgroup B of S_n, determine algorithmically a representative set for the double cosets of A and B in S_n, i.e., determine a subset $\{\gamma_1, \gamma_2, \ldots, \gamma_t\}$ of S_n such that $S_n = \bigcup_{i=1}^{t} A\gamma_i B$ and $(A\gamma_i B) \cap (A\gamma_j B) = \emptyset$ for $i \neq j$.

The correspondence between graph labeling and double cosets and the use of double cosets as a basis for chemical nomenclature have been investigated by Ruch, Hässelbarth and Richter [8].

Although the double coset formulation of the graph labeling problem presents the problem in a conceptually less obvious form, it does permit the techniques of constructive group theory to be applied directly to the problem. Moreover, our algebraic solutions are directly implementable on a computer.

2.1. Example. Let G be the graph in figure 1a. Let L consist of 3 labels N and 7 labels C . The topological symmetries of G are:

b_0: The identity transformation.

b_1: Reflection about the line l_1.

b_2: Reflection about the line l_2.

b_3: 180° rotation about the center of G.

Index the nodes of G as in figure 1b and the labels in L as $x_1 = x_2 = x_3 = N$ and $x_4 = \ldots = x_{10} = C$.
Then, the labelings of G by L can be considered as elements in S_{10}. E.g., the permutation $\gamma_1 = (1\ 2\ 5\ 6\ 7\ 3\ 8\ 4\ 9\ 10)$ in S_{10} corresponds to the labeling of G given in figure 2a and the permutation $\gamma_2 = (3\ 5\ 4\ 9\ 8\ 2\ 1\ 7\ 10\ 6)$ to the labeling in figure 2b. Here, we use the notation for S_n which identifies $\gamma \in S_n$ with the n-vector $(\gamma(1), \gamma(2), \ldots, \gamma(n))$.

The topological symmetry group of G determines the subgroup B of S_{10} via

$$b_0 \leftrightarrow \beta_0 = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10),$$

$$b_1 \leftrightarrow \beta_1 = (10\ 9\ 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1),$$

$$b_2 \leftrightarrow \beta_2 = (5\ 4\ 3\ 2\ 1\ 10\ 9\ 8\ 7\ 6),$$

$$b_3 \leftrightarrow \beta_3 + (6\ 7\ 8\ 9\ 10\ 1\ 2\ 3\ 4\ 5).$$

The label subgroup of S_{10} associated with L is $A = S_{(3)} \times S_{(7)}$, a subgroup of order 3!7! . For example, the permutation $\alpha = (2\ 1\ 3\ 4\ 7\ 10\ 6\ 5\ 9\ 8)$ is in A, and the permutations γ_1 and γ_2 are Δ-equivalent since $\gamma_2 = \alpha\gamma_1\beta_3$, i.e., the labeled graphs in figures 2a and 2b are topologically equivalent.

By Polya's enumeration formula, there are 32 distinct double cosets of A and B in S_{10}, i.e., there are 32 topologically distinct labelings of G by L.

3. <u>General theory</u>. Let A and B be subgroups of the finite group G. A straightforward group theoretic argument shows that the double cosets of A and B in G partition G. This partition, unlike a single coset partition of G, is generally not a partition into subsets of equal size, and there is no simple analogue to LaGrange's theorem. There is, however, a certain regularity in a double coset partition as evidenced by the following known theorem:

3.1. <u>Theorem</u>. For any $g \in G$, let R_g be a set of right coset representatives of $(g^{-1}Ag \cap B)$ in B. Then the double coset AgB consists precisely of the union of right cosets $\bigcup_{x \in R_g} Agx$.

Moreover, this union is disjoint. Symmetrically, if L_g is a set of left coset representatives of $(A \cap gBg^{-1})$ in A, then AgB is the disjoint union $\bigcup_{y \in L_g} ygB$.

<u>Proof.</u> Let $R_g = \{x_1, x_2, \ldots, x_k\}$, i.e., B is the disjoint union $\bigcup_{i=1}^{k} (g^{-1}Ag \cap B)x_i$, and let $u \in AgB$, say $u = agb$. Now b in B implies that $b = hx_i$ for some $1 \leq i \leq k$ and $h \in g^{-1}Ag \cap B$. Also, h is of the form $g^{-1}a_1g$, $a_1 \in A$. Thus $u = agg^{-1}a_1gx_i = (aa_1)gx_i$, and $u \in \bigcup_{x \in R_g} Agx$, i.e., $AgB = \bigcup_{x \in R_g} Agx$. If $Agx_i = Agx_j$, then $x_ix_j^{-1} = g^{-1}a_2g$ for some $a_2 \in A$. Since x_i and x_j are in B, $x_ix_j^{-1} \in g^{-1}Ag \cap B$. Therefore, $(g^{-1}Ag \cap B)x_i = (g^{-1}Ag \cap B)x_j$, and, since R_g is a right coset representative set for $g^{-1}Ag \cap B$

in B, we must have $i = j$. Hence the union is a disjoint union. |

For any finite set T, we denote by $|T|$ the number of elements in T. From Theorem 3.1 and LaGrange's theorem we have:

3.2. <u>Corollary</u>. $|AgB| = |A||B| \,/\, |g^{-1}Ag \cap B|$
$$= |A||B| \,/\, |A \cap gBg^{-1}| .$$

Theorem 3.1 does yield the following algorithmic method for determining a list D of double coset representatives of A and B in G:

1. Determine a list of right coset representatives for A in G, say $R = \{ a,b, \ldots, t \}$, and form the list D with initially $D = \emptyset$.
2. For the first member m in R, place m on D and determine a set of right coset representatives of $(m^{-1}Am \cap B)$ in B, say $T_m = \{ x,y, \ldots, z \}$.
3. For each w in T_m, determine the unique element h in R such that $Ah = Amw$ and eliminate h from the list R.
4. If $R = \emptyset$, stop, otherwise go to step 2.

The difficulty with the above algorithm is that any direct implementation is computationally prohibitive in terms of both machine time and core store even for relatively small groups, e.g., $G = S_{10}$. Our objective now is to derive certain modifications to this algorithm in the case $G = S_n$ and A is a label subgroup so that the modified algorithm admits efficient machine implementation. The main device used is the natural ordering of S_n.

The group S_n admits a natural linear ordering. This ordering is a very powerful computational tool, and it has been used by Sims [9] and others in devising group theoretic algorithms. The ordering is defined as follows:

Consider S_n as the set of bijective maps from $[1,n]$ to itself. For $\pi \in S_n$, identify π with the integral n-vector $(\pi(1),\pi(2), \ldots, \pi(n))$. Using this latter representation of S_n, the natural linear ordering is the lexicographical ordering on the n-vectors induced by the usual ordering on $[1,n]$, i.e., if we denote the order relation on S_n by "<<", then $\pi_1 << \pi_2$ if and only if either $\pi_1 = \pi_2$ or for some $k \in [1,n]$, $\pi_1(i) = \pi_2(i)$ for $1 \leq i < k$ and $\pi_1(k) < \pi_2(k)$. This relation can be extended to subsets of S_n via $T << U$ if and only if for every $\tau \in T$ and $\eta \in U$, $\tau << \eta$.

Given any partition P of S_n, this linear ordering permits us to easily specify a canonical representative set for P. Namely, we choose as the representative for $P \in P$ the "least" element in P with respect to $<<$, i.e., we choose the unique $\pi \in P$ satisfying $\pi << P$.

Let A and B be subgroups of S_n. The canonical representative sets for the right cosets of A in S_n, the left cosets of B in S_n, and the double cosets of A and B in S_n are

${}_AS_n = \{ \alpha \in S_n \mid \alpha << A\alpha \}$, ${S_n}_B = \{ \beta \in B \mid \beta << \beta B \}$
and ${}_A{S_n}_B = \{ \pi \in S_n \mid \pi << A\pi B \}$, respectively.[5] Since A and B

contain the identity element of S_n, if π is in ${}_AS_{nB}$, then π must satisfy $\pi << A\pi$ and $\pi << \pi B$. The converse, unfortunately, is not true. We will call a double coset representative set <u>small</u> if it is contained in ${}_AS_n$. In particular, ${}_AS_{nB}$ is small.

The following technical lemma, which is due to Sims [9], gives a criterion for when $\pi << \pi B$.

3.3. <u>Lemma</u>. Let B be a subgroup of S_n. Let H_i be the subgroup of B fixing elementwise $[1,i-1]$, and let O_i be the orbit of i with respect to H_i, i.e., $O_i = \{ \tau(i) \mid \tau \in H_i \}$. Then for any $\pi \in S_n$, $\pi << \pi B$ if and only if $\pi(i) \leq \pi(x)$ for each $x \in O_i$, $i = 1,2,\ldots,n$.

<u>Proof</u>. For any $1 \leq i \leq n$ and any $x \in O_i$, there is a $\beta_{i,x} \in B$ such that $\beta_{i,x}(j) = j$ for $1 \leq j < i$ and $\beta_{i,x}(i) = x$. Assume that $\pi << \pi B$. Then $\pi << \pi\beta_{i,x}$, and since $\pi(j) = \pi\beta_{i,x}(j)$ for $1 \leq j < i$, we must have $\pi(i) \leq \pi\beta_{i,x}(i) = \pi(x)$. Conversely, assume that $\pi(i) \leq \pi(x)$ for every $x \in O_i$. For any $\beta \in B$, if $\pi \neq \pi\beta$, let i_β be the least argument for which π and $\pi\beta$ differ, i.e., $\pi(j) = \pi\beta(j)$ for $1 \leq j < i_\beta$ and $\pi(i_\beta) \neq \pi\beta(i_\beta)$. Since π is bijective, $\beta(j) = j$ for $1 \leq j < i_\beta$. Hence $\beta \in H_{i_\beta}$ and $\beta(i_\beta) \in O_{i_\beta}$. Thus $\pi(i_\beta) < \pi\beta(i_\beta)$ and $\pi << \pi\beta$. ||

The subgroups H_i in this lemma form a descending sequence $B = H_1 \supset H_2 \supset \cdots \supset H_n = \{\iota\}$ where ι denotes the identity element of S_n. Thus if k is the least index such that $H_k = \{\iota\}$, then $H_j = \{\iota\}$ and $O_j = \{j\}$ for $k \leq j \leq n$. Hence in applying lemma 3.3, we need only check those indices i with $i < k$. For example, if B

is transitive, i.e., if $O_1 = [1,n]$, and if $H_j = \{i\}$ for $j \geq 2$, then $\pi << \pi B$ if and only if $\pi(1) = 1$.

Let A be a label subgroup of S_n, say A is the subgroup corresponding to the partition $n_1 + \ldots + n_k = n$. We claim that the set of all $\pi \in S_n$ satisfying $\pi << \pi A$ can be constructed as follows:

1. Form all the distinct ordered partitions $P_i = \{P_{i1}, \ldots, P_{ik}\}$ of $[1,n]$ into k subsets P_{ij} satisfying $|P_{ij}| = n_j$. There are $c = n!/n_1!\ldots n_k!$ *such* partitions.

2. For each P_i and for each $P_{ij} \in P_i$ list the elements of P_{ij} in their natural order, say $h_{ij1} < h_{ij2} < \ldots < h_{ijn_j}$.

3. For $i = 1, \ldots, c$, define π_i by $\pi_i(h_{ijs}) = \sum_{r=1}^{j-1} n_r + s$.

Each P_i is a partition of $[1,n]$, and the integral intervals $I_j = [\sum_{r=1}^{j-1} n_r + 1, \sum_{r=1}^{j} n_r]$, $j = 1, \ldots, k$, also partition $[1,n]$. Thus, since $|P_{ij}| = |I_j|$, $1 \leq j \leq k$, the π_i are distinct, well-defined elements of S_n.

3.4. Lemma. $\{ \pi_i \mid 1 \leq i \leq c \} = \{ \pi \in S_n \mid \pi << A\pi \}$.

Proof. For $\alpha \in A$, assume that $\pi_i \neq \alpha\pi_i$. Let t be the least integer in $[1,n]$ for which $\pi_i(t) \neq \alpha\pi_i(t)$. Say $t \in P_{ij}$ and $t = h_{ijs}$ for some $1 \leq j \leq k$ and $1 \leq s \leq n_j$. Now $\pi_i(t) = \sum_{r=1}^{j-1} n_r + s \in I_j$. Since A is a label subgroup, $\{ \alpha(m) \mid m \in I_j \} = I_j$. Also, by the choice of t, $\pi_i(h_{ijp}) = \alpha\pi_i(h_{ijp})$ for $1 \leq p < s$. Thus, since $\pi_i(t) \neq \alpha\pi_i(t)$, we must have $\alpha\pi_i(t) = \alpha(\sum_{r=1}^{j-1} n_r + s) > \sum_{r=1}^{j-1} n_r + s = \pi(t)$. Hence,

$\pi_i << \alpha\pi_i$, and $\pi_i << A\pi_i$. By the above, $\{ \pi_i \mid 1 \leq i \leq c \} \subset \{ \pi \in S_n \mid \pi << A\pi \}$. Since the latter set forms the canonical set of right coset representatives of A in S_n, by LaGrange's theorem $|\{\pi \in S_n \mid \pi << A\pi \}| = |S_n|/|A| = c$. Hence $\{ \pi_i \mid 1 \leq i \leq c \} = \{ \pi \in S_n \mid \pi << A\pi \}$. ||

3.5. <u>Corollary.</u> The set ${}_AS_n = \{ \pi \in S_n \mid \pi << A\pi \}$ can be naturally identified with the set D of all integral n-strings containing n_k, 0-digits; n_{k-1}, 1-digits; ... ; n_1, (k-1)-digits. More explicitly, define $\tau : [1,n] \to [0,k-1]$ by $\tau(s) = k-j$ where $s \in I_j$. Then the map $\psi : {}_AS_n \to D$ given by $\psi(\pi) = (\tau\pi(1), \ldots, \tau\pi(n))$ is a bijection.

<u>Proof</u>. For π_1 and π_2 in ${}_AS_n$, let $H_{ij} = \{ h \in [1,n] \mid \pi_i(h) \in I_t \}$, $i = 1,2$; $j = 1, \ldots, k$. Now $\psi(\pi_1) = \psi(\pi_2)$ if and only if $H_{1j} = H_{2j}$, $1 \leq j \leq k$. Linearly order the sets H_{1j}, say $h_{j1} < h_{j2} < \ldots < h_{js_j}$, $1 \leq j \leq k$. Then, since π_1 and π_2 are in ${}_AS_n$, by the proof of lemma 3.4, $H_{1j} = H_{2j}$ implies that $\pi_1(h_{js}) = \sum_{i=1}^{j-1} n_i + s = \pi_2(h_{js})$. thus , $\psi(\pi_1) = \psi(\pi_2)$ implies that $\pi_1 = \pi_2$, and ψ is injective. Since $|{}_AS_n| = n!/n_1!\ldots n_k! = |D|$, ψ is bijective.

In the special case where k = 2, i.e., A is the label subgroup of S_n corresponding to a partition of n of the form m + (n-m) = n, the identified set of canonical right coset representatives takes a particularly simple form. Namely, it is the set D_m^n of all n-bit binary strings with m, 1-bits and (n-m), 0-bits. Moreover, the natural ordering of the elements of D_m^n considered as binary integers agrees inversely with the ordering << on S_n. Explicitly, if for α in D_m^n we denote by $\bar{\alpha}$ the permutation in S_n associated with α, i.e., $\alpha = \psi(\bar{\alpha})$ where ψ is the bijective map of corollary 3.5, then:

3.6. Lemma. For any α and β in D^n_m, $\alpha \geq \beta$ if and only if $\overline{\alpha} << \overline{\beta}$.

Proof. Let $\alpha = (a_1a_2 \ldots a_n)$ and $\beta = (b_1b_2 \ldots b_n)$. Assume that $\alpha > \beta$. Let i be the least index such that $a_i \neq b_i$. Then we must have $a_i = 1$ and $b_i = 0$. Hence, by the definition of $\overline{\alpha}$ and $\overline{\beta}$, $\overline{\alpha}(j) = \overline{\beta}(j)$ for $1 \leq j < i$, and $\overline{\alpha}(i) \leq m < \overline{\beta}(i)$. Thus $\overline{\alpha} << \overline{\beta}$. Conversely, if $\overline{\alpha} << \overline{\beta}, \overline{\alpha} \neq \overline{\beta}$, the converse argument yields that $\alpha > \beta$. ||

Let C be the collection of all linearly ordered m-element subsets of [1,n], i.e., C is the collection of all linearly ordered combinations of the elements of [1,n] taken m at a time. Any α in D^n_m uniquely determines an element $\omega(\alpha) : 1 \leq a_1 < a_2 < \ldots < a_m \leq n$ of C where the a_i-th digit (from the left) of α is 1. ω is a bijective map from D^n_m to C, and we have:

3.7. Lemma. For any α and β in D^n_m, $\alpha \geq \beta$ (as binary integers) if and only if $\omega(\alpha) \leq \omega(\beta)$ (lexicographically).

Proof. Let $\omega(\alpha) : 1 \leq a_1 < a_2 < \ldots < a_m \leq n$, and $\omega(\beta) : 1 \leq b_1 < b_2 < \ldots < b_m \leq n$. Then, $\alpha > \beta$ if and only if there exists an index i, $1 \leq i \leq m$, such that $a_j = b_j$, $1 \leq j < i$, and $a_i > b_i$ if and only if $\omega(\alpha)$ is lexicographically less than $\omega(\beta)$. ||

We can combine the correspondence between D^n_m, ${}_AS_n$ and m-element combinations with lemma 3.3 to give a method for describing the canonical right coset representatives of A in S_n which are also canonical left coset representatives of B in S_n. Namely, if we let O_i, i = 1, . . . , n, be as in lemma 3.3, then:

3.8. Lemma. Let D be the set of all linearly ordered m-element subsets A of [1,n] satisfying $O_i \cap A = \emptyset$ if $i \notin A$. Then there is a bijective map υ from $\mathcal{D}$ to the subset R of all $\alpha \in D^n_m$ satisfying $\bar{\alpha} \ll \overline{\alpha B}$. Explicitly, for $A: 1 \leq a_1 < a_2 < \ldots < a_m \leq n$, in $\mathcal{D}$, $\upsilon(A) = (e_1 e_2 \ldots e_n)$ where $e_j = 1$ if $j \in A$ and 0 otherwise.

Proof. Let $[1,n]/A = \{ b_1 < b_2 < \ldots < b_{n-m} \}$, and let $\overline{\upsilon(A)} = \gamma$. Then $\gamma(j) = \begin{cases} t & , j = a_t \\ m+t, & j = b_t \end{cases}$. Choose any $j \in [1,n]$ and $x \in \mathcal{O}_j$.

If $j = a_t$, then $x \geq a_t$ and $\gamma(x) \geq t$. Hence $\gamma(j) = t \leq \gamma(x)$.

If $j = b_t$, then $j \notin A$ and, by hypothesis, $x \notin A$. Thus $x = b_s$ for some $s \geq t$, and $\gamma(j) = m + t \leq m + s = \gamma(x)$. Therefore, $\gamma(j) \leq \gamma(x)$ for any $x \in \mathcal{O}_j$, and by lemma 3.3, $\gamma = \overline{\upsilon(A)} \ll \overline{\upsilon(A)B}$. Hence υ is a map from $\mathcal{D}$ to R. The converse argument shows that υ is surjective. Clearly, υ is injective, and thus υ is bijective.||

Note that in the special case when B is transitive, $\mathcal{O}_1 = [1,n]$, and hence for any allowable m-element subset A in $\mathcal{D}$, $A \cap \mathcal{O}_1 = A \neq \emptyset$, and we must have $1 \in A$.

The results of this section admit a straightforward generalization. For any subset X of [1,n], say $X = \{ x_1, \ldots, x_m \}$, denote by S_X the full permutation group on X, i.e., the group of all bijective maps from X to X. The natural bijective map λ from [1,m] to X defined by $\lambda(i) = x_i$ induces the isomorphism τ from S_X to S_m by $\tau(\pi) = \lambda^{-1}\pi\lambda$. We call a subgroup A of S_X the label subgroup of S_X corresponding to the partition $m_1 + \ldots + m_k = m$ of m if and only if $\tau(A)$ is the label subgroup of S_m corresponding to this partition of m. Also, we take as the linear ordering on S_X the ordering induced via τ by the natural ordering of S_m, i.e., for α and β in S_X, $\alpha << \beta$ if and only if $\tau(\alpha) << \tau(\beta)$. This ordering is dependent on the indexing of X. With these definitions, all of the above results immediately generalize to S_X.

4. Basic recursive schemes. We see from section 3 that for computing double coset representatives on a binary machine, it would be advantageous to reduce the general double coset representative problem to the special case where the label subgroup corresponds to a partition of n of the form $m + (n-m) = n$. In terms of the graph, such a reduction is conceptually clear. For example, we can label an n-node graph G with n_1 labels L_1, n_2 labels L_2 and n_3 labels L_3, $n_1 + n_2 + n_3 = n$, as follows:

1. Determine all topologically distinct labelings of G with n_1 labels L_1 and $(n-n_1)$ blanks.
2. For each such labeling, determine all distinct labelings of the blank labeled nodes with n_2 labels L_2 and n_3 labels L_3.

The following procedure formalizes this concept and yields the desired recursive scheme:

Let X be a subset of $[1,n]$, say $X = \{ x_1, x_2, \ldots, x_m \}$, and let B be a subgroup of S_X. For any subset Y of X and any $\beta \in B$ such that $\beta(Y) = Y$, denote by $\beta|_Y$, β restricted to Y, and denote by $B|_Y$ the group $\{ \beta|_Y \mid \beta \in B \text{ with } \beta(Y) = Y \}$. Then, if A is the label subgroup of S_X corresponding to the partition $m_1 + m_2 + \ldots + m_k = m$ of m, $k > 2$, we claim that a double coset representative set R for A and B in S_X can be obtained as follows:

1. Determine a double coset representative set R_1 of A_1 and B in S_X where A_1 is the label subgroup of S_X corresponding to the partition $m_1 + (m-m_1) = m$.
2. Do for each α in R_1:
   a) Determine $N_\alpha = \alpha^{-1}(\{ x_{m_1+1}, \ldots x_m \})$ and $B|_{N_\alpha}$
      Index the elements of N_α, say $N_\alpha = \{y_1, \ldots, y_{m-m_1}\}$.
   b) Determine a double coset representative set R_α of A_α and $B|_{N_\alpha}$ in S_{N_α} where A_α is the label subgroup of S_{N_α} corresponding to the partition $m_2 + \ldots + m_k = m-m_1$.
3. Set $R = \bigcup_{\alpha \in R_1} \{ \gamma * \alpha \mid \gamma \in R \}$, where $\gamma*\alpha \in S_X$ is defined by

$$\gamma*\alpha(x) = \begin{cases} \alpha(x), & x \in N_\alpha \\ x_{m_1+j}, & x \in N_\alpha \text{ and } \gamma(x) = y_j. \end{cases}$$

4.1. Lemma. R is a double coset representative set for A and B in S_X.

Proof. Since $\alpha(X/N\ ) = \{ x_1, \ldots, x_{m_1} \}$ and $\{ x_{m_1+j} \mid y_j \in \gamma(N_\alpha) \}$ partition X, each $\gamma*\alpha$ is a well-defined element of S_X.

We will first show that R contatins a representative set. For any $\pi \in S_X$, since R_1 is a representative set for A_1 and B in S_X, there exist $\alpha \in R_1$, $\delta_1 \in A_1$ and $\beta_1 \in B$ such that $\delta_1\pi\beta_1 = \alpha$. Define $(\pi\beta_1)'$ by $(\pi\beta_1)'(x) = y_j$ where $\pi\beta(x) = x_{m_1+j}, x \in N_\alpha$. $(\pi\beta_1)'(N_\alpha) = N_\alpha$, and $(\pi\beta_1)'$ is in S_N. Since R_α is a representative set for A and $B|_{N_\alpha}$ in S_{N_α}, there exist $\gamma \in R_\alpha$, $\delta_2 \in A_\alpha$ and $\beta_2 \in B|_{N_\alpha}$ such that $\delta_2(\pi\beta_1)'\beta_2 = \gamma$. Choose $\beta \in B$, $\beta(N_\alpha) = N_\alpha$, satisfying $\beta|_{N_\alpha} = \beta_2$. Define δ by

$$\delta(x) = \begin{cases} x_{m_1+s}, & x = x_{m_1+t} \text{ and } \delta_2(y_t) = y_s \\ \alpha\beta^{-1}\alpha^{-1}\delta_1(x), & x \in \{x_1, \ldots, x_{m_1}\} \end{cases}$$

A direct computation shows that $\delta \in A$ and $\delta\pi(\beta_1\beta) = \gamma*\alpha$. Hence R contains a representative set.

Now, assume that for some $\gamma_1*\alpha_1$ and $\gamma_2*\alpha_2$ in R there exist $\delta \in A$ and $\beta \in B$ such that $\delta\gamma_1*\alpha_1\beta = \gamma_2*\alpha_2$. Then,

$$\begin{aligned} \gamma_2*\alpha_2(X/N_\alpha) &= \alpha_2(X/N_\alpha) \\ &= \{x_1, \ldots, x_{m_1}\} \\ &= \delta(\gamma_1*\alpha_1)\beta(X/N_\alpha) \\ &= \alpha_1\beta(X/N_\alpha) \, . \end{aligned}$$

Thus, $(\alpha_1\beta)\alpha_2^{-1}(\{x_1, \ldots, x_{m_1}\}) = \{x_1, \ldots, x_{m_1}\}$, and $\alpha_1\beta$ and α_2 differ only by an element of A_1. Since R_1 is a representative set, $\alpha_1 = \alpha_2$. From $\delta\gamma_1*\alpha_1\beta = \gamma_2*\alpha_1$, we have that for $x \in N_\alpha$, $\gamma_2*\alpha_1(x) = x_{m_1+j} = \delta\gamma_1*\alpha\beta_1(x)$, where $\gamma_2(x) = y_j$. Therefore $\beta(x) \in N_\alpha$, and $y_j = \delta\gamma_1\beta(x) = \gamma_2(x)$. Hence, $\delta|_{N_{\alpha_1}}\gamma_1\beta|_{N_{\alpha_1}} = \gamma_2$.

Since R_α is a representative set, $\gamma_1 = \gamma_2$ and $\gamma_1*\alpha_1 = \gamma_2*\alpha_2$. Thus the members of R determine distinct double cosets, and R is a representative set for the double cosets of A and B in S_X.||

Let B be a subgroup of S_X, $|X| = n$, and let A be a label subgroup of S_X. The computation of a representative set of the double cosets of A and B in S_X admits a further recursive reduction based on the orbits of B. By lemma 4.1, we can assume for this recursive scheme that A corresponds to the partition $m+(n-m)=n$.

Conceptually, the reduction scheme works as follows:

1. Choose a fixed node x of the graph G, and let N be the image nodes of x under the symmetry group B of G.

2. Do for i, $\max(0, |N|+m-n) \leq i \leq \min(|N|, m)$:

   i. Determine all distinct (with respect to B) labelings of N with i labels of the first type and $|N|-i$ labels of the second type.

   ii. For each such labeling of N, let U be the subgroup of B which preserves that labeling of N, and determine all distinct (with respect to U) labelings of the remaining nodes of G with $(m-i)$ labels of the first type and $(n-|N|-m+i)$ labels of the second type.

   iii. Compose each labeling of N and its associated labelings of G/N.

Formally we have:

Let $X = \{x_1, \ldots, x_n\}$, and let O be an orbit of B, i.e., $O = \{\pi(x_t) \mid \pi \in B\}$ for some fixed $x_t \in X$. Then a representative set R of the double cosets of A and B in S_X can be obtained as follows:

1. Index the elements of O and $X/O = \overline{O}$, say $\{y_1, \ldots, y_k\}$ and $\{w_1, \ldots, w_{n-k}\}$, respectively. Since O is an orbit, $\beta(O) = O$ and $\beta(\overline{O}) = \overline{O}$ for any $\beta \in B$.

2. Do for $i = \max(0, m+k-n), \ldots, \min(k, m)$:

   i. Determine a double coset representative set T_i of A_i and $B|_O$ in S_O where A_i is the label subgroup of S_O corresponding to the partition $i + (k-i) = k$.

ii. Do for each $\alpha \in T_i$:

a) Form $N_\alpha = \alpha^{-1}(\{y_1, \ldots, y_i\})$ and $B^\alpha = \{\pi \in B \mid \pi(N_\alpha) = N_\alpha\}$.

b) Determine a double coset representative set H_α of $A_{\bar{i}}$ and $B^\alpha|_{\bar{O}}$ in $S_{\bar{O}}$ where $A_{\bar{i}}$ is the label subgroup corresponding to the partition $(m-i) + (n-k-m+i) = n-k$.

c) Form $R_\alpha = \{\gamma \circ \alpha \mid \gamma \in H_\alpha\}$ where

$$\gamma\circ\alpha(x) \begin{cases} x_t, & x \in N_\alpha,\ \alpha(x) = y_t \\ x_{m-i+t}, & x \in O/N_\alpha,\ \alpha(x) = y_t \\ x_{i+t}, & x \in \bar{O}, \gamma(x) = w_t,\ t \leq m-i \\ x_{k+t}, & x \in \bar{O},\ \gamma(x) = w_t,\ t > m-i. \end{cases}$$

3. Set $R = \bigcup_i \bigcup_{\alpha \in T_i} R_\alpha$, $\max(0, m+k-n) \leq i \leq \min(k,m)$.

4.2. Lemma. R is a double coset representative set for A and B in S_X.

Proof. For any $\pi \in S_X$, let $N_1 = \{x \in O \mid \pi(x) = x_t,\ t \leq m\}$, say $N_1 = \{y_{t_1}, \ldots, y_{t_i}\}$ and $O/N_1 = \{y_{s_1}, \ldots, y_{s_{k-i}}\}$. Define $\pi_1 \in S_O$ by

$$\pi_1(y) = \begin{cases} y_j, & y \in N_1,\ y = y_{t_j} \\ y_{i+j}, & y \notin N_1,\ y = y_{s_j} \end{cases} .$$

Since T_i is a representative set, there exist $\alpha \in T_i$, $\delta_1 \in A_i$ and $\beta_1 \in B_O$ such that $\delta_1\pi_1\beta_1 = \alpha$. Choose $\beta \in B$ satisfying $\beta|_O = \beta_1$. Let $N_2 = \{x \in \bar{O} \mid \pi\beta(x) = x_t,\ t \leq m\}$, say $N_2 = \{w_{t_1}, \ldots, w_{t_{m-i}}\}$ and $\bar{O}/N_2 = \{w_{s_1}, \ldots, w_{s_{n-k-mti}}\}$. Define $\pi_2 \in S_{\bar{O}}$ by

$$\pi_2(w) = \begin{cases} w_j, & w \in N_2,\ w = w_{t_j} \\ w_{m-i+j}, & w \notin N_2,\ w = w_{s_j} \end{cases} .$$

Since H_α is a representative set, there exist $\gamma \in H_\alpha$, $\delta_2 \in A_{\bar{i}}$ and $\beta_2 \in B^\alpha|_{\bar{\mathcal{O}}}$ such that $\delta_2\pi_2\beta_2 = \gamma$. Choose $\beta' \in B$ satisfying $\beta'|_{\bar{\mathcal{O}}} = \beta_2$, and let $N_\gamma = \gamma^{-1}(\{w_1, \ldots, w_{m-i}\})$.

$$\begin{aligned} \beta(N_\alpha) &= \beta_1(N_\alpha) \\ &= \pi_1^{-1}\delta_1^{-1}\alpha(N_\alpha) \\ &= \pi_1^{-1}(\{y_1, \ldots, y_i\}) \\ &= N_1. \end{aligned}$$

Similarly, $\beta' N_\gamma = N_2$. Thus,

$$\begin{aligned} \pi\beta\beta'(N_\alpha \cup N_\gamma) &= \pi N_1 \cup \pi\beta N_2 \\ &= \{x \in X \mid \pi(x) = x_t,\ t \leq m\} \\ &= \gamma o \alpha(N_\alpha \cup N_\gamma). \end{aligned}$$

Hence, $\pi\beta\beta'$ and $\gamma o\alpha$ differ only by an element in A and $A\pi B = A\gamma o\alpha B$.

Assume that there exist $\gamma_1 o\alpha_1$ and $\gamma_2 o\alpha_2$ in R, $\delta \in A$ and $\beta \in B$ such that $\delta\gamma_1 o\alpha_1\beta = \gamma_2 o\alpha_2$. Then,

$$\begin{aligned} \gamma_2 o\alpha_2(N_{\alpha_2}) &= \delta\gamma_1 o\alpha_1\beta(N_2) \\ &= \{x_1, \ldots, x_i\}, \end{aligned}$$

for some $0 \leq i \leq m$. Thus $\beta N_{\alpha_2} \subset N_{\alpha_1}$. Symmetrically, $\beta^{-1}N_{\alpha_1} \subset N_{\alpha_2}$, and hence $\beta N_{\alpha_2} = N_{\alpha_1}$. This implies that α_1 and α_2 are both in the same $_1T_i$, and $\alpha_1\beta|_0$ and a_2 differ only by an element of A_i. Hence $\alpha_1 = \alpha_2$ and $\beta \in B^{\alpha_1}$. A similar argument using $N_{\gamma_i} = \{ w \in \bar{\mathcal{O}} \mid \gamma_i(w) = w_t,\ t \leq m-i \}$, $i = 1,2$, shows that $\gamma_1 = \gamma_2$.

Thus the elements of R determine distinct double cosets, and R is a representative set for A and B in S_X.||

Since the only property of $\mathcal{O}$ used in the above proof is that $\beta(\mathcal{O}) = \mathcal{O}$ for all $\beta \in B$, we have:

4.3. Corollary. Lemma 4.2 is valid if $\mathcal{O}$ is a union of orbits of B.

As we have seen in section 3, we can always choose a double coset representative set R for A and B in S_X, $|X| = n$, such that $R \subset {}_AS_X$, the canonical representative set for the right cosets of A in S_X, i.e., we can always choose a small double coset representative set. Moreover, by corollary 3.5, such a small representative set can be identified with a set of certain integral n-strings. We will assume, henceforth, that such an identification has been made. In particular, in the special case where A is a label subgroup corresponding to a partition of the form $m + (n-m) = n$, any small double coset representative set is a set of n-bit binary strings with m, 1-bits and $(n-m)$, 0-bits. If α is such a binary string, we will denote by α the associated permutation in S_X.

In many cases the following lemma when applied to the T_i in step 2(i) of lemma 4.2 reduces considerably the number of steps in the process:

4.4. Lemma. Let T be a small representative set for the double cosets of A and B in S_X, $|X| = n$. Say A is the label subgroup of S_X corresponding to the partition $k + (n-k) = n$. Let $\hat{A}$ be the label subgroup of S_X corresponding to the partition $(n-k) + k = n$. Then a small representative set $\hat{T}$ for the double cosets of $\hat{A}$ and B in S_X can be obtained by simply forming the binary complements $\hat{\alpha}$ of each α in T, i.e., $T = \{\, \hat{\alpha} = (2^n-1) - \alpha \mid \alpha \in T \,\}$.

Proof. Define $\delta \varepsilon S_X$ by $\delta(x_i) = x_{n+1-i}$. Note that $\delta = \delta^{-1}$. For any small representative α, let $\bar{\alpha}$ be the corresponding permutation in S_X. We will first show that $T_\delta = \{ \delta\bar{\alpha} \mid a \varepsilon T \}$ is a representative set for the double cosets of $\hat{A}$ and B in S_X.

For any $\pi \varepsilon S_X$, $\delta\pi$ is also in S_X. Since T is a representative set, there exist $\alpha \varepsilon T$, $\gamma \varepsilon A$ and $\beta \varepsilon B$ such that $\delta\pi = \gamma\bar{\alpha}\beta$. Thus $\delta^2\pi = \pi = \delta\gamma\alpha\beta = \delta\gamma\delta(\delta\alpha)\beta$. Since $\gamma \varepsilon A$,
$\delta\gamma\delta(\{x_1, \ldots, x_{n-k}\}) = \delta\gamma(\{x_{k+1}, \ldots. x_n\}) = \delta(\{x_{k+1}, \ldots, x_n\})$
$= \{x_1, \ldots, x_{n-k}\}$. Hence $\delta\gamma\delta \varepsilon \hat{A}$ and π is in the double coset determined by $\delta\bar{\alpha}$. Now assume that for some α_1 and α_2 in T, $\gamma\delta\bar{\alpha}_1\beta = \delta\bar{\alpha}_2$ for some $\gamma \varepsilon \hat{A}$ and $\beta \varepsilon B$. Then,
$\delta\gamma\delta\bar{\alpha}_1\beta = \delta^2\bar{\alpha}_2 = \bar{\alpha}_2$. As above, $\delta\gamma\delta \varepsilon A$, and, since T is a representative set, $a_1 = \alpha_2$ and $\delta\bar{\alpha}_1 = \delta\bar{\alpha}_2$. Hence we have that T_δ is a representative set.

We will now show that for any $\alpha \varepsilon T$, $\bar{\hat{\alpha}} \varepsilon \hat{A}\delta\bar{\alpha}B$. By the definition of $\hat{\alpha}$, $\bar{\hat{\alpha}}(\bar{\alpha}^{-1}(x_i)) = x_{n-k+i}$ for $1 \leq i \leq k$, and
$\bar{\hat{\alpha}}(\bar{\alpha}^{-1}(x_{k+i})) = x_i$ for $1 \leq i \leq n-k$. Therefore, for $1 \leq i \leq n-k$,
$\overline{\hat{\alpha}\alpha}^{-1}\delta(x_i) = \overline{\hat{\alpha}\alpha}^{-1}(x_{n+1\ i}) = x_{n-k+1-i}$, i.e.,
$\overline{\hat{\alpha}\alpha}^{-1}\delta(x_i) \varepsilon \{x_1, \ldots, x_{n\ k}\}$, and $\overline{\hat{\alpha}\alpha}^{-1}\delta \varepsilon \hat{A}$. Thus $\bar{\hat{\alpha}} \varepsilon \hat{A}\delta\bar{\alpha}B$, and $\hat{T}$ is a small representative set for the double cosets of A and B in S_X.||

Using the results of this section, we now can describe the two algorithms.

5. Double coset algorithms. The analysis done in the previous sections yields two efficient computer implementable algorithms for determining a small double coset representative set of A and B in S_X $(X \subset [1,n]$, $|X| = k)$ where A is the label subgroup of S_X corresponding to the partition $m_1+m_2+\ldots+m_t = k$ of k.

As is often the case, the form of the data structures in the machine implementations of the algorithms determines the form of the aigorithms, and conversely. In the implementations, any subset X of [1,n] is represented by the binary n-string U where the i-th bit (from the left) of U is 1 if and only if $i \in X$. Thus, there is no distinction between subsets and their associated binary strings, and the elements of a subset are implicitly indexed, Each such string U is carried right justified in a machine word.

Any element i of [1,n] when considered as an element in the domain of S_n is represented as the machine word 2^{n-i}, and a small right coset representative.is represented as an&vector in the form given by corollary 3.5 if $t > 2$ and as a binary word if $t=2$. For example, if A is the label subgroup of S_7 corresponding to the partition $2 + 2 + 3 = 7$ (respectively, $3 + 4 = 7$) and

$$A = \begin{pmatrix} 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 2^5 & 2^3 & 2^0 & 2^6 & 2^1 & 2^4 & 2^2 \end{pmatrix} \in S_7,$$ then the small double ooset

representative of A is (2, 1, 0, 2, 0, 1, 0), (respectively, (1 0 0 1 0 1 0)).

This compact representation of subsets and coset representatives is needed in practice since for even relatively small values of n, the number of distinct double cosets can be very large. This latter number is optionally computed in advance via the generalized Pólya enumeration formula, and it is used to help decide if the desired construction is even feasible in-terms of time and core store.

A permutation π in a symmetry group B contained in S_n is represented in the implementations in two ways. It isrepresented as the n-vector of the images, $c(\pi)=(\pi(2^{n-1})=2^{n-\pi(1)},\ldots,\pi(2^1)=2^{n-\pi(n)})$ and also as a list $\bar{P}(\pi)$ where the members of $P(\pi)$ are the sets of elements in the non-trivial cycles of π. For example,

$$\pi = \begin{pmatrix} 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 2^5 & 2^1 & 2^3 & 2^0 & 2^7 & 2^2 & 2^6 & 2^4 \end{pmatrix} \text{ is carried as } c(\pi) =$$

$(2^5, 2^1, 2^3, 2^0, 2^7, 2^2, 2^6, 2^4)$ and as $P(\pi)=\{(10101000), (01000010)$, $(00010001)\}$. For many of the necessary computations, the second representation is the most efficient. However, the first representation is also needed since $P(\pi)$ does not uniquely determine π.

These representations permit most of the computations to be performed as logical hardware operations,, For example, if A corresponds to the partition $m+(n-m)=n$, e is a small right coset representative, $\pi \in B$, and U is a subset of $[1,n]$, then $\{j \in U \mid \text{the } j\text{-th digit of } e \text{ is } 1\}$ is represented by $U \wedge e$, and $\pi(U) = U$ if and only if $p \wedge U = p$ or 0 for all $p \in P(\pi)$.

We will describe the algorithms using these representations, 5.1. Algorithm I. This algorithm is recursive both in the number of terms in the partition of k and in the orbits of B. The algorithm is presented as three nested subalgorithms,

Subalgorithm Ic. The deepest level subalgorithm.

Purpose. To determine the canonical set of double coset representatives of A and B in S_X in the special case where A corresponds to the partition $m + (k-m) = k$ of $k = |X|$, and B is transitive, i.e., B has only one orbit.

Technique. The subalgorithm is based on corollary 3.5 and lemmas 3.3, 3.4, 3.6 and 3.8. It first generates the small subset $P_1 = \{\pi \in {}_AS_X \mid \bar{\pi} \ll \overline{\pi B}\}$, $P_1 \subset D_m^k$, i.e., the subset of canonical right coset representatives which are also canonical left coset representatives. It then eliminates from P_1 any elements π not satisfying $\bar{\pi} \ll A\overline{\pi B}$.

Input. The binary n-string U corresponding to X, $k = |X|$, m, and a list which is the n-vector form of a set C of permutations in S_n such that $\gamma(X) = X$ for every $\gamma \in C$ and $C|_X = B$.

Output. A list R_0 of binary n-strings e, $e \wedge U = e$, which corresponds to the canonical set of double coset representatives of A and B in S_X.

Ordered lists: R_0, R_1, D_0, D_1.

START

*[Determine the elements of X].

1. Determine $s(i) \in [1,n]$, $1 \le i \le k$, such that $s(i) \wedge U \ne 0$ and

$s(i) > s(j)$ if $i < j$.

* [The following handles the special cases where there must be only one double coset].

2, If $m=0$, $R_0 \leftarrow 0$; else if $m=1$, $R_0 \leftarrow s(1)$; else if $m=k-1$, $R_0 \leftarrow U - s(k)$, else if $m=k$, $R_0 \leftarrow U$; else go to 4.

3. RETURN.

* [Generate the orbits O of lemma 3.3].

4. Initialize: $N \leftarrow C$.

5. Do 7, $i=2, \ldots, k$.

6. $N \leftarrow \{ \pi \in N \mid \pi(s(i-1)) = s(i-1) \}$.

7. $O(i) \leftarrow \bigvee_{\pi \in N} \pi(s(i))$.

* [Generate all allowable m-element subsets as per lemma 3.8].

8. Initialize: $R_1 \leftarrow s(1)$ $D_1 \leftarrow 0$, $R_0 \leftarrow \emptyset$, $D_0 \leftarrow \emptyset$.

9. Do 16, $t=1, \ldots, m-1$.

10. Do 15 for each W in R_1 using its corresponding D in D_1.

11. Determine $\max \{ d \mid s(d) \wedge W \neq 0)$.

12. Do 14, $i = d+1, \ldots, (k-m+1)+t$.

13. If $D \wedge O(i) = 0$, put $W \vee s(i)$ on R_0 and D on D_0.

14. $D \leftarrow D \vee O(i)$.

15. Continue,

16. $R_1 \leftarrow R_0$, $D_1 \leftarrow D_0$, $R_0 \leftarrow \emptyset$, $D_0 \leftarrow \emptyset$.

* [Eliminate redundant representatives].

17. Do 22 for $e \in R_1$.

18. Do 21 for $\pi \in C / \{identity)$,

19. Do 20, $i=1, \ldots, k$.

20. If $\pi(s(i)) \wedge e \neq 0$ and $s(i) \wedge e = 0$, go to 17;
else if $\pi(s(i)) \wedge e = 0$ and $s(i) \wedge e \neq 0$, go to 18.

21. Continue.

22. Put e on R_0,

23. RETURN.

END

Subalgorithm Ib. The intermediate level subalgorithm

Purpose. To determine a small set of double coset representatives of A and B in S_X in the special case where A corresponds to a partition of k of the form $m+(k-m) = k$ and B is any subgroup of S_X.

Technique. This subalgorithm is recursive and is based on lemma 4.2, i.e., on recursion on orbits. It uses subalgorithm Ic.

Input. The binary n-string U corresponding to X, $k=|X|$, m, and two lists which contain the n-vector form and the cycle set form, respectively, of a set C of permutations in S_n such that $\gamma(X) = X$ for every $\gamma \in C$ and $C|_X = B$.

output. A list R of binary n-strings e, $e \wedge U = e$, which corresponds to a small double coset representative set of A and B in S_X.

Ordered lists: R, R(h,i), V(h,j).

START

1. Initialize: $U(1) \leftarrow U$, $C(1) \leftarrow C$, $k(1) \leftarrow k$, $m(1) \leftarrow m$, $h \leftarrow 1$.

* [The following is the reduction part of the recursion].

2. $s \leftarrow \max \{2^d \mid 2^d \wedge U(h) \neq 0\}$.

3. $\mathrm{Obt}(h) \leftarrow \bigvee_{\pi \varepsilon C(h)} \pi(s)$.

4. $t(h) \leftarrow$ 1-bit count of $\mathrm{Obt}(h)$.

5. $i(h) \leftarrow \max\{0, m(h) + t(h) - k(h))$, $u(h) \leftarrow \min\{t(h), m(h).)$,

   $i_1 \leftarrow \max\{ i(h), t(h) - u(h)\}$, $u_1 \leftarrow \min\{u(h), t(h) - i(h)\}$.

6. Do 8, $i \varepsilon H = [i_1, \min\{u_1, \lceil t(h)/2 \rceil -1\}]$.

7. Call subalgorithm Ic with input $\mathrm{Obt}(h)$, $t(h)$, i, $C(h)$; getting as output $R(h,i)$.

8. $R(h, t(h) - i) \leftarrow \{\mathrm{Obt}(h) - e \mid e \,\varepsilon R(h,i)\}$.

9. Do 10 for $i \,\varepsilon\, [i(h), u(h)]$. $(H \cup \{t(h) - j \mid j \,\varepsilon\, H\})$.

10. Call subalgorithm Ic with input $\mathrm{Obt}(h)$, $t(h)$, i, $C(h)$; getting as output $R(h,i)$.

11. If $t(h) = k(h)$, go to 17.

12. Remove the first element $e(h)$ from $R(h,i(h))$.

13. $C(h + 1) \leftarrow \{ \pi \varepsilon\, C(h) \mid p \wedge e(h) = p$ or 0 for all $p \,\varepsilon\, P(\pi)\}$.

14. $U(h + 1) \leftarrow U(h) - \mathrm{Obt}(h)$, $m(h+1) \leftarrow m(h) - i(h)$, $k(h + 1) \leftarrow k(h)-t(h)$.

15. $h \leftarrow h + 1$.

16. Go to 2.

* [The following is the expansion part of the recursion].

17. If $h=1$, $R \leftarrow R(1, i(1))$ and RETURN.

18. $h \leftarrow h-1$.

19. Put the elements of $\{ f \vee e(h) \mid f\ R(h+1, i(h+1))\}$ on $V(h,m(h))$.

20. If $R(h, i(h)) = \emptyset$, $i(h) \leftarrow i(h) + 1$; else go to 12,

21. If $i(h) \leq u(h)$, go to 12.

22. If $h=1$, $R \leftarrow V(1,m)$ and RETURN.

23. h $\leftarrow$ h-l.

24. Put the elements of $\{f\forall e(h) \mid f \epsilon V(h+1, m(h+1)\}$ on V(h, m(h)).,

25. Go to 20.

END

Subalgorithm Ia. The highest level subalgorithm.

Purpose. To determine a small set of double coset representatives of A and B in S_n where A is the label subgroup of S_n corresponding to the partition $n_1 + n_2 + \ldots + n_q = n$ and B is any subgroup of S_n.

Technique. The main loop of the subalgorithm is based on lemma 4.1, i.e., on induction on the number of terms in the partition of n. The subalgorithm uses subalgorithms Ib and Ic.

Input. $q, n_1, \ldots, n_q$, and two lists which contain the n-vector form and cycle set form, respectively, of B.

Output. A list R of integral n-vectors if $q > 2$ or binary n-strings if $q \leq 2$ which corresponds (as in corollary 3.5) to a small double coset representative set for A and B in S_n; and a list P of subgroups of B where if e is the i-th element of R, then the i-th element of P is $\bar{e}^{-1} A \bar{e} \cap B$.[6]

Ordered lists: R, R_1, P, P_1, T, T_1.

START

* [Trivial partition case].

1. If q = 1: R $\leftarrow$ 0, P $\leftarrow$ B and STOP.

* [Initialization procedure].

2. Call subalgorithm Ib with input 2^n-1, n, $n-n_q$, B; getting as output T.

3. Do 4 for e εT.

4. P $\leftarrow$ B(e) = $\{\pi \varepsilon B \mid p \wedge e = p$ or 0 for every $p \varepsilon P(\pi)\}$.

5. If q = 2, R $\leftarrow$ T and STOP.

6. Do 8 for e εT.

7. Form w = (w(1), . . . , w(n)) where $w(j) = \begin{cases} 1, & 2^{n-j} \wedge e \# 0 \\ 0, & \text{otherwise} \end{cases}$.

8. Put w on R.

9. $n_0 \leftarrow n$.

* [Induction section].

10. Do 18, i=2, ... q -1.

11. Initialize: $n_0 \leftarrow n_0 - n_{q+2-i}$, $R_1 \leftarrow R$, $P_1 \leftarrow P$, $T_1 \leftarrow T$, $R \leftarrow \emptyset$, $P \leftarrow \emptyset$, $T \leftarrow \emptyset$.

12. Do for 17 each w = (w(1), ... w(n)) εR_1 and its corresponding e(w) εT_1 and B(w)ε P_1.

13. Call subalgorithm Ib with input e(w), n_0, $n_0 - n_{q+1-i}$, B(w) ; getting as output T.

14. Do 16 for f εT.

15. Form f*w = (v(1), ..., v(n)) where

$$v(j) = \begin{cases} i, & 2^{n-j} \wedge f \neq 0 \\ w(j), & \text{otherwise} \end{cases} .$$

16. Put f*w on R, put $B(f{*}w) = \{\pi \varepsilon B(w) \mid p \wedge f = p$ or 0 for every $p \varepsilon P(\pi)\}$ on P.

17. Continue,

18. Continue.

19. STOP.

'END

5.2. Algorithm II. This algorithm is a variant of the first algorithm. It uses recursion on the number of terms in the partition of n, i.e., it uses the technique of subalgorithm Ia. We will describe only that part of algorithm II which differs essentially from algorithm I.

Subalgorithm IIb.

Purpose. To determine a canonical set of double coset representatives of A and B in S_X, $X \subset [1,n]$, in the special case where A corresponds to a partition of $k = |X|$ of the form $m + (k-m) = k$ and B is any subgroup of S_X.

Technique. This subalgorithm is based directly on theorem 3.1. It also uses lemmas 3.3 and 3.6 and corollary 3.5. It systematically generates the binary n-strings contained in X with m l-bits. As each such string e is generated, the subalgorithm checks if e is on BL (bad list). If e is not on BL, it is put on GL (good list), and all other n-strings which correspond to small right coset representatives of A in S_X which belong to the double coset determined by e are computed. These latter n-strings are merged into BL. For each e in GL, the group $e^{-1}Ae \cap B$ is determined in the course of the computation and is saved on GLG.

Input. The binary n-string U corresponding to X, $k = |X|$, m , and two lists which contain the n-vector form and the cycle set form, respectively, of a set C of permutations in S_n such that

$\gamma(X) = X$ for every $\gamma \in C$ and $C|_X = B$.

output. A list GL of binary n-strings e, $e \wedge U = e$, which corresponds to the canonical set of double coset representatives of A and B in S_X , and for each e on GL the set $\{ \pi \in C \mid \pi|_X \in \bar{e}^{-1} A \bar{e} \cap B \}$ on the list GLG.

Ordered lists: GL, BL, GLG, OL, OB.

START .

1. Initialize: GL $\leftarrow \emptyset$, BL $\leftarrow \emptyset$, GLG $\leftarrow \emptyset$.

* [Trivial cases].

2. If m=0, GL $\leftarrow$ 0; else if m=k, GL $\leftarrow$ U; else go to 4.

3. GLG $\leftarrow$ C and RETURN.

* [Determine the elements of X].

4. Determine $s(i) \in [1,n]$, $1 \leq i \leq k$, such that $s(i) \wedge U \neq 0$ and $s(i) > s(j)$ if $i < j$.

* [Transfer out of main routine in special cases].

5. If m=1 or m=k-1, go to 29.

* [Main loop].

6. Initialize: $e \leftarrow \bigvee_{i=1}^{m} s(i)$; $t(i) \leftarrow m+1-i$, $1 \leq i \leq m$.

7. Put e on CL.

* [Determine $\bar{e}^{-1} A \bar{e} \cap B$].

8. $T \leftarrow \{\pi \in C \mid p \wedge e = p$ or 0 for every $p \in P(\pi)\}$,

9. Put T on GLG.

* [Compute the orbits O of lemma 3.3 for T in B].

10. Initialize: $N \leftarrow T / \{\text{identity}\}$ . . .

11. Do 13, i=1, ..., k-1.

12. $\quad O(i) \leftarrow \{\pi(s(i)) \mid \pi \in N\}$.

13. $\quad N \leftarrow \{\pi \in N \mid \pi(s(i)) = s(i)\}$.

* [Determine the left cosets of T in B using lemma 3.3, and via theorem 3.1 determine the right cosets contained in $A\bar{e}B$].

14. Do 20 for $\pi \in C / \{\text{identity}\}$.

15. $\quad$ Do 18, i=1, ... k-l.

16. $\quad\quad$ Do 17 for $s \in O(i)$.

17. $\quad\quad\quad$ If $n(s) > \pi(s(i))$, go to 14.

18. $\quad\quad$ Continue.

19. $\quad f \leftarrow \bigvee_{j=1}^{m} \pi(s(t(j)))$.

20. $\quad$ If $f \neq e$, merge f into BL (largest first).

* [Generate the next binary string].

21. Do 22, i=1, ...m.

22. $\quad$ If $t(i) < k-i$, go to 24.

23. RETURN.

24. $e \leftarrow e \wedge$ binary complement $(2 \cdot s(t(i)) - 1)$.

25. Do 27, j=1, • *es i.

26. $e \leftarrow e \vee s(t(i)+j)$.

27. $t(j) \leftarrow t(i) + (i+1-j)$.

28. If e is equal to the first member of BL, delete this member from BL and go to 21; else go to 7.

* [Special cases: Compute orbit representatives for C].

29. Initialize: $OL \leftarrow \emptyset$, $OB \leftarrow \emptyset$.

30. Do 35, $i=1, \ldots, k$.

31. If $OB \wedge s(i) \neq 0$, go to 35.

32. Put i on OL.

33. Do 34 for $\pi \in C$.

34. $OB \leftarrow OB \vee \pi(s(i))$.

35. Continue.

* [Special cases: Determine double coset representatives].

36. Do 38 for $i \in OL$.

37. Put s(i) on GL.

38. Put $\{ \pi \in C \mid \pi(s(i)) = s(i) \}$ on GLG.

39. If m=1, RETURN.

40. Replace each e on CL by its binary complement .

41. RETURN.

END

5.3. There are significant operational differences in the two algorithms. Algorithm I is computationally more complex than Algorithm II. Also, subalgorithm Ic does initially construct a list of double coset representatives with redundancies which is later pruned, while in subalgorithm IIb the pruning process is incorporated directly into the main loop. A possible compensation for the additional complexity of Algorithm I is that for many graphs, most of the cases when subalgorithm Ic is called are the trivial cases in which there must be only one double coset.

The first algorithm essentially as described and a variant of the second algorithm not using recursion on the number of distinct labels have been coded in LISP for the Stanford Computation Center's IBM 360/67. The recursive and list processing capabilities of LISP make it well-suited for coding these algorithms.

The empirical evidence obtained in running the coded algorithms clearly indicates that the key recursion in the described algorithms is the recursion on the number of distinct labels. The coded variant of Algorithm II is much slower than Algorithm I. The typical running time for Algorithm I is under .01 per distinct double coset. The described version of Algorithm II should be even more efficient.

6. Example. Let G be the planar graph in figure 3. Using Algorithm II we will determine all topologically distinct labelings of G with one label a, two labels b and three labels c.

The topological symmetry group of G consists of:

π_0 : Identity transformation.

π_1 : Reflection about the line l_1.

π_2 : Reflection about the line l_2.

π_3 : 180° rotation about the center.

The input to Algorithm II is:

U = (111111); n=6; q=3; n_1=1; n_2=2; n_3=3; the two lists corresponding to the symmetry group:

	List 1.	List 2.
π_0:	$(2^5, 2^4\ 2^3\ 2^2, 2^1, 2^0)$	
π_1:	$(2^3$, 24: 25,, $2^2, 2^1, 2^0)$	{(1 0 1 0 ∅ 0 0)}
π_2:	$(2^5, 2^2, 2^3, 2^4, 2^0, 2^1)$	{(0 1 0 1 0 0), (0 0 0 0 1 1)}
π_3:	$(2^3, 2^2, 2^5, 2^4, 2^0, 2^1)$.	{(1 0 1 0 0 0), (0 1 0 1 0 0),(0 0 0 0 11))

First, subalgorithm IIb is called with input:

U = (111111); k=6; m=3; List 1, List 2.

The initial input for the main loop at IIb is:

s(1) = (100000), s(2) = (010000), s(3) = (001000),

s(4) = (000100), s(5) = (000010), s(6) = (000001);

e = (111000); t(1) = 3, t(2) = 2, t(3) = 1.

The loop first determines:

T = $\{\pi_0, \pi_1\}$; $\mathcal{O}(1)$ = {(001000)} ; $\mathcal{O}(j) = \emptyset$, $2 \leq j \leq 5$.

Since $\pi_j(2^3) = 25 > \pi_j(2^5) = 23$ for j=1 and 3, π_1 and π_3 produce no elements for BL (bad list). π_2 produces $f = \pi_2(2^3) \vee \pi_2(2^2) \vee \pi_2(2^1)$ = (1001100) which is mergedinto BL.

At the end of the first time through the main loop of IIb, we have:

GL: (1 1 1 0 0 0)

GLG: $\{\pi_0, \pi_1\}$

BL: (1 0 1 1 0 0).

With the given input, IIb goes through its main loop 8 times producing:

	GL	GLG
e_1:	(1 1 1 0 0 0)	$\{\pi_0, \pi_1\}$
e_2:	(1 1 0 1 0 0)	$\{\pi_0, \pi_2\}$
e_3:	(1 1 0 0 1 0)	$\{\pi_0\}$
e_4:	(1 1 0 0 0 1)	$\{\pi_0\}$
e_5:	(1 0 1 0 1 0)	$\{\pi_0, \pi_1\}$
e_6:	(1 0 0 0 1 1)	$\{\pi_0, \pi_2\}$
e_7:	(0 1 0 1 1 0)	$\{\pi_0, \pi_1\}$
e_8:	(0 1 0 0 1 1)	$\{\pi_0, \pi_1\}$.

Next, the following 6-vector list is computed from the elements of GL:

$w_1 = (1, 1, 1, 0, 0, 0)$, $w_2 = (1, 1, 0, 1, 0, 0)$

$w_3 = (1, 1, 0, 0, 1, 0)$, $w_4 = (1, 1, 0, 0, 0, 1)$

$w_5 = (1, 0, 1, 0, 1, 0)$, $w_6 = (1, 0, 0, 0, 1, 1)$

$w_7 = (0, 1, 0, 1, 1, 0)$, $w_8 = (0, 1, 0, 0, 1, 1)$.

Subalgorithm IIb is called for each w_i . For example, for w_2 , IIb is called with input:

$U = (110100)$; $k=3$; $m=1$; the two lists:

	List 1	List 2
π_0:	$(2^5, 2^4, 2^3, 2^2, 2^1, 2^0)$	$\emptyset$
π_2:	$(2^5, 2^2, 2^3, 2^4, 2^0, 2^1)$	$\{(0\ 1\ 0\ 1\ 0\ 0), (0\ 0\ 0\ 0\ 1\ 1)\}$.

With this input, IIb transfers to the special case section and computes OL = {1, 2} and

	GL	GLG
f_1:	(1 0 0 0 0 0)	$\{\pi_0, \pi_2\}$
f_2:	(0 1 0 0 0 0)	$\{\pi_0\}$.

The main routine determines:

f_1*w_2: (2, 1, 0, 1, 0, 0)

f_2*w_2: (1, 2, 0, 1, 0, 0).

w_1, w_2, w_5 and w_6 each induce 2 distinct labelings of G , and w_3, w_4, w_7 and w_8 each produce 3 distinct labelings of G . The 20 distinct labelings of G with *a, b, b, c, c, c* are given in figure 4.

Acknowledgements. The DENDRAL concept and its applications to organic chemistry were originally conceived by Professor Joshua Lederberg. The authors thank Professor Lederberg for his guidence and his critical suggestions which have made this work possible. The authors also thank Professors B. G. Buchanan, G. Pólya and N. S. Sridharan for their interest in this problem and their valuable comments.

Computer Science Department, Stanford University, Stanford, California 94305

REFERENCES

1. E. F. Beckenbach (Ed.), Applied combinatorial mathematics(Wiley, New York, 1964).

2. B. G. Buchanan and J. Lederberg, The heuristic DENDRAL program for explaining empirical data, IFIP Congess 71, TA-2 (August 1971),91-100.

3. B. G. Buchanan, G. L. Sutherland and E.A. Feigenbaum, A program for generating explanatory hypotheses in organic chemistry, in: Machine Intelligenge 4 .(Edinburgh University Press, Edinburgh, 1969) 209-54.

4. N. G. DeBruijn, Generalization of Pólya's fundamental theorem in enumerative combinatorial analysis, Ned. Akad. Wetenschap., Proc. Ser. A62, Inday. Math. 21 (1956), 59-79.

5. J. Lederberg, Topology of molecules, in: The mathematical sciences (The MIT Press, Cambridge, 1969) 37-51.

6. C. L. Liu, Introduction to combinatorial mathematics (McGraw-Hill, New York, 1968).

7. G. Pólya, Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und Chemische Verbindungen, Acta Math. 68 (1937), 145-254.

8. E. Ruch, W. Hässelbarth and B. Richter, Doppelnebenklassen als Klassenbegriff und Nomenklaturprinzip für Isomere und ihre Abzählung, Theoret. Chim. Acta (Berl.) 19 (1970), 288-300.

9. C. C. Sims, Computations with permutation groups, in: Proceedings of tha second symposium on symbolic and algebraic manipulation (ACM, New York, 1971).

FOOTNOTES

1. This work was supported in part by ARPA Contract SD-183 and NSF Grant GP-16793.

2. A complete description of Pólya's theory of counting can be found, for example, in [1] and [6].

3. The cyclic structure generation algorithms will be described in a later paper.

4. For consistency with our choice of notation, one should always view a labeling α in S_n as a map from the nodes of G to labels in L.

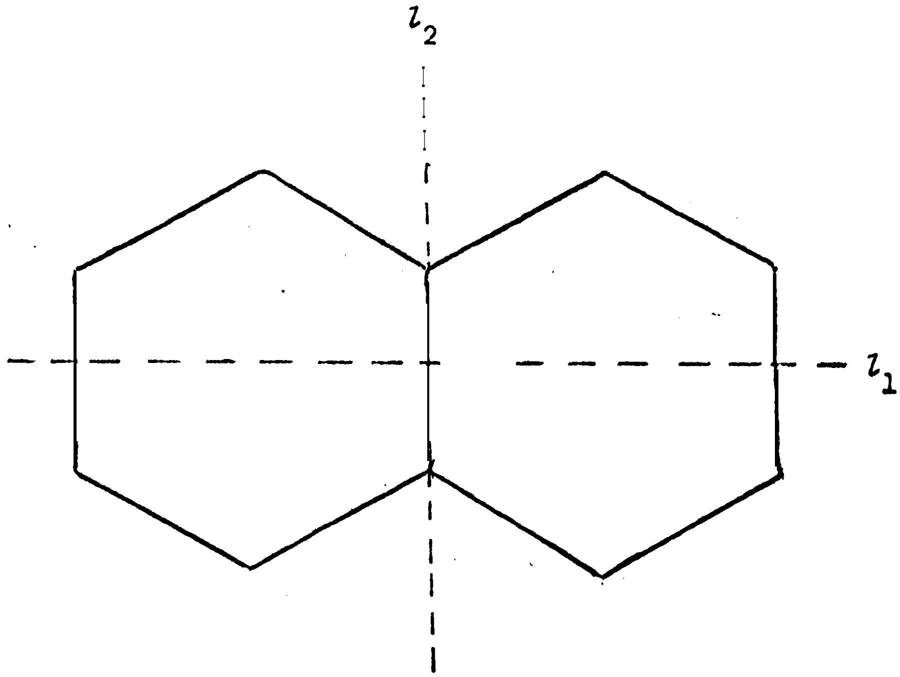
5. Note, however, that in terms of the graph this "canonicalness" is completely dependent on the indexing of the nodes and labels.

6. $\bar{e}^{-1} A \bar{e} \cap B$ corresponds to the subgroup of the topological symmetry group of the graph which preserves the labeling determined by e. This subgroup is needed in many applications of the labeling algorithm.

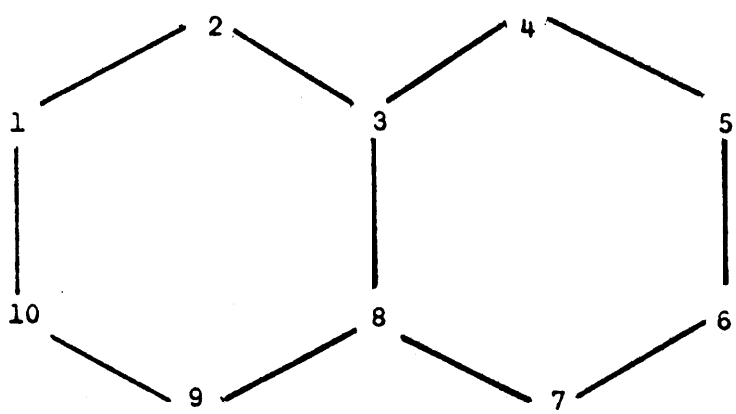
7. Recall that j ε [1,n] is represented by 2^{n-j}.

8. Here we use the property that the inverse of a left coset representative set is a right coset representative set.

Figure 1.

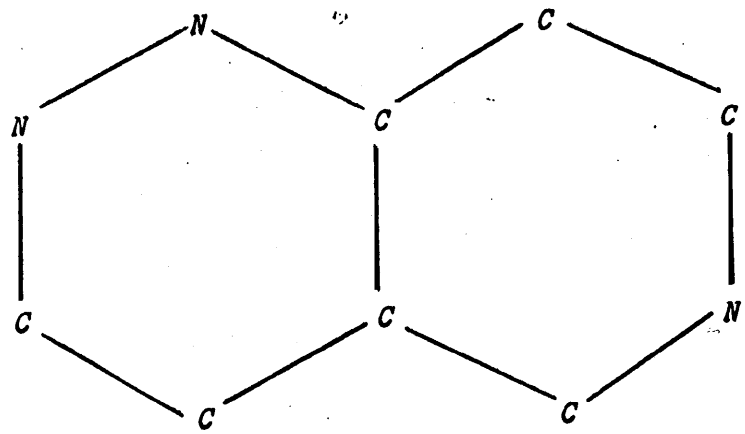


--. (a)

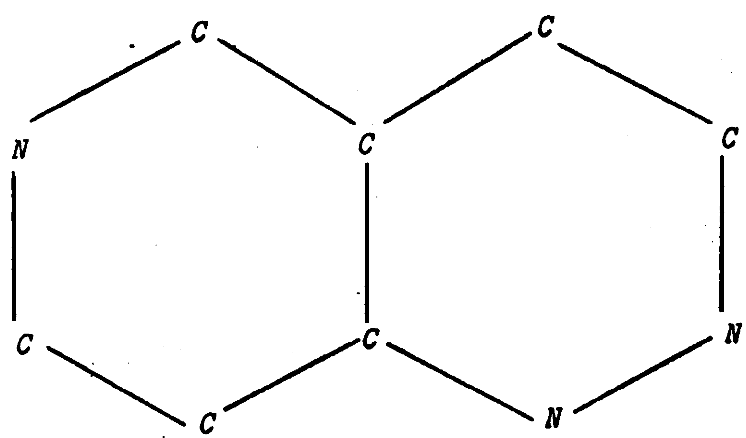


(b)

Figure 2.



(a)



(b)

Figure 3.

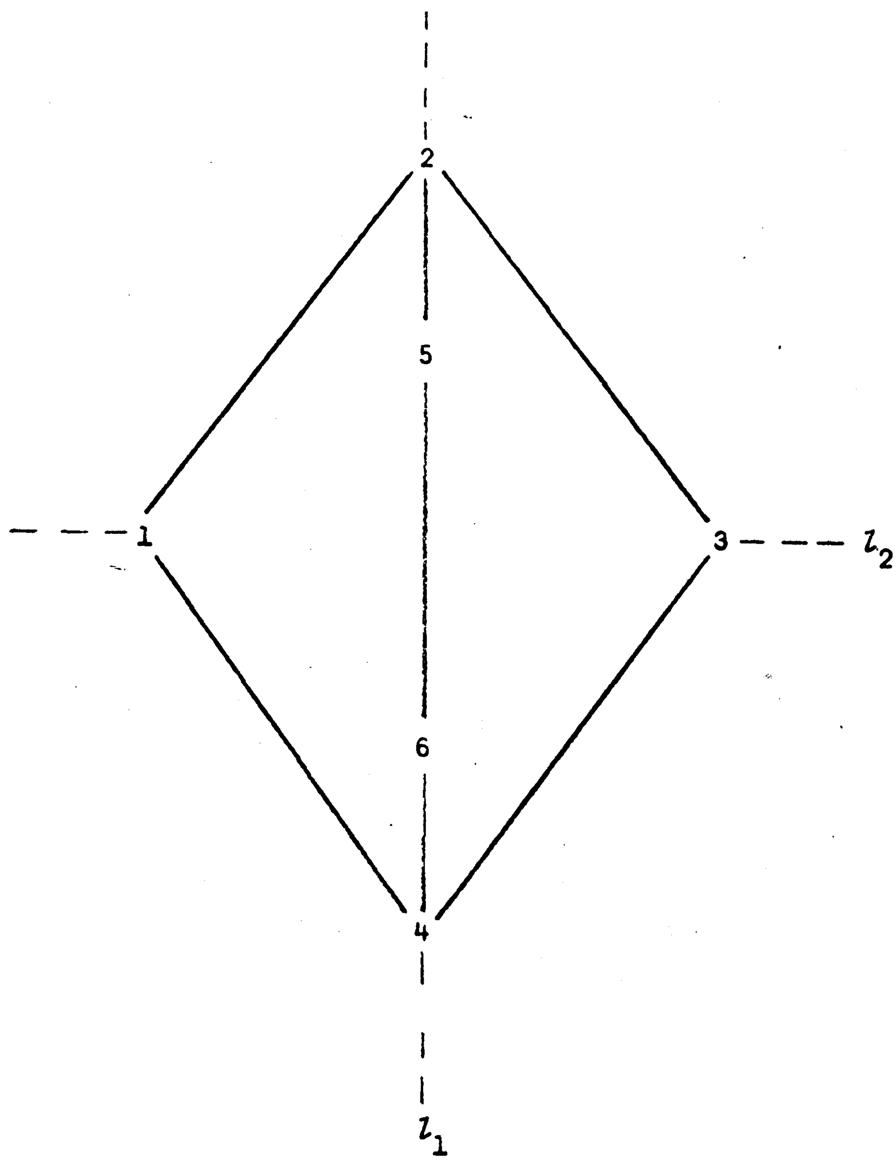


Figure 4.

		b			a				
		c			c				
w_1 :	a		b	b		b			
		c			c				
		c			c				

		b			a				
		c			c				
w_2 :	a		c	b		c			
		c			c				
		b			b				

		b			a			b	
		b			b			a	
w_3 :	a		c	b		c	b		c
		c			c			c	
		c			c			c	

		b			a			b	
		c			c			c	
w_4 :	a		c	b		c	b		c
		b			b			a	
		c			c			c	

		0			c				
		b			a				
w_5 :	a		c	b		c			
		c			c				
		c			c				

$$w_6 : \quad \begin{array}{ccc} & c & \\ & b & \\ a & & c \\ & b & \\ & c & \end{array} \qquad \begin{array}{ccc} & c & \\ & a & \\ b & & c \\ & b & \\ & c & \end{array}$$

$$w_7 : \quad \begin{array}{ccc} & a & \\ & b & \\ c & & C \\ & c & \\ & b & \end{array} \qquad \begin{array}{ccc} & b & \\ & b & \\ c & & C \\ & C & \\ & a & \end{array} \qquad \begin{array}{ccc} & b & \\ & a & \\ c & & c \\ & c & \\ & b & \end{array}$$

$$w_8 : \quad \begin{array}{ccc} & a & \\ & b & \\ C & & c \\ & b & \\ & C & \end{array} \qquad \begin{array}{ccc} & b & \\ & a & \\ C & & c \\ & b & \\ & c & \end{array} \qquad \begin{array}{ccc} & b & \\ & b & \\ c & & C \\ & a & \\ & c & \end{array}$$

SYMBOLS

$\subset$ Set containment

$\cup$ Set union

$\cap$ Set intersection

$\wedge$ Logical and

$\vee$ Logical or