The term two-connected is used here in conjunction with ring-superatoms for a more precise description. For example, biphenyl may be viewed as a single ring system or two rings depending on the chemical context. In this work, however, biphenyl consists of two ring-superatoms (two phenyl rings) linked by a single bond.

(17) Use of the term degree with reference to the degree list refers to the number of bonds other than free valences, with double bonds being counted twice. A free valence may or may not eventually be attached to a hydrogen atom in the final structure.

(18) (a) N. S. Sridharan, Stanford Computer Science Memo STAN-CS-73-381, 1973; (b) Stanford Computer Science Memo STAN-CS-74-404, 1974; (c) L. M. Masinter, unpublished results.
(19) R. S. Cahn, C. K. Ingold, and V. Prelog, *Angew. Chem., Int. Ed. Engl.*, **5**, 385 (1966).
(20) (a) K. B. Wiberg and G. J. Burgmaier, *J. Amer. Chem. Soc.*, **94**, 7396 (1972); (b) K. B. Wiberg, G. J. Burgmaier, K. Shen, S. J. LaPlaca, W. C. Hamilton, and M. D. Newton, *J. Amer. Chem. Soc.*, **94**, 7402 (1972).
(21) B. G. Buchanan, A. M. Duffield, and A. V. Robertson in "Mass Spectrometry, Techniques and Applications," G. W. A. Milne, Ed., Wiley, New York, N.Y., 1971, p 121.
(22) (a) L. A. Paquette, S. Kirschner, and J. R. Malpass, *J. Amer. Chem. Soc.*, **92**, 4330 (1970); (b) S. Masamune and N. Darby, *Accounts Chem. Res.*, **5**, 272 (1972); (c) E. L. Allred and B. R. Beck, *J. Amer. Chem. Soc.*, **95**, 2393 (1973); (d) L. T. Scott and M. Jones, Jr., *Chem. Rev.*, **72**, 181 (1972).
(23) See D. G. Bobrow and B. Raphael, "New Programming Languages for AI Research," Report No. CSL-73-2, 1973, and W. Teitelman, "INTERLISP Reference Manual," Bolt Beranek and Newman and Xerox Corp., 1974, both available from Xerox PARC, 3180 Porter Dr., Palo Alto, California 94304.
(24) I. Ugi, D. Marquarding, H. Klusacek, G. Gokel, and P. Gillespie, *Angew. Chem., Int. Ed. Engl.*, **9**, 703 (1970).
(25) P. Gillespie, P. Hoffman, H. Klusacek, D. Marquarding, S. Pfohl, F. Ramirez, E. A. Tsolis, and I. Ugi, *Angew. Chem., Int. Ed. Engl.*, **10**, 687 (1971).
(26) (a) W. G. Klemperer, *J. Amer. Chem. Soc.*, **94**, 6940 (1972); (b) *ibid.*, **94**, 8360 (1972); (c) *ibid.*, **95**, 380 (1973); (d) *ibid.*, **95**, 2105 (1973).
(27) The form of eq 7 results from the fact that the number of univalents (a_i) cannot exceed the number of free valences necessary to connect the superatoms, leaving one valence free for the radical valence.
(28) NOTE ADDED IN PROOF. Since the acceptance of this manuscript major steps have been taken to effect a partial solution to the problem of constraints. The resulting program is also available *via* SUMEX (see Experimental Section).

Applications of Artificial Intelligence for Chemical Inference. XIII. Labeling of Objects Having Symmetry[1,2]

L. M. Masinter, N. S. Sridharan, R. E. Carhart, and D. H. Smith*

Contribution from the Departments of Computer Science and Chemistry, Stanford University, Stanford, California 94305. *Received March 22, 1974*

Abstract: An algorithm for finding a complete set of nonequivalent labelings of a symmetric object and applications of the algorithm to problems in chemistry are presented.

Combinatorial problems which deal with finding a complete set of nonisomorphic objects under various constraints and based upon various concepts of isomorphism occur in many phases of chemistry. Solutions to some have been achieved using sophisticated applications of graph theory and group theory.[3,4] Perhaps the most common such problem involves attaching, in all unique ways, a fixed set of ligands to a given molecular skeleton.[5,6] This can be viewed as a *labeling* problem, in which all distinct assignments of a given set of labels to the parts of a symmetric object are sought. In developing the concepts set forth by Lederberg,[7] we have found that the labeling problem is common to many aspects of the generation of acyclic and cyclic isomers (see accompanying paper).[2] Herein, our solution to the labeling problem is presented.

Part A of this paper may be read as a brief tutorial on the nature of the problem and an introduction to the terminology found in more technical treatments. Part B is a textual description of a method for the solution of this type of problem. Part C is a summary of the procedure in a more algorithmic form; an even more formal description and a proof of correctness is available elsewhere.[8] In part D, certain generalizations of the basic algorithm are presented. Finally, in part E, a sample application of the method to a complex isomerism problem in organic chemistry is outlined.

The algorithm described here represents a concrete procedure for the solution of problems which previously were solvable only *via* "intuition." An intuitive approach to labeling problems is satisfactory for small cases, but can easily break down when applied to more complex problems, *e.g.*, the adamantane example in part D. It has been known how to compute the number of solutions,[4b,9] but an efficient method of actually constructing the solutions has not previously been published;[10] certainly the latter information is of substantially greater use to a chemist, as he is interested in the identity of each structure.

A computerized version of the labeling algorithm has been coded in the INTERLISP language. It is imbedded within the structure generator described in the accompanying paper,[2] and a listing of the source text of the program will be provided by the authors upon request. The program itself is available for use as described in the Experimental Section of ref 2.

The labeling program has been tested extensively using cases for which numbers of solutions have been published.[4b,9c] In each case, the program was used to construct the complete set of labelings, the solutions were counted, and the count was compared with the published value. No discrepancies have been found except in the largest case checked by the program of the node labeling of naphthalene (six labels of one type, two of a second, and two of a third). In this case, Balaban and Harary[4b] obtained a count of 329 solutions, while we obtained 330; further investigation showed that the expansion of the "generating function"[4b] for naphthalene was incorrect in this and at least one higher term, and that 330 was in fact the correct figure.

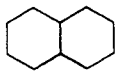
Part A. Definitions

The object to be labeled may correspond to any finite collection of discrete parts (*e.g.*, faces of a cube or atoms of a molecule), as long as the symmetries of the object can be described as rearrangements of the parts which leave the object unchanged. The labels may be any properties or entities which can be associated with the parts (*e.g.*, colors, ligands, isotopic "tags"), as long as the labels themselves

are not altered by the symmetry operations on the object. Although the method described here is general, parts A–C of this paper will be concerned with a specific type of problem: the labeling of the *nodes* of a *graph*, where each node receives just one label. In chemical terms, a graph is a molecular skeleton ("cyclic skeleton"[2]) devoid of both atom names and three-dimensional information. The atom positions in a graph are called nodes, and the bonds are called *edges* (multiple edges are allowed). Discussion of the generalizations of the method is postponed until part D.

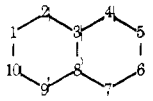
Numbering and its Relationship to Symmetry. In the discussion of symmetry, it is necessary for one to define some frame of reference within which to work. In this paper, a *numbering* of the graph (*i.e.*, an association of integers 1 through n to the n nodes of the graph) is used for this purpose, and the relationship between different numberings is used to characterize symmetry.

For a fixed orientation of an n-node graph, there are $n!$ ways of numbering. If the graph has no symmetry, then each of these ways is distinct from the rest. However, if there is some symmetry, as for example, in the decalin skeleton (**1**), then relative to any particular numbering (*e.g.*,

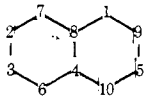


1

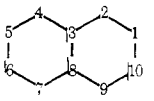
2a), some of the ways are different (*e.g.*, **2b**), while others are symmetrically equivalent (*e.g.*, **2c**). Intuitively, **2a** and



2a



2b



2c

2c are equivalent, because one can rotate **2a** about the 3–8 axis to obtain **2c.** There is an explicit way of determining the topological "sameness" of such numberings which is easily applied to even complicated cases, and which is well suited to computer applications.

> Two numberings of a graph are *equivalent* if the connection table derived from one can be made identical with that derived from the other by rearrangements of the rows and of the connection lists within each row.

Table I contains the connection tables of structures **2a–c,** written in a standard, *ordered* form (with node numbers ascending and each connection list ascending). Because the ordered connection table for **2a** is identical with that of **2c,** while that of **2b** is different, **2a** and **2c** are equivalent, while **2b** is distinct.

The above definition of equivalence deals only with topological properties. If two numberings of a graph are equivalent, then node i has the same valence in each, the nodes i and j are connected similarly in each, for all i and j. One result is that, for labeling problems in chemistry, the valences of an atom are completely interchangeable (*i.e.*, these valences have no spatial orientation). When stereochemical considerations are important, topological equivalence may not be sufficiently strong. In such cases other types of equivalence, phrased in terms of superimposability of two numbered skeletons, can be used. The test for topological equivalence is particularly simple, though, and is satisfactory for many chemical situations.

Table I. Connection Tables for Structures **2a–c**

2a		2b		2c	
Node	Connection List	Node	Connection List	Node	Connection List
1	2,10	1	8,9	1	2,10
2	1,3	2	3,7	2	1,3
3	2,4,8	3	2,6	3	2,4,8
4	3,5	4	6,8,10	4	3,5
5	4,6	5	9,10	5	4,6
6	5,7	6	3,4	6	5,7
7	6,8	7	2,8	7	6,8
8	3,7,9	8	1,4,7	8	3,7,9
9	8,10	9	1,5	9	8,10
10	1,9	10	4,5	10	1,9

Permutations and Permutation Groups. Given a numbering of a graph as a reference, one can use a condensed notation to write down other numberings. All that is needed is a list of integers which are reassigned, respectively, to nodes 1 through n of the reference. Using **2a** as a reference, condensed notations for **2a–c** are given in Table II. In the **2b** case, the row of numbers means that the node numbered 1 in **2a** is now numbered 2, the node numbered 2 in **2a** is now numbered 7, and so on.

Table II. Condensed Notations for Numberings **2a–c**

(Reference) **2a**:	1	2	3	4	5	6	7	8	9	10
b:	2	7	8	1	9	5	10	4	6	3
c:	5	4	3	2	1	10	9	8	7	6

With this notation, one can view a numbering as a transformation which carries the integers $(1, 2, \ldots, n)$ into themselves in a one-to-one fashion. Such transformations are called permutations. The permutation π for **2c,** for example, is $\pi_{2c}(1) = 5$, $\pi_{2c}(2) = 4$, $\pi_{2c}(3) = 3, \ldots, \pi_{2c}(10) = 6$, while that for the reference is the identity $\pi_{2a}(i) = i$, $i = 1, 2, \ldots, n$.

The symmetry of any graph is fully described by the set of numberings which are equivalent to the reference. The permutations corresponding to these satisfy the properties of a mathematical group, called the *symmetry group* of the graph. More explicitly:

> The symmetry group of a graph is the set of all permutations whose corresponding numberings yield ordered connection tables identical with that of the reference numbering.

If, in the definition of equivalence, properties other than the connection table are used, other types of symmetry groups may be defined in an analogous manner. For the decalin skeleton, there are four permutations in the symmetry group. These topological symmetries, given in Table III,

Table III. The Symmetry Group of the Decalin Skeleton[a]

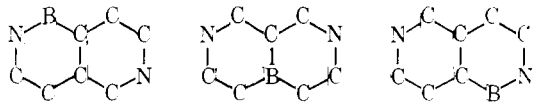
π_I	1	2	3	4	5	6	7	8	9	10
π_v	5	4	3	2	1	10	9	8	7	6
π_h	10	9	8	7	6	5	4	3	2	1
π_{180}	6	7	8	9	10	1	2	3	4	5

[a] The reference numbering here corresponds to **2a**.

correspond directly to the geometric symmetries π_I = identity, π_v = rotation about the vertical (3–8) axis, π_h = rotation about the horizontal axis, and π_{180} = in-plane rotation (all are 180° rotations).

Representation of Labelings. Once a reference numbering has been chosen, one can define a compact notation for labelings as well as numberings. All that is needed is a list of symbols which represent the labels associated, respectively, with nodes 1, 2, . . . , n in the reference. Thus, with **2a** as the reference, labelings **3a–c** of the nodes of the decalin skeleton have the following representations.

In these cases, the symbols used for representation are



3a 3b 3c

3a: N B C C C N C C C C
b: N C C C N C C B C C
c: N C C C C N B C C C

identical with the labels themselves, but in general any kind of symbol may be used as long as the correspondence between symbols and labels is understood.

Equivalence Classes and Canonical Labelings. Permutations, discussed earlier in terms of numberings, can be thought of as operators which act upon labelings. The action of the permutation

$$I_1 I_2 \ldots I_n$$

on the representation of a labeling is interpreted as follows. Replace the symbol in position 1 with that in I_1, replace the symbol in position 2 with that in I_2, and so on. Thus, the permutation

$$\pi = 1\ 8\ 3\ 4\ 6\ 5\ 7\ 2\ 10\ 9$$

acting upon **3a** gives **3b.**

Two labelings of the nodes of a graph are equivalent if and only if at least one permutation in the group of the graph, when acting upon one labeling, yields the other. Thus **3a** and **3c** are equivalent, because π_{180} in Table III, when applied to **3a,** yields **3c.**

Given a labeling, it is possible to generate all other labelings which are equivalent by applying to it each permutation in the symmetry group. The labelings obtained in this way, taken as a set (*i.e.*, with identical representations eliminated), form an *equivalence class* of labelings. Table IV shows the equivalence class to which **3a** (and **3c**) belongs.

Table IV. The Equivalence Class[a] of Labelings to Which **3a** Belongs

Permutation[b]	Results when applied to **3a**									
π_1	N	B	C	C	C	N	C	C	C	C
π_5	C	C	C	B	N	C	C	C	C	N
π_{21}	C	C	C	C	N	C	C	C	B	N
π_{180}	N	C	C	C	C	N	B	C	C	C

[a] The reference numbering corresponds to **2a**. [b] Taken from Table III.

The problem treated in this paper is one of finding a complete list of nonequivalent labelings, which amounts to selecting exactly one representative from each equivalence class of labelings. This goal is realized most efficiently if one can define some property of the labelings which automatically distinguishes one member of each class as somehow unique. One method is to sort the labelings on the basis of their representations, thus establishing an ordering of them. One can then define the unique labeling in an equivalence class to be the "smallest" member, that is, the one which occurs first in the sorted list. Such a labeling will be called a *canonical* one.

To sort a list of representations, one needs (in addition to a reference numbering) an ordering of the label symbols. This ordering is arbitrary but must be used consistently for any particular equivalence class. For example, in labeling the decalin skeleton with two N's, one B, and seven C's, one might choose the alphabetic convention B < C < N. Once the label symbols have been ordered, the representations of any two labelings can be compared, symbol-by-symbol, with the first inequality establishing the overall ordering of the labelings. (This is just a "dictionary" ordering of representations.) Thus, **3a** is "smaller" than **3c** because, although the first symbols in their representations are equal, the second symbol (B) of **3a** is "smaller" than the second symbol (C) of **3c.** With the alphabetic convention, π_5(**3a**) in Table IV is the smallest member of its equivalence class and is thus a canonical labeling.

The concept of canonicity is used in the kernel technique (below) to establish specific "target" labelings in each equivalence class. Thus, the problem of generating *all* members of a class and comparing them to eliminate symmetry duplicates is avoided.

Part B. Solution to the Labeling Problem

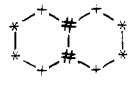
An obvious method of finding the distinct labelings would be to generate all possible labelings and, for each, to check if an equivalent one was previously constructed. Unfortunately, this method can take an exorbitant amount of computation. Below, a method is discussed which we believe uses an amount of time roughly proportional to the number of solutions (*i.e.*, the number of equivalence classes of labelings) and requires only knowledge of the symmetry group, in terms of permutations. Thus, the procedure is useful for labeling objects using their geometric symmetry[11] as well as the topological symmetry defined above. We first discuss several special cases, then outline the solution to the general problem.

1. Special Cases. There are three special cases of labeling in which the problem can be solved immediately. Although they may be amenable to treatment with the more general algorithm, their solution is computationally simpler. These special cases are frequently encountered in the reductions (see below) of the general problem.

1.1. One Type of Label. If the number of labels of a given type is equal to the number of nodes to be labeled, then there is only one way to carry out the labeling. A check for this trivial case is necessary, because subproblems of this form are often encountered during orbit recursion (see below).

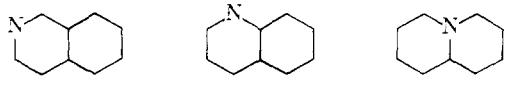
1.2. Two Types of Labels and One Label of a Given Type. In cases where there is one label of a given type and $n - 1$ of another (where n is the number of nodes to be labeled), it is only necessary to identify the classes of symmetrically related nodes, or *orbits*,[12] and, for each orbit, to associate the single label with one node therein. Thus, the number of distinct labelings is equal to the number of orbits. Within each orbit, the single label is by convention associated with the node which bears the smallest reference number.

For the decalin skeleton, there are three orbits, marked with *, +, and # in **4.** If the graph is to be labeled with one



4

N and nine "blanks," each distinct labeling corresponds to the association of an N with the "first" node of each orbit. Thus, there are three distinct labelings **5a–c** (**2a** is used as the reference numbering).



5a 5b 5c

1.3. Two Types of Labels and an Unsymmetric Graph. When there is no symmetry (*i.e.*, the group consists of only the identity permutation), and there are two label types (n_1 of the first type and $n - n_1$ of the second), the labeling re-

duces to a simple combinatorial problem: given n distinct objects, find all distinct ways of selecting n_1 of them. This can be accomplished by the following recursive[13] algorithm. To find all selections of k objects out of a set S whose size is n

(1) If $k = 1$, pick each element of the set S, in turn, to obtain n solutions.

(2) If $k = n$, the set S comprises the only solution.

(3) Otherwise, pick an element x from S:

(a) Find all selections of k objects out of the set $S - (x)$. (b) Find all selections of $k - 1$ objects out of the set $S - (x)$, and to each of these add the element x. (c) The solution is the union of the results from steps 3a and 3b.

A subset of S with k elements either contains the element x or not. In case 3a, one finds those selections which do not contain x, while in 3b, one finds those that do. Each of these cases is simpler than the original selection problem, because the size of the set, as well as the value of k in 3b, is reduced. The terminating conditions ($k = 1$ or k equal to the size of the set) ensure that the process will halt.

2. General Case. In the general labeling case, there are two important techniques used to reduce the problem. The first is called *label recursion*[13] and the second *orbit recursion*. The idea behind label recursion is that one can deal with just two types of labels at a time. The idea behind orbit recursion is that one can label just one orbit at a time. These reductions are discussed in detail below.

2.1. Label Recursion. If one is given many (more than two) kinds of labels, say n_1 of type 1, n_2 of type 2, . . . , n_k of type k, one may proceed as follows. Solve the labeling problem for n_1 labels of type 1 and $n_2 + n_3 + \ldots + n_k$ labels of another type, called "blank." Take each of the results and label the "blank" nodes with n_2 labels of type 2 and $n_3 + \ldots + n_k$ "blanks," and so forth. It has been proved[8] that the result of this series of steps, each of which is carried out with only two types of labels, is a list of all distinct solutions to the original problem.

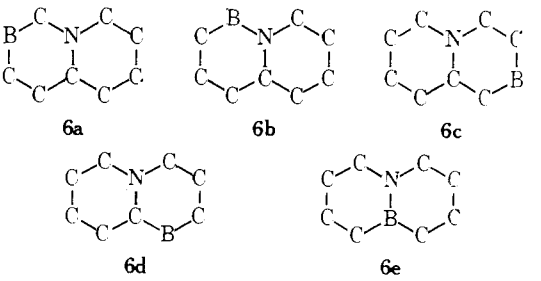
Each labeling step may reduce the symmetry of the object being labeled. Therefore, after each such step it is necessary to calculate the new group, termed the *reduced symmetry group*,[14] of the graph before proceeding to the next step.

It is computationally most efficient to order the k types of labels so that $n_1 \leq n_2 \leq \ldots \leq n_k$. Special cases are more likely to be encountered immediately, and, in general, it is easiest to label a graph when the number of labels of one type is small. Also, each labeling tends to reduce the symmetry of the graph, making subsequent labelings simpler.

To treat an example, consider the labeling of the decalin skeleton with one N, one B, and eight C's. One first labels with one N and nine "blanks," an instance of special case 1.2. The result is the set of three labelings discussed above, **5a–c.** There are now three new problems: to label the "blanks" of **5a–c,** under their respective reduced symmetries, with one B and eight C's, and again special case 1.2 is applicable. For **5a** and **5b,** placement of the N has destroyed all symmetry of the graph, and thus each "blank" has its own orbit. Therefore, there are nine distinct labelings in each case. For **5c,** there are five orbits in the reduced symmetry group, and five labeled structures result (**6a–e**).

Note that the above labelings all reduce to one of the special cases. Had there been more than one N or B, the methods described below would have been needed.

2.2. Orbit Recursion. As a result of label recursion, each labeling step is carried out with at most two types of labels, say n_1 of one type and n_2 of another. Here, $n_1 + n_2 = n$,



6a **6b** **6c** **6d** **6e**

the number of nodes to be labeled. When no special cases apply, one has by definition a graph with some symmetry, and $n_1, n_2 > 1$. If, in addition, the nodes fall into more than one orbit, further simplification is possible.

Suppose one of the orbits is chosen, customarily the one containing the node of lowest reference number. The problem can then be treated in two stages. First, the nodes within the orbit are labeled with a subset of the given label set; then, for each partial labeling thus obtained, the remaining nodes (*i.e.*, those not in the chosen orbit) are labeled with the remaining labels, using the reduced symmetry group. Several cases must usually be considered in this fashion, each corresponding to a distribution of the given labels over the two types of nodes.

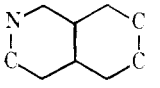
Consider, for example, the labeling of the decalin skeleton with three N's and seven C's. The four-node orbit (1,5,6,10) is chosen, and the labels are partitioned into the four possible distributions shown in Table V. Each case may

Table V. Partitions of Three N's and Seven C's between the Orbit (1,5,6,10) and the Remaining Nodes in the Decalin Skeleton[a]

Case no.	Labels going to orbit (1,5,6,10)		Labels going to remaining nodes	
	No. of N's	No. of C's	No. of N's	No. of C's
1	3	1	0	6
2	2	2	1	5
3	1	3	2	4
4	0	4	3	3

[a] The reference numbering here corresponds to **2a**.

be solved independently. Focusing upon case 3, the first stage involves the labeling of (1,5,6,10) with one N and three C's. This is an instance of special case 1.2, and because there is only one orbit, one labeling (**7**) results. The

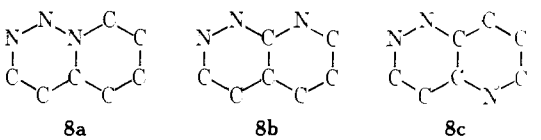


7

second stage involves the labeling of the remaining nodes (2,3,4,7,8,9) with the two N's and four C's. The initial labeling has removed all symmetry from the graph, and thus this stage reduces to special case 1.3. There are 15 ways to pick two elements from (2,3,4,7,8,9):

(2,3), (2,4), (2,7), (2,8), (2,9)
(3,4), (3,7), (3,8), (3,9)
(4,7), (4,8), (4,9)
(7,8), (7,9)
(8,9)

Associating the two N's with the two chosen nodes in each case yields 15 distinct labelings, the first three of which are shown (**8a–c**).



8a **8b** **8c**

2.3. Kernel Technique. Label and orbit recursion usually reduce the general labeling problem to one of the special cases given above. However, situations occur (*e.g.*, case 2 of Table V) in which an m-node orbit is to receive $m_1 > 1$ labels of a first type and $m_2 > 1$ labels of a second. No further reductions are possible, and a special method called the *kernel technique* has been developed to treat this fundamental labeling problem.

The goal of the kernel technique is the generation of all canonical labelings (see part A) of the orbit with the given label set. In the ordering of representations, it will be assumed here that a label of the first type is "smaller" than one of the second type. The procedure is a constructive one in which nodes with successively higher node numbers are given labels. At a typical intermediate stage, partial labelings L_k of the first k nodes are used as starting points for the labeling of the $(k + 1)$th node. For each L_k, one or two cases are considered. The $(k + 1)$th node may receive a label of either type, but one or the other of these possibilities may be invalid (*i.e.*, if m_1 labels of the first type or m_2 labels of the second already appear in L_k). The newly constructed partial labelings L_{k+1} are then tested as described below, and only those which do not violate the canonicity condition are retained for the generation of the L_{k+2}'s. It can be shown that any canonical labeling must associate a label of the first (smaller) type with the first node of the orbit, and thus only one L_1 is needed to begin the procedure. The process terminates when k reaches m, and the complete labelings are tested for canonicity in the usual fashion (see part A).

The purpose of testing the partial labelings is to eliminate, as early in the construction as possible, those which can never form the basis for a canonical labeling. An attempt is made, therefore, to show that for at least one permutation π in the (possibly reduced) symmetry group of the graph, a given partial labeling L_j could never be "smaller" than the image $\pi(L_j)$ of L_j under π. Preliminary to the presentation of the actual test, it is necessary to discuss the representation of partial labelings.

A partial labeling L_j can be represented as a list of m symbols which indicate the labels associated, respectively, with the first, second, . . . , mth node of the orbit. For clarity, label symbols are omitted for nodes which are not in the orbit. The symbol 0 stands for a label of the first type, 1 for a label of the second type, and the symbol "_" is used to indicate that a node has not yet been labeled. Thus, the first j symbols in the representation of L_j are of the 0–1 type, and these are followed by m–j symbols "_." The representation for $\pi(L_j)$ contains the same number of 0's, 1's, and symbols "_" but (possibly) in a different order, determined by the action of π on the representation of L_j (see part A).

The testing is done by filling in the symbols "_" of L_j with 0's, thus defining a full labeling, called $\min(L_j)$, which is as small as or smaller than any other labeling created from L_j. Similarly, for each π, the symbols "_" of $\pi(L_j)$ are filled in with 1's, thus defining a labeling, called $\max(\pi(L_j))$, which is as large as or larger than any image of L_j under π. If there is a π such that $\min(L_j)$ is larger than $\max(\pi(L_j))$, then L_j cannot possibly give rise to a canonical labeling and may thus be discarded. Figure 1 shows an example of a partial labeling which would be eliminated in this way, assuming C < N. Here

$$L_3 = 0\ 1\ 0\ -\ -\ -$$
$$\min(L_3) = 0\ 1\ 0\ 0\ 0\ 0$$

and for the threefold rotation

$$\pi = 3\ 1\ 2\ 6\ 4\ 5$$

one obtains

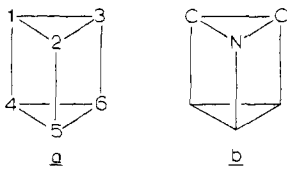


Figure 1. The skeleton corresponding to prismane: (a) reference numbering; (b) a partial labeling from which no canonical full labeling can be derived.

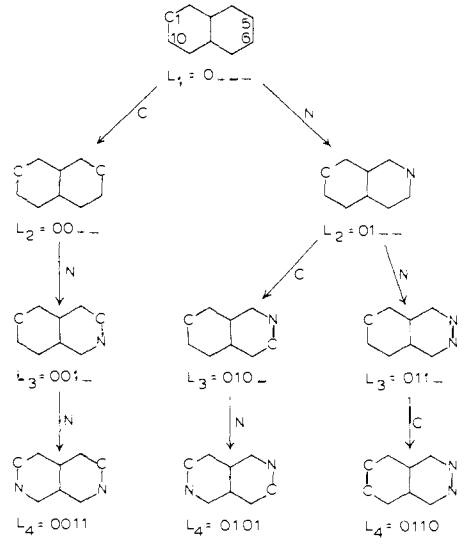


Figure 2. Summary of the steps in labeling the orbit (1,5,6,10) of the decalin skeleton with two C's and two N's. The reference numbering corresponds to **2a** and the label symbols are ordered C < N.

$$\pi(L_3) = 0\ 0\ 1\ -\ -\ -$$
$$\max(\pi(L_3)) = 0\ 0\ 1\ 1\ 1\ 1$$

Because the first elements of $\min(L_3)$ and $\max(\pi(L_3))$ are equal, the second elements must be compared, giving

$$\min(L_3) > \max(\pi(L_3))$$

This indicates that L_3 may be excluded in the construction of all further labelings.

If a full labeling is tested in this fashion (*i.e.*, if there are no symbols "_" in the representation), this procedure reduces to the standard canonicity test.

In the decalin example, the orbit labelings are particularly simple in that a "noncanonical" partial labeling is never generated. Figure 2 summarizes the steps in labeling the orbit (1,5,6,10) with two N's and two C's (case 2 of Table V), a problem for which there are three solutions. To complete this case, one would need to compute the reduced symmetry group of each of the three labelings in Figure 2, and for each to label the remaining nodes with one N and five C's using special case 1.2.

Part C. Summary of Labeling Steps

The Problem. Given a graph with n nodes, along with its symmetry group, find all nonequivalent ways of attaching to the nodes n given labels which are not necessarily distinct. The symmetry group is usually expressed as a set of permutations relative to a reference numbering of the graph.

The Solution. The steps are best described as three nested, recursive procedures. The process begins with a call[15] to procedure I; the results returned from this call constitute the desired set of labelings. The parallel between the following algorithm and the preceding textual description is indicated by the numbers within square brackets. These

refer to the section numbers in part B.

Procedure I. Any Number of Label Types and Orbits.

(A) If there are less than three label types, call procedure II directly and return (from I) the results.
(B) Otherwise, carry out label recursion [2.1] as follows.
(1) Calling procedure II, label the nodes with labels of one type, using "blank" labels for the remaining types.
(2) For each of these partial labelings, compute the reduced symmetry group and, calling procedure I recursively, label the "blank" nodes with the remaining labels.
(3) Return (from I) the list of full labelings thus obtained.

Procedure II. Two Label Types, Any Number of Orbits.

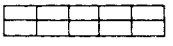
(A) Test for special cases [1] as follows.
(1) If there is only one label type [1.1] (trivial case), return the labeling directly.
(2) If there is only one label of either type [1.2], compute the orbits and return (from II) the list of labelings which result from assigning the single label to the first node in each orbit.
(3) If the symmetry group consists of only the identity permutation [1.3], call the algorithm described in part B, section 1.3, and return (from II) the results.
(B) Compute the orbits.
(C) If there is only one orbit, call procedure III directly and return (from II) the results.
(D) Otherwise, carry out orbit recursion [2.2] as follows.
(1) Choose the orbit which contains the node of lowest reference number and partition the labels in all possible ways between the orbit and the remaining nodes.
(2) For each partition, do the following: (a) call procedure II recursively to label the chosen orbit with the label set for this partition; (b) for each of these partial labelings, compute the reduced symmetry group and, calling procedure II recursively, label the remaining nodes with the remaining labels for this partition.
(3) Return (from II) the collected results of (2).

Procedure III. Two Label Types and One Orbit. "Kernel Technique" [2.3].

(A) Attach a label of the first type to the first node of the orbit.
(B) Extend the current labeling as follows.
(1) If all nodes of the orbit have been labeled, then place the labeling on the output list and go to (4).
(2) If any labels of the first type remain, then (a) place a label of the first type on the first unlabeled node of the orbit; (b) test this extended labeling L_k checking that, for each permutation π in the (possibly reduced) symmetry group of the graph, $\min(L_k)$ is not larger than $\max(\pi(L_k))$; (c) if the test is passed, call (B) recursively to further extend the labeling; (d) remove the label placed in (a), thus "contracting" the labeling to its previous state.
(3) If any labels of the second type remain, exercise a procedure completely analogous to steps 2a–d, but using labels of the second, rather than first, type.
(4) Return (from B).
(C) Return (from III) the output list.

Part D. Generalizations of the Method

The labeling algorithm has been described in the context of labeling the nodes of a graph. However, the only information needed to carry out these steps is: (a) the number of each type of label; and (b) a group of permutations describing the symmetry of the object relative to a reference numbering of the parts. Thus, the procedures are applicable to any problem for which the above information is available. For example, the labeling of the decalin skeleton with three N's and seven C's is equivalent to finding all ways of blackening three (and whitening seven) of the rectangular faces of **9** such that no symmetry duplicates are produced, assuming that **9** is free to undergo rotations and reflections in the plane.



9

In our approach to the exhaustive generation of cyclic and acyclic isomers,[2] there are two general types of problems other than node labeling: the labeling of *edges* and of *free valences*. In each case, a simple modification of the basic algorithm allows for an efficient solution to the problem.

Edge Labeling. Under some circumstances, one might wish to label the edges of a graph rather than the nodes. Instead of numbering the edges and obtaining the symmetry group independently, one can use the group on the nodes to simplify the situation. Let the representation for each edge be the unordered pair of numbers indicating the nodes at the end points of the edge. Then each permutation π_i in the symmetry group of the graph gives rise to a permutation π_i^e in the symmetry group on the edges, defined by

$$\pi_i^e(j,k) = \pi_i(j), \pi_i(k)$$

Thus, for example, π_v in Table III generates π_v^e as follows

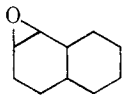
$$\pi_v^e(1,2) = \pi_v(1), \pi_v(2) = 5,4 = 4,5$$
$$\pi_v^e(2,3) = \pi_v(2), \pi_v(3) = 4,3 = 3,4$$

and so on. Table VI shows the complete symmetry group on the edges of the decalin skeleton (**2a** is used as the reference numbering).

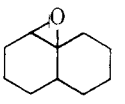
Finding the set of possible epoxides derived from the decalin skeleton is an example of an edge-labeling problem. Here, there are 11 labels: one bridging oxygen and ten "blanks." The labeling falls under special case 1 of the algorithm. Examination of Table VI shows that the edges fall into four orbits:

(1,2	4,5	9,10	6,7)
(2,3	3,4	8,9	7,8)
(3,8)			
(5,6	1,10)		

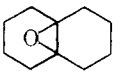
Assigning the single label to one edge in each orbit, one obtains the four solutions **10a–d.**



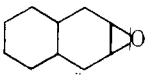
10a



10b



10c



10d

In a graph with one or more multiple edges, the same procedure can be used to obtain the symmetry group, but during the labeling, multiple edges need to receive more than one label. This problem shares some common features with free-valance labeling, and the general approach will be discussed at the end of the following section.

Free-Valence Labeling. In the case of free-valance labeling, each node in the graph has some number of equivalent[16] free valences, which are the parts to be labeled. The labels consist of ligands. Rather than number the free valences and compute the symmetry group independently, one can use the group on the nodes (which may be a reduced symmetry group if some of the original symmetry of the graph is destroyed by the distribution of free valences and/or by the presence of atom names on the nodes) together with a modified algorithm.

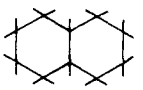
One modification affects special case 1.3 (see part B), which is now inapplicable unless each free valence to be labeled is attached to a different node. This is because even though a structure has no symmetry interrelating its atoms, there will still be valid permutations which interchange the free valences upon any node that carries two or more.

Table VI. Symmetry Group[a] of the Edges of the Decalin Skeleton[b]

π_1[c]	1,2	2,3	3,8	3,4	4,5	5,6	6,7	7,8	8,9	9,10	1,10
π_v[c]	4,5	3,4	3,8	2,3	1,2	1,10	9,10	8,9	7,8	6,7	5,6
π_h[c]	9,10	8,9	3,8	7,8	6,7	5,6	4,5	3,4	2,3	1,2	1,10
π_{180}[c]	6,7	7,8	3,8	8,9	9,10	1,10	1,2	2,3	3,4	4,5	5,6

[a] See Table III for the node permutations which generate these edge permutations. [b] The reference numbering here corresponds to **2a**.

A second modification affects orbit recursion. The orbits of the nodes are calculated, and one orbit is selected as before, but one must then distribute the labels over the free valences (rather than nodes) within and not within the orbit. The m nodes in the chosen orbit have the same number, say v, of free valences, and thus the orbit must be assigned ($m \times v$) labels. Suppose, for example, that one wishes to label the 18 free valences in **11** with 4 OH's and 14



11

H's. If the chosen orbit is (1,5,6,10) (**2a** is the reference numbering), then $m = 4$ and $v = 2$. Thus, the orbit must receive eight labels and the remaining nodes, ten. The possible distributions are shown in Table VII.

The final modification influences the kernel technique, which is now carried out in two stages: a *grouping* stage and a *labeling* stage. In the grouping stage, the labels are partitioned, in all distinct ways, into m sets of v labels each (m and v being respectively the number of nodes in the orbit and the number of free valences on each node). Each set is called a *multilabel of degree* v and represents a set of ligands which may be attached to any node of the orbit. A familiar example would be the *gem*-dimethyl group, a multilabel of degree 2. Because the multilabels of each partition are to be associated with nodes rather than free valences, standard node-labeling techniques may be used in the labeling stage to find all unique associations.

In the decalin example, consider case 1 of Table VII.

Table VII. Partitions of 4 OH's and 14 H's between the Free Valances of Orbit (1,5,6,10) and the Remaining Free Valences of **11**[a]

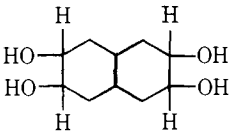
	Labels going to orbit (1,5,6,10)		Labels going to remaining nodes	
Case. no.	No. of OH's	No. of H's	No. of OH's	No. of H's
1	4	4	0	10
2	3	5	1	9
3	2	6	2	8
4	1	7	3	7
5	0	8	4	6

[a] The reference numbering here corresponds to **2a**.

Here, there are four OH's and four H's to be attached to the eight free valences of the orbit (1,5,6,10) in **11**. There are three ways to partition the eight labels into four multilabels of degree 2.

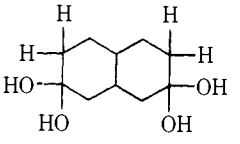
(OH,OH) (OH,OH) (H,H) (H,H) (partition 1)
or (OH,OH) (OH,H) (OH,H) (H,H) (partition 2)
or (OH,H) (OH,H) (OH,H) (OH,H) (partition 3)

For each partition, the nodes of the orbit (1,5,6,10) must be labeled with the multilabel set. This is particularly simple for partition 3, because only one type of multilabel [(OH,H)] is present. Application of special case 1.1 (part b) yields only one solution (**12**).

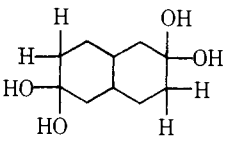


12

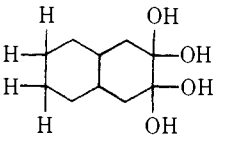
In partition 1, there are two multilabels of the type (OH,OH) and two of the type (H,H), and thus, the standard kernel technique must be used. The situation here is precisely analogous to the labeling of the nodes in the orbit (1,5,6,10) with two N's and two C's, which was given as an example in part B, section 2.3 (see also Figure 2). Three labelings (**13a-c**) result from this application of the standard kernel technique to partition 1.



13a

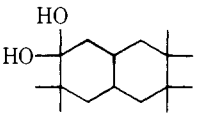


13b



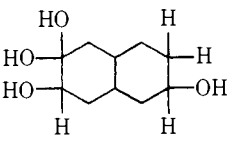
13c

Partition 2 is somewhat more complex. There is one label of the two types (OH,OH) and (H,H), and two of type (OH,H). Because there are three label types, label recursion (part B, section 2.1) is necessary. As the first step, the nodes may be labeled with one (OH,OH) and three "blanks," for which special case 1.2 is used. There is only one orbit, so one partial labeling (**14**) results. The "blanks"

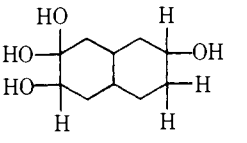


14

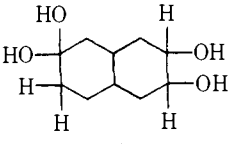
of 14 must then be labeled with two multilabels (OH,H) and one multilabel (H,H), and again special case 1.2 is applicable. Because **14** has no symmetry relating its nodes, each of 5, 6, and 10 constitutes a separate orbit, and thus three solutions (**15a-c**) are obtained.



15a



15b



15c

Using the modified kernel technique, then, one obtains seven solutions (**12, 13a–c,** and **15a–c**) to the labeling of the chosen orbit in case 1 of Table VII.

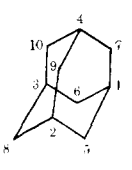
The modifications discussed here can be viewed in a context more general than that of free-valence labeling. In essence, the free-valence labeling is simply a node labeling in which each node receives not just one label, but as many labels as it has free valences. This kind of problem, in which some or all of the parts of a symmetric object receive fixed numbers of labels other than one, occurs in other contexts, most notably in the edge labeling of graphs with one or more multiple edges. The above modifications of the basic algorithm may be transferred directly to such problems if the term "node" is replaced by "part" and "number of free valences on a node" is read as "allowed number of labels on a part." In the edge labeling case, this allowed number corresponds to the multipliciy of the edge.

Part E. Applications of the Algorithm

The labeling algorithm is a powerful tool in determining the scope and limits of many isomerism questions. It has been used,[17] for example, to determine the complete set of Diels–Alder ring systems which can be formed using atoms from the set $C_6N_6S_4O_4$ (along with an appropriate number of hydrogens), and to verify, *via* construction, the fact that there are 13 isomers of the porphyrin ring system substituted with four ethyl and four methyl groups. This latter case is an interesting one because the number of isomers is commonly given as four[18] and has recently been "corrected" to eight.[19] Although it has been noted[20] that the correction is itself erroneous, the actual number (13) has apparently not been published.[22]

In the following, an example is presented which would be difficult to solve without the help of a systematic procedure. The problem is as follows. Given the adamantane skeleton (**16**) with 16 free valences and given, as ligands, 1 hydroxyl group, 2 methyl groups, and 13 hydrogens, construct all topologically distinct molecules. Note that because only topological isomers are to be considered, there is no distinction between enantiomers, nor between structures which differ only in the orientation (*i.e.*, "axial" *vs.* "equatorial") of ligands. To construct these geometrical isomers, one would first number, for reference, the 16 free valences of **16,** then express the 12 spatial rotations (including the identity) which leave **16** unchanged as permutations relative to this numbering, and finally carry out the labeling using the techniques of part B, above. As stated, the problem is somewhat simpler, exemplifying a free-valence labeling as discussed in part D, above. Structure **17** shows the reference numbering[21] to be used for the nodes, and Table VIII gives the topological symmetry group on the nodes relative to this numbering.

16



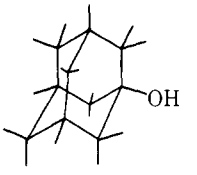
17

Because there are three label types (H, OH, and CH_3), label recursion is necessary. First, the skeleton is labeled with 1 OH and 15 "blanks," an occurrence of special case 1.2; the orbits are (1,2,3,4) and (5,6,7,8,9,10), and associating the single label with the first node of each orbit, one obtains two partial labelings (**18a–b**). In the second step of label recursion, the "blank" free valences of **18a** and **18b**

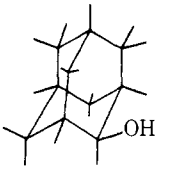
Table VIII. The Symmetry Group for the Nodes of the Adamantane Skeleton (**16**)[a]

Reference no.	Permutation									
1	1	2	3	4	5	6	7	8	9	10
2	1	2	4	3	5	7	6	9	8	10
3	1	3	2	4	6	5	7	8	10	9
4	1	3	4	2	6	7	5	10	8	9
5	1	4	2	3	7	5	6	9	10	8
6	1	4	3	2	7	6	5	10	9	8
7	2	1	3	4	5	8	9	6	7	10
8	2	1	4	3	5	9	8	7	6	10
9	2	3	1	4	8	5	9	6	10	7
10	2	3	4	1	8	9	5	10	6	7
11	2	4	1	3	9	5	8	7	10	6
12	2	4	3	1	9	8	5	10	7	6
13	3	1	2	4	6	8	10	5	7	9
14	3	1	4	2	6	10	8	7	5	9
15	3	2	1	4	8	6	10	5	9	7
16	3	2	4	1	8	10	6	9	5	7
17	3	4	1	2	10	6	8	7	9	5
18	3	4	2	1	10	8	6	9	7	5
19	4	1	2	3	7	9	10	5	6	8
20	4	1	3	2	7	10	9	6	5	8
21	4	2	1	3	9	7	10	5	8	6
22	4	2	3	1	9	10	7	8	5	6
23	4	3	1	2	10	7	9	6	8	5
24	4	3	2	1	10	9	7	8	6	5

[a] The reference numbering here corresponds to **17**.



18a



18b

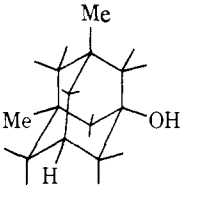
are labeled with 2 CH_3's and 13 H's under the appropriate reduced symmetries. The details are given below.

Case 1. Labeling of 18a. Node 1 is distinguished by the fact that it carries an hydroxyl group, and thus any permutation in Table VIII which does not have "1" as its first entry is no longer valid. This leaves permutations 1 through 6 (Table VIII) in the reduced symmetry group of **18a,** and the new orbits are (2,3,4), (5,6,7), and (8,9,10). Here, as in the rest of this section, orbits are given only for those nodes which are still to receive labels.

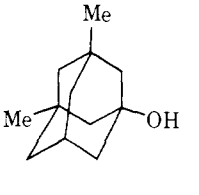
At this stage, there are more than two of each type of label and more than one orbit. None of the special cases apply, and thus orbit recursion is needed. If the orbit (2,3,4), which has three free valences, is chosen, there are three label distributions to be considered.

Distribution 1. (a) The orbit (2,3,4) receives two CH_3's and one H. Special case 1.2 applies, and because there is only one orbit, one partial labeling (**19**) results.

(b) The remaining free valences of **19** receive ten H's. Special case 1.1 applies, and one full labeling (**20**) is obtained. For clarity, hydrogens are omitted in **20** and in all other fully labeled structures in this example.



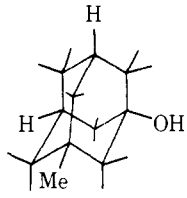
19



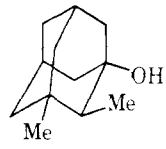
20

Distribution 2. (a) The orbit (2,3,4) is given one CH_3 and two H's. Again, special case 1.2 applies, yielding the partial labeling **21.**

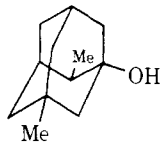
(b) The remaining free valences of **21** receive one CH_3 and nine H's. Special case 1.2 applies. The labeling in (a) reduces the symmetry group to permutations 1 and 2 in Table VIII, and the new orbits are (5), (6,7), (8,9), and (10). Associating the single label with the first node in each, one obtains four full labelings (**22–25**).



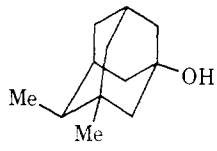
21



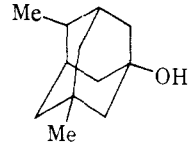
22



23

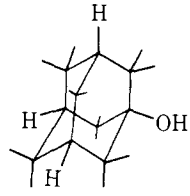


24



25

Distribution 3. (a) The orbit (2,3,4) is labeled with three H's, for which special case 1.1 is used. The partial labeling **26** results.

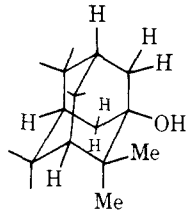


26

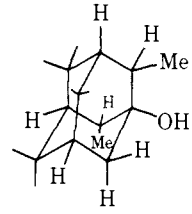
(b) The remaining free valences of **26** are given two CH_3's and eight H's. The labeling in (a) does not reduce the symmetry group, and permutations 1 through 6 are still valid. The new orbits are (5,6,7) and (8,9,10), and because no special cases apply, a second, quite similar level of orbit recursion is necessary.

Distribution 1. (a) Two CH_3's and four H's are placed on (5,6,7). A call to the modified kernel technique is needed, and two partial labelings (**27a–b**) result (see below).

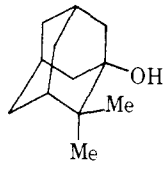
(b) The remaining free valences of **27a** and **b** are given six H's, for which special case 1.1 is used. Two full labelings (**28–29**) result.



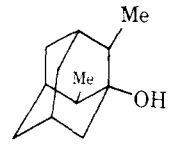
27a



27b



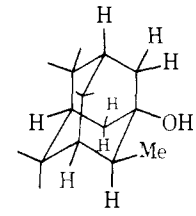
28



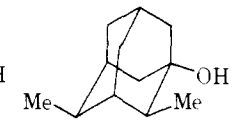
29

Distribution 2. (a) One CH_3 and five H's go onto (5,6,7). Special case 1.2 gives one partial labeling (**30**).

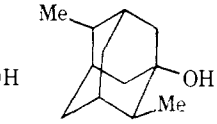
(b) The six free valences of **30** are labeled with one CH_3 and five H's using special case 1.2. Only permutations 1 and 2 of Table VIII are valid for **30**, and the new orbits are (8,9) and (10). Thus two full labelings (**31–32**) result.



30



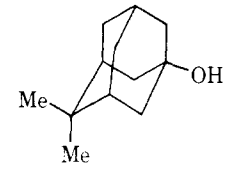
31



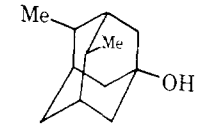
32

Distribution 3. (a) Finally, (5,6,7) is given six H's through special case 1.1.

(b) Two CH_3's and four H's are attached to the six free valences in (8,9,10), which is still an orbit because the labeling in (a) destroys no symmetry. The modified kernel technique gives (see below) two full labelings (**33–34**).



33



34

In the above, two references are made to the modified kernel technique. In each use, the orbit contains three nodes with two free valences apiece, the symmetry group includes all six permutations of these nodes, and the label set consists of two CH_3's and four H's. There are three ways to partition these six labels into three multilabels of degree 2. For each partition, there are two multilabels of one type and one of the other, so special case 1.2 may be used to carry out the node labeling. There is one orbit, so each partition gives one result. Structures **27a** and **33** are derived from partition 1, while **27b** and **34** are obtained from partition 2.

(CH_3,CH_3)	(H,H)	(H,H)	(partition 1)
(CH_3,H)	(CH_3,H)	(H,H)	(partition 2)

Case 2. Labeling of 18b. Here, node 5 bears an OH (along with an unfilled free valence which must eventually be labeled), and only permutations 1, 2, 7, and 8 are still valid. The new orbits are (1,2), (3,4), (5), (6,7,8,9), and (10). The steps in this labeling are analogous to those described in case 1, except that four levels of orbit recursion are needed rather than two. Figure 3 gives a schematic illustration of these steps, which together result in 19 full labelings (structures **35** through **53** in Figure 3). There is one call to the modified kernel technique in which the orbit is (6,7,8,9), with each node bearing two free valences, the symmetry group contains permutations 1, 2, 7, and 8 of Table VIII, and the label set is composed of two CH_3's and four H's. These eight labels may be grouped into four multilabels as follows.

(CH_3,CH_3)	(H,H)	(H,H)	(H,H)	(partition 1)
(CH_3,H)	(CH_3,H)	(H,H)	(H,H)	(partition 2)

Labeling the nodes of the orbit with the multilabels in partition 1 may be accomplished using special case 1.2. For the second partition, the standard kernel technique is needed. The steps parallel exactly those given for the decalin example in part B, section 2.3 (see also Figure 2), with three resulting labelings. In Figure 3, structure **50** is derived from partition 1, while structures **51–53** are derived from partition 2.

In all, there are 30 unique labelings for this sample prob-

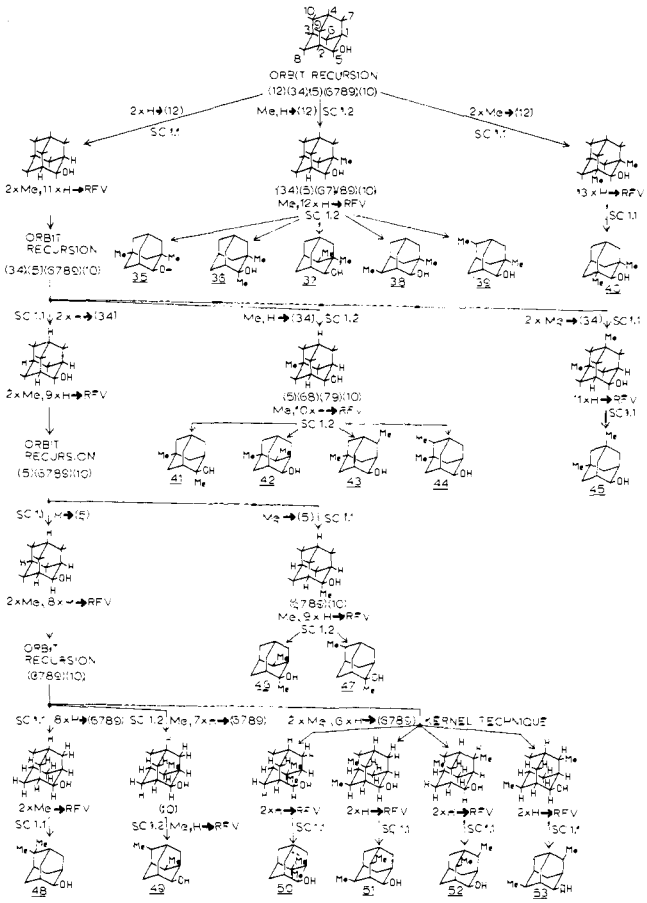


Figure 3. Schematic illustration of the steps in labeling the free valences of **18b** with 2 methyl groups and 13 hydrogens. Each boldface arrow indicates a separate labeling step, with the pertinent label set written on the left and the orbit being labeled written on the right (RFV stands for "remaining free valences"). The abbreviations "SC 1.1" and "SC 1.2" refer to special cases 1.1 and 1.2, respectively (see part B). New orbits, where they are needed, are written below the corresponding partial labelings. For clarity, hydrogens are omitted in the fully labeled structures, which bear the structure numbers **35–53.**

lem: structures **20, 22–25, 28, 29, 31–34,** and (in Figure 3) **35–53.**

Acknowledgments. We gratefully acknowledge the contributions of Larry Hjelmeland who formulated the initial problem, Professor Harold Brown who proved the correctness of the original algorithms, and Professor Joshua Lederberg whose advice and guidance has always been an inspiration.

References and Notes

(1) We gratefully acknowledge the support of this research by the National Institutes of Health (RR 00612-03) and the Advanced Research Projects Agency (SD-183).
(2) Part XII: L. M. Masinter, N. S. Sridharan, J. Lederberg, and D. H. Smith, *J. Amer. Chem. Soc.*, **96,** 7702, (1974).
(3) (a) I. Ugi, D. Marquarding, H. Klusacek, G. Gokel, and P. Gillespie, *Angew. Chem., Int. Ed. Engl.*, **9,** 703(1970); (b) P. Gillespie, P. Hoffman, K. Klusacek, D. Marquarding, S. Pfohl, F. Ramirez, E. A. Tsolis, and I. Ugi, *ibid.*, **10,** 687 (1971).
(4) (a) A. T. Balaban, *Rev. Roum. Chem.*, **18,** 635 (1973) and references therein; (b) A. T. Balaban and F. Hararay, *ibid.*, **12,** 1511 (1967).
(5) (a) W. G. Klemperer, *J. Amer. Chem. Soc.*, **94,** 6940 (1972); (b) *ibid.*, **94,** 8360 (1972); (c) *ibid.*, **95,** 380 (1973); (d) *ibid.*, **95,** 2105 (1973).
(6) (a) W. Hässelbarth and E. Ruch, *Theor. Chim. Acta*, **29,** 259 (1973); (b) V. R. Riemschneider, *Z. Naturforschg. B*, **11,** 228, 291 (1956).
(7) J. Lederberg, "DENDRAL-64, Part I. Notational Algorithm for Tree Structures," NASA Star No. N65-13158, NASA CR-57029; "Part II. Topology of Cyclic Graphs," NASA Star No. N66-14074, NASA CR-68898; "Part III. Complete Chemical Graphs: Embedding Rings in Trees," NASA Star No. N71-76061, NASA CR-123176.
(8) H. Brown, L. M. Masinter, and L. Hjelmeland, *Discrete Math.*, 7, 1 (1974).
(9) (a) N. G. DeBruijn, *Kon. Ned. Akad. Wetensch., Proc., Ser. A*, **62,** 59 (1956); (b) "Applied Combinatorial Mathematics," E. Bechenbach, Ed., Wiley, New York, N.Y., 1964, p 144; (c) see also ref 6–8 of ref 2.
(10) An alternative algorithm has been described to us in a private communication from D. M. Perlman, University of California, San Diego, Calif.
(11) See, *e.g.*, Appendix B of ref 2 for a discussion of various kinds of symmetries which can be considered for molecules.
(12) Orbits can be calculated easily from the table of the symmetry group (*e.g.*, Table III). With the permutations as the rows, the numbers in each column, taken as a set, form an orbit. The orbits of the group in Table III are (1,5,6,10), (2,4,7,9), and (3,8). The nodes in each orbit correspond to symmetrically equivalent nodes.
(13) A *recursive* technique is one which is defined in terms of itself. It is necessary that, at each step, the problem is reduced, and that a terminating condition is eventually met. The general solution to the labeling problem is described in terms of less complicated labelings, and several terminating conditions (special cases) are defined.
(14) The symmetry group of a labeled object is always a subgroup of the symmetry group of the unlabeled object. Thus, one needs to know which permutations in the original group must be discarded. To do this, one writes the symmetry group in tabular form (*e.g.*, Table III), then writes the label associated with each node next to all occurrences of the corresponding node number. If in any column, there is an element which has a different label from the label in the reference numbering (identity permutation), then the row containing that element can be discarded. That is, a permutation of the original group is valid for the labeled structure if and only if it acts upon the representation of that labeling (see part A) to yield the identical representation.
(15) In this section, "call X" means "transfer control to X until a return from X is encountered, then resume processing at the point where the call was issued." Owing to the recursive nature of the algorithm, complex hierarchies of calls are often created during complicated labelings.
(16) As long as one is seeking isomers which differ in topology (rather than geometry), then all free valences upon a node may be considered equivalent. That is, only the *numbers* of various types of ligands upon each node are significant, not the spatial distribution of those ligands. The generalization outlined here is not applicable to cases in which stereochemistry is important.
(17) J. Simek, unpublished results.
(18) See, *e.g.*, H. R. Mahler and E. H. Cardes, "Biological Chemistry," 2nd ed., Harper & Row, New York, N.Y., 1971, p 419.
(19) D. Blackman, *J. Chem. Educ.*, **50,** 258 (1973).
(20) W. H. Eberhardt, *J. Chem. Educ.*, **50,** 728 (1973).
(21) This is not the standard chemical numbering but was chosen to simplify the manual computation of the symmetry group. This group is derived from that of the tetrahedron with nodes 1–4 occupying the vertices. In our computerized implementation of the labeling algorithm, topological symmetry groups of graphs are calculated automatically.
(22) NOTE ADDED IN PROOF. The correct number has been given by R. L. C. Pilgrim, *J. Chem. Educ.*, **51,** 316 (1974).