Page of Drake Report	Compound	Drake Report	This System
	(HO)(HS)PH	phosphonothious acid	thiophosphinedioic acid
29	Cl(MeO)PPh	methyl phenylphosphono-chloridite	methyl chlorophenylphosphinoate or chloromethoxyphenylphosphine, etc.
	H_2NPH_2	phosphinous amide	aminophosphine
	$HSPH_2$	phosphinothious acid	thiophosphinoic acid or mercaptophosphine
29	H_2NPO_2	phosphenic amide	aminodioxophosphane
	$HSPS_2$	phosphenotrithioic acid	mercaptodithiophosphane
	$HOP(NH)_2$	phosphenodiimidic acid	hydroxydiiminophosphane, etc.
30	HNPN	phosphenimidic nitride	iminonitrilophosphane
	CH_3NHPO_2	methylphosphenic amide	(methylamino)dioxophosphane
	$C_4H_9OPO_2$	butyl phosphenate	butoxydioxophosphane
	$CH_3OP(O)NH$	methyl phosphenimidate	iminomethoxyoxophosphane
	H_2NPO	phosphenous amide	aminooxophosphine
	HOPS	phosphenothionous acid	hydroxythiophosphine
	HSPO	phosphenothiolous acid	mercaptooxophosphine
	H_2NPNH	phosphenimidous amide	aminoiminophosphine
	$CH_3OP{=}O$	methyl phosphenite	methoxyoxophosphine
	CH_3NHPNH	*N*-methylphosphenimidous amide	imino(methylamino)phosphine
	CH_3NPNH_2	*N'*-methylphosphenimidous amide	amino(methylimino)phosphine, etc.

Resolution of Ambiguities in the Nomenclature of Spiro Ring Systems*

J. E. RUSH and L. J. WHITE
Department of Computer and Information Science, The Ohio State University, Columbus, Ohio 43210

Received March 2, 1970

Owing to the laxity extant in the existing rules for naming spiro ring systems, it was necessary to characterize unambiguously spiro ring systems before attempting to eliminate other ambiguities. Following this characterization, procedures are given for naming all types of spiro ring systems including those which contain more than two terminal rings and for naming branched spiro ring systems. A procedure for transforming the names into their corresponding graphical representations is also described.

The problems to which this paper is addressed are: eliminating the ambiguity which arises when one tries to transform into graphical representation the name of a spiro ring system which contains more than two terminal rings, providing a method for naming branched spiro ring systems, and providing a procedure for transforming such names into graphical representations. Solution of these problems is necessary to enable the user of indexes such as those to *Chemical Abstracts* to retrieve information on these compounds unambiguously, to improve the chances for successful translations of the names into atom-bond connection tables[1] or into linear notations such as the Wiswesser Line Notation,[2] to improve the chances for successful computer and human transformation of the names into graphical representations,[3] and to provide correct names to which stereochemical notations may be affixed.[4]

This paper is divided essentially into two parts. The first part is a discussion, on a rather intuitive level, of the ambiguities which exist in the rules for naming spiro ring systems. The second part is a mathematical treatment of the subject and is presented in part to illustrate the proper approach to the development of nomenclature rules.

DEFICIENCIES IN EXISTING RULES

The naming of spiro ring systems seems, at first glance, as simple a task as can be imagined within the realm of chemical nomenclature. Naming the systems is, however, only half the battle, since the utility of the

*Presented in part before the Division of Chemical Literature, 157th Meeting, ACS, Minneapolis, Minnesota, April 1969.

names derives from the ability to transform them into graphical representations of the ring systems from which the names were originally produced.

Careful analysis of the IUPAC rule for naming spiro ring systems[5] reveals that it is not really a rule at all, but a guide to the intuitive construction of a name. Definitions within the rule are, at best, poor; no provision whatsoever is made for the naming of branched systems or systems with three or more terminal rings, and no procedure is provided for the transformation of names into graphical representations. These faults alone justify the work described herein.

The portion of the rule which is in question is the following:[5]

> A-41.6 Polyspiro compounds consisting of three or more alicyclic systems are named by placing "dispiro-," "trispiro-" "tetraspiro-," etc., before the name of the normal acylic hydrocarbon of the same total number of carbon atoms. The numbers of carbon atoms linked to the spiro atoms in each ring are indicated in brackets in the same order as the numbering proceeds about the ring. Numbering starts with a ring atom next to a terminal spiro atom and proceeds in such a way as to give the spiro atoms as low numbers as possible.

To be thorough, rule A-41.1, A-41.2, A-41.3, and A-41.6 will subsequently be reformulated. (See statement of the Rule for Naming Spiro Ring Systems.)

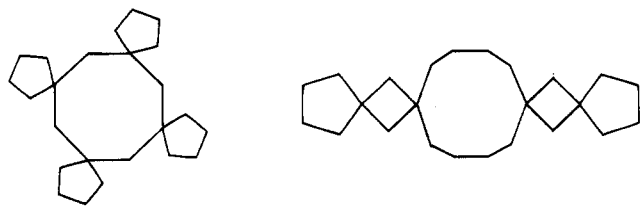
The heart of a name generated under rule A-41 is the bracketed series of numbers, here called the "descriptor string" (see Figure 1 for illustration). For a descriptor string of n numerals (numerals separated by periods):

> The number of spiro atoms is $S_T = n/2$;
>
> The terminal rings (those containing exactly one spiro atom) are the lst and $[(n/2) + 1]$th members of the string, provided the ring system is unbranched and has only two terminal rings;
>
> The total number of atoms in the system is
>
> $$A_T = S_T + \sum_{i=1}^{n} C_i$$
>
> where C_i is the ith cardinal number of the descriptor string.

Before going on to consider the various problems which must be resolved, let us consider a somewhat intuitive definition of spiro ring systems. A spiro ring system consists of at least two circuits (simple closed paths) with no edges in common. Each atom of the ring system is either of degree 2 or of degree 4 (that is, an atom of the ring system is either joined to 2 other atoms or to 4 other atoms). A terminal ring of the system is one which contains only one node of degree 4.

SPIRO RING SYSTEMS WITH MORE THAN TWO TERMINAL RINGS

Tetraspiro[4.1.4.1.4.1.4.1]tetracosane is the name of the following ring systems.



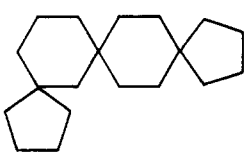
Having derived these possibilities from the one name, it becomes obvious why spiro ring systems containing more than two terminal rings cannot be adequately named according to existing rules.

There are several possible ways in which this difficulty can be overcome. The simplest of these, from all points of view, is to place a bar ($\bar{}$) over each member of the descriptor string which corresponds with the terminal ring. Thus, regardless of the number of terminal rings in a spiro system, all can be easily identified, and transformation of their names into graphical representations can be effected easily and unambiguously.

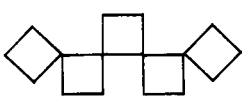
BRANCHED SPIRO RING SYSTEMS

Branched spiro ring systems are not only endowed with more than two terminal rings, but they also have spiro atoms which can only be connected by a branched path (see Figure 2). The existing nomenclature rules are wholly inadequate to handle such systems. To illustrate this point consider the descriptor string for the ring system of Figure 3 (incorporating the change proposed in the preceding section): [$\bar{4}$.1.0.1.$\bar{5}$.2.2.1.0.2.$\bar{4}$.2.0.2.$\bar{4}$.2.2.1.1.3]. From this descriptor string it is impossible to derive, on the basis of known logical principles, any semblance of the ring system illustrated in Figure 3. One first encounters difficulty in the transformation when atom 8 is reached, since nothing in the descriptor string indicates that atoms 8 and 17 are connected. The stepwise partial transformation of the descriptor string into the graph is shown in Figure 4 to make this point clear. Only when the third zero (0) of the descriptor string is reached is one made aware that the transformation has gone awry. Obviously,

Dispiro[5.1.5.2]pentadecane



Trispiro[4.1.2.4.4.2.3]nonadecane



Tetraspiro[3.0.0.0.3.2.2.2]hexadecane

Figure 1. Typical spiro ring systems and their corresponding names (note especially the descriptor strings)
These names have been formed by existing rules

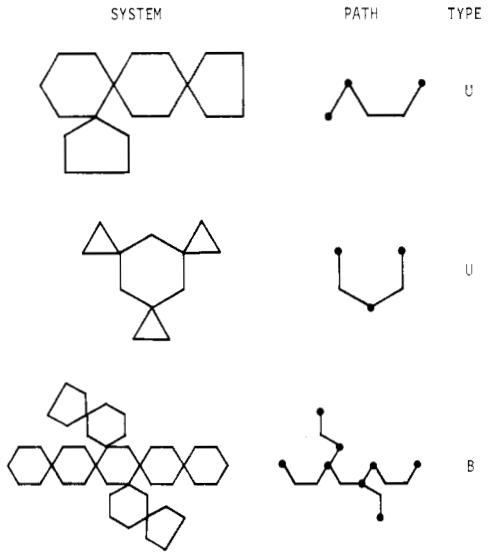


Figure 2. Examples of unbranched (U) and branched (B) spiro ring systems

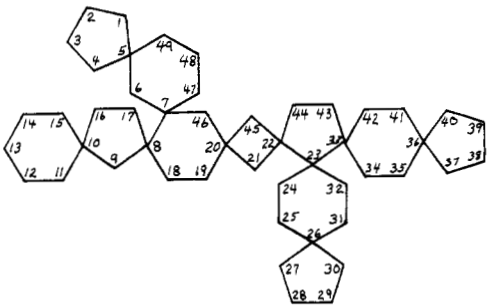


Figure 3. A typical branched spiro ring system which confounds existing nomenclature rules

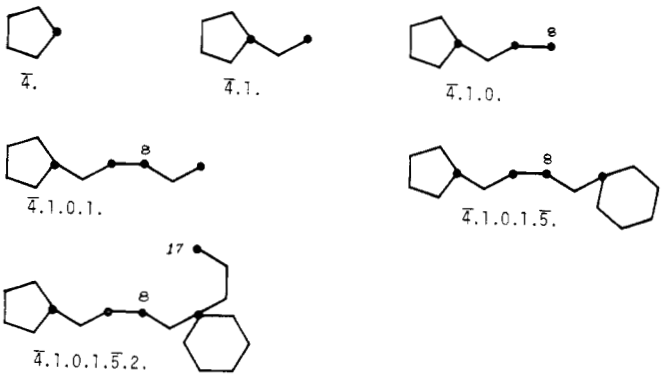


Figure 4. Partial (attempted) reconstruction, from the descriptor string, of the ring system of Figure 3

the descriptor string is inadequate in its present form to permit successful transformation into graphical representation. What is needed is an additional operator, similar to the bar ($\bar{}$) and the period (.), by means of which branches can be identified. The parentheses, long used to denote branching in complex acyclic systems, are proposed for this purpose. Thus, the descriptor string for the branched spiro ring system illustrated in Figure 3 (not necessarily the unique descriptor string) becomes $[\bar{4}.1.0(1.\bar{5}.2)2.1.0(2.\bar{4}.2)0.2.\bar{4}.2.2.1.1.3]$.

One additional point not covered by existing rules relates to the way in which the ring system is numbered. The existing rule[5] states that the graph is to be numbered ". . . .in such a way as to give the spiro atoms as low numbers as possible." For the example of Figure 3, a question arises: Which atom to number 18? If atom 46 had been numbered 18, the spiro atoms would have received lower numbers, in accordance with the rule. But this would have necessitated retracing the path between 8 and 7 causing the addition of a fictitious path (containing no atoms) to the descriptor string, which is clearly not permissible and should therefore be disallowed in the rules. The proposed revised rules given in this paper provide for this difficulty. The numbering rules are also extremely cumbersome in application because of the necessity for trying several starting points and having to do a considerable amount of lookahead, especially in highly symmetrical systems. An alternative procedure for numbering the atoms of a spiro-fused ring system is given in the following section.

CHARACTERIZATION OF SPIRO RING SYSTEMS

Before attempting to state the proposed rule explicitly, it is necessary first to characterize spiro ring systems and to define appropriate terms. In addition, certain principles of graph theory are invoked in order to facilitate the naming and regeneration of structures from these names.

For convenience in this discussion, we will henceforth refer to spiro ring systems as "spiro graphs."[6] Define a spiro graph, G, as a graph such that

There are two types of nodes (atoms): nodes of degree 4, called spiro nodes, and nodes of degree exactly 2;
G is finite and connected;
Two distinct circuits of G have no edges in common;
G contains at least one spiro node.

Figure 5 shows typical spiro graphs in which some of the terminology that has been employed in the above definitions, and in some of those to follow, is illustrated. One additional observation of particular importance in

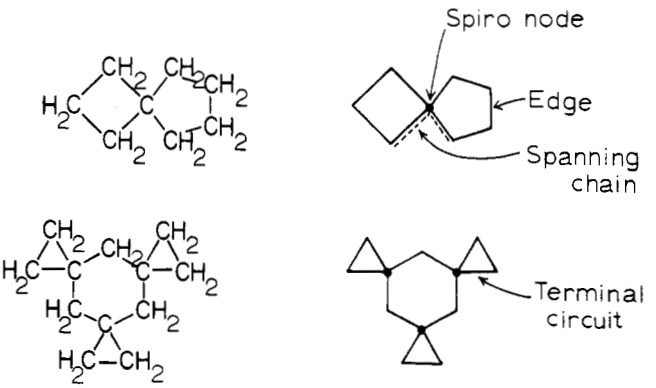


Figure 5. Spiro compounds and corresponding spiro graphs with illustration of some of the terminology used in the text

characterizing spiro graphs is: given a spiro graph, G, there exist at least two terminal circuits. Inspection of a typical spiro graph shows that this statement is intuitively obvious, hence no proof is offered (proofs are given elsewhere[7]).

To simplify the present discussion further as well as to simplify the application of the rules which we are now developing, we here introduce the concept of a reduced spiro graph, which we shall call an S-graph, defined as follows:

Given a spiro graph, G, construct a corresponding reduced spiro graph (S-graph) G^* according to the following procedure:

Remove each terminal circuit from G and label the remaining spiro node from that circuit with a positive integer representing the number of nodes of degree 2 which were removed. Assign the label zero (0) to those spiro nodes which are not contained in terminal circuits.

Contract each simple path which terminates in spiro nodes and which consists otherwise only of nonspiro nodes to a single edge whose end points are the two aforementioned spiro nodes. Assign to this edge a nonnegative integer weight corresponding to the number of nonspiro nodes removed from the original path.

Note that parallel edges may be produced in G^*, even though such edges do not exist in G. The S-graph thus constructed has all of the characteristics of the original spiro graph, with the following distinctions: There is a degenerate case for G^*, which consists of the single node with two positive integer weights but which has no edges, and parallel edges may exist in G^* whereas parallel edges are not allowed in G. G^* may not, however, contain loops. Examples of the transformation of spiro graphs into S-graphs are given in Figure 6. Since S-graphs are fundamental to the remainder of the discussion we offer the following theorem (without proof[7]) as providing a complete characterization of S-graphs.

Theorem 1: A graph G^* with nonnegative node and edge weights is an S-graph of some unique (to isomorphism) spiro graph, G, if the following conditions hold:

G^* is connected and finite and contains at least one node

If G^* contains only one node and no edges, this node has two associated positive weights

If G^* contains more than one node, each node has either degree 2 or degree 4 and has a unique weight; if the node is of degree 2, its weight is positive; if of degree 4 its weight must be zero

Two distinct circuits of G^* have no edge in common

Parallel edges may exist in G^*, but not loops

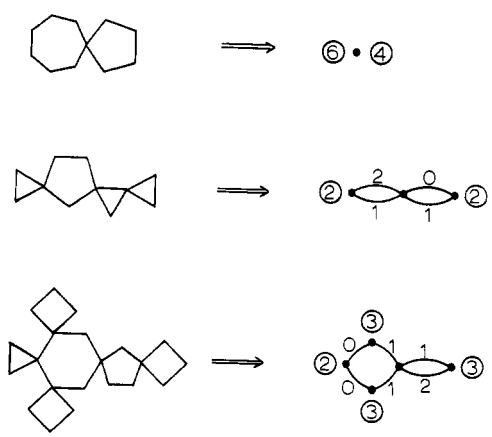


Figure 6. Examples of spiro graphs and their corresponding S-graphs (note: the first example is the degenerate case)

Branched and Unbranched Spiro Graphs. Let us now distinguish between two basic types of spiro graph, called "branched" and "unbranched" spiro graphs. An unbranched spiro graph G is one in which there exists a simple path, P, which includes at least one edge from every circuit of G; call such a path P a "spanning chain" of G. A branched spiro graph is one for which no spanning chain exists. These definitions apply equally well to S-graphs, and it is in this context in which we shall principally employ the definitions. Additionally, we define a "terminal circuit" of an S-graph as a circuit which contains only one node of degree 4. Since G^* was defined so as to exclude loops, this means that a terminal circuit of G^* must contain at least one node of positive weight. Examples of unbranched and branched S-graphs are given in Figure 7. To provide an unequivocal means of identifying branched and unbranched spiro graphs, the following theorem is offered.

Theorem 2: A spiro graph G is branched if and only if its corresponding S-graph G^* contains a circuit which includes more than two nodes of zero weight—i.e., of degree 4.

The above theorem shows that if no such circuit C exists, then a spanning chain P exists in G^*. The identification of P can be accomplished by describing an algorithm which can be applied to an arbitrary S-graph. The algorithm will either find a circuit containing more than two nodes of zero weight, thus showing G^* to be branched, or else will construct a spanning chain, P. (Note that an upper bound on the computation grows linearly with N, where N is the number of nodes in G^*, since each node need be considered at most once.)

Branch Algorithm

1. If no node of zero weight exists in G^*, then G^* consists of a single circuit, and a spanning chain obviously exists.
2. Begin at any node of zero weight v_1 in G^* and search a circuit, not previously examined, containing v_1 to see if

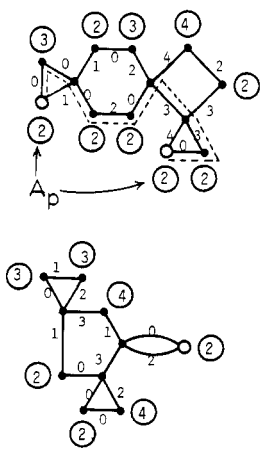


Figure 7. Examples of unbranched (upper) and branched (lower) S-graphs

Node (2) and edge (1) weights are shown as are the preferred starting nodes (A_p) for tracing a spanning chain (---) of the graph (see Rule 4.2 of the proposed rules)

it includes more than two nodes of zero weight. If so, terminate the algorithm, as G^* is branched.

3. If a circuit, C, containing v_1 is a terminal circuit, it defines one end of the spanning chain, P. Return to the node v_1 of zero weight initially selected and consider unexplored circuits by repeating step 2. Terminate the algorithm when both ends of the spanning chain are found.

4. If C is not a terminal circuit, identify $v_2 \neq v_1$ as the other node of zero weight in C. Extend P by adjoining either path between v_1 and v_2 within C.

5. Redefine v_2 as v_1 and return to step 2.

The course of this algorithm is depicted in Figure 8, for both a branched and an unbranched S-graph.

CONSTRUCTION OF THE DESCRIPTOR STRING

Having now provided unequivocal means of distinguishing between branched and unbranched spiro graphs, there remains the task of constructing the descriptor string for the spiro graph. Under the existing rule, the construction of the descriptor string requires that the nodes of the spiro graph G be labeled with nonnegative integers. This rule states that the graph is to be numbered "....in such a way as to give the spiro atoms as low numbers as possible." Such a labeling procedure is extremely inefficient, and in highly symmetrical spiro graphs the number of elementary steps required to label the nodes of the graph successfully may be on the order of $N!$ and in most spiro graphs will require at least on the order of N^k steps where k is a positive integer whose value will depend upon the exact structure of the graph and on the values of the edge or node weights. To overcome this inefficiency, we will shortly give an improved labeling algorithm for spiro graphs. First, however, let us demonstrate the general way in which the descriptor string is to be constructed.

Consider an Euler tour of a given S-graph G^* which begins at a node v_1 of positive weight, and define a descriptor string D as follows. Starting at v_1, construct D by entering positive node weights (with bars over the integers) and all edge weights as the Euler tour is traversed. Since v_1 may be any node of G^*, it is necessary, in order to construct a canonical form of the descriptor string, to have the capability of defining a unique starting point

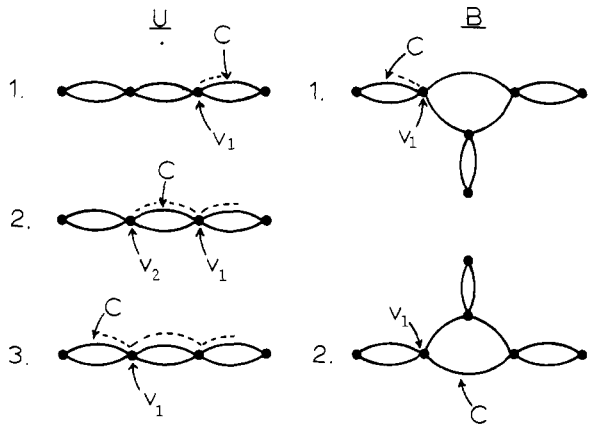


Figure 8. Course of the branch algorithm for an unbranched (U) and a branched (B) S-graph

Note that a spanning chain (---) is found in U but not in B

and a unique method of tracing the Euler tour for each S-graph. To achieve these goals, we must first define a partial ordering of the descriptor strings. Consider two descriptor strings, D_1,D_2, with $2N$ entries of nonnegative integers (with bars over some entries). Then D_1 is smaller than D_2, denoted $D_1 < D_2$, if the integer entries of D_1 are smaller lexicographically than those of D_2 and where by convention $\bar{k} > k$, for integer k.

LABELING SPIRO GRAPHS

Let us now deal with the labeling of unbranched spiro graphs, and then we will subsequently generalize this to deal with any arbitrary spiro graph. Since an unbranched S-graph contains at most two terminal circuits, we can identify four classes of unbranched spiro graphs, the nature of which depend upon the nature of the terminal circuits.

I. G^* contains no nodes of zero weight and hence no terminal circuits.

II. Both terminal circuits of G^* contain more than one node of positive weight.

III. One terminal circuit of G^* contains exactly one node of positive weight and the other terminal circuit contains more than one node of positive weight.

IV. Both terminal circuits of G^* contain exactly one node of positive weight.

In spite of its apparent simplicity, case I provides the worst case for the improved algorithm to be described, since the number of steps required is potentially larger than for any other case. Nevertheless, when the branch algorithm described previously is applied to an S-graph of this type, it is easily determined that G^* is of case I when the spanning chain is obtained.

Unbranched Labeling Algorithm. Consider the following algorithm to obtain a unique descriptor string D for a spiro graph G whose corresponding S-graph G^* is of type I.

Consider the subset X of all nodes of G^* which have a minimum weight.

Construct Euler tours from each node of X, one in each direction around the circuits of G^*. Generate descriptor strings for each Euler tour.

Eliminate the larger descriptor strings D (in the sense defined above), until a unique smallest string D is generated. There may be several Euler paths which correspond with this descriptor string D.

It is clear that the upper bound on the growth of this algorithm is on the order of N^2 and, therefore, does not offer any significant advantage over the labeling procedure given in the original rule. Recall, however, that this is the worst case for the improved algorithm described here. Note also that it can be shown that the descriptor string generated by the above algorithm is in one-to-one correspondence with G^* and hence with the spiro graph G.

Now let us consider an algorithm for labeling the nodes of the remaining three types of S-graph depicted above.

1. Consider the two terminal circuits C_1,C_2 obtained from the branch algorithm described previously.

2. If C_1 contains only one node v_1 of positive weight, then go to step 3. Otherwise consider the two nodes of positive weight u_1,u_2 which are adjacent in C_1 to the node of zero weight. Starting from u_1 and u_2, construct two descriptor

strings D_1' and D_2' by traversing C_1 in such a way as to encounter all nodes of positive weight in C_1, entering node weights and edge weights in alternating fashion until the node of zero weight of C_1 is encountered in each string. If one string is less than the other, choose its corresponding starting node as v_1 and therefore as an endpoint of the spanning chain P. If the two strings are identical at this point, arbitrarily choose the distinguished node v_1 from u_1,u_2.

3. If circuit C_2 contains only one node v_2 of positive weight, go on to step 4. Otherwise repeat step 2 for C_2 in order to determine the other endpoint of P. Then go to step 4. (Note v_1 and v_2 have been found by entirely local operations, each involving only one circuit of an arbitrarily complex unbranched spiro graph.)

4. Beginning at v_1, construct D_1 by entering positive node weights (with bars above them) and all edge weights as the spanning chain P is traversed. As a new circuit is encountered, extend P by selecting the path within the circuit between the two nodes of zero weight, such that the constructed string D_1 is the smallest. When the two paths between these two nodes of zero weight lead to identical terms being entered into D_1, then either path may be included in P.

5. When v_2 is encountered, continue tracing the unique Euler tour and constructing D_1, terminating at v_1 when D_1 contains $2N$ entries.

6. Repeat steps 4 and 5 with the roles of v_1 and v_2 interchanged, constructing string D_2. Let D be the smaller of D_1,D_2.

This algorithm grows only linearly with N and the following theorem shows another advantage of the algorithm.

Theorem 3: For a descriptor string, D, constructed by the above labeling algorithm, two unbranched spiro graphs possess the same descriptor string if and only if they are isomorphic.

Generalized Labeling Algorithm. As we pointed out earlier, it is possible to generalize the algorithm for labeling unbranched spiro graphs to include any arbitrary spiro graph. Such a generalized labeling algorithm is presented below.

To follow more easily the course of the algorithm, reference may be made to Figure 9, in which the algorithm has been applied to the partial labeling of the graph illustrated.

1. Find all terminal circuits $C_1,C_2,\ldots,C_k$ in an S-graph, G^*, that is, all circuits which contain only one node of zero weight.

2. Identify the one or two nodes in each circuit as distinguished, which are incident to the node of zero weight.

3. Locally, in each circuit C_i $(i = 1,2,\ldots,k)$ identify a single distinguished node v_i. If there are two candidates, begin with each in turn and construct descriptor strings D_1 and D_2' by traversing the terminal circuit C_i in such a way as to encounter all nodes of positive weight in C_i, entering node weights and edge weights in alternating fashion until the node of zero weight of C_i is encountered in each string. If one string is less than the other, choose its corresponding starting node as v_i. If strings D_1' and D_2' are identical, arbitrarily choose the starting node of one of the strings as the distinguished node v_i.

4. For every terminal circuit C_i of G^*, perform steps 4 through 6. Construct a simple path P_i as follows:

   a. Extend P_i in C_i by beginning at v_i and traversing C_i in such a way as to encounter all nodes of positive weight in C_i, until a node of zero weight is encountered. Call this node v_0.

   b. If v_0 is contained in a circuit having more than two

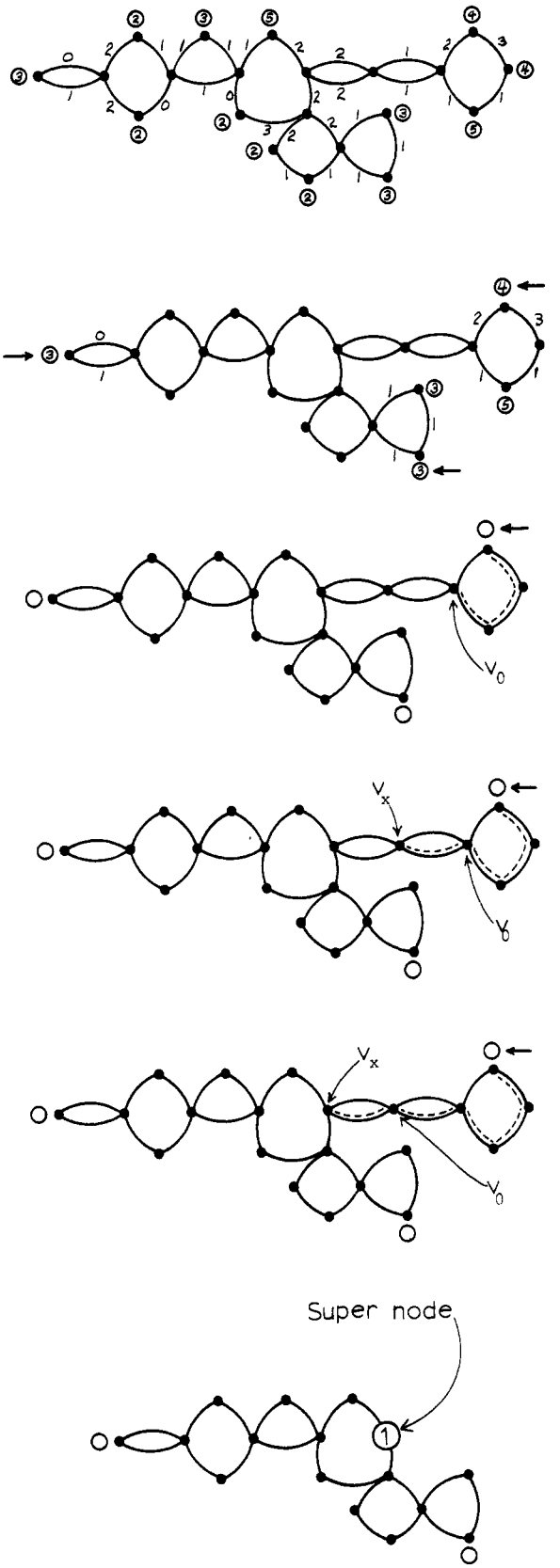


Figure 9. Partial labeling of a branched S-graph by means of the generalized labeling algorithm. One branch (of 2) is reduced to a super node (1)

The descriptor string (D_1) for this branch is [$\bar{4}$.3.$\bar{4}$.1.$\bar{5}$.1.1.2*2.1.2] and the string, L_1 is [2.1.1.$\bar{5}$.1.$\bar{4}$.3.$\bar{4}$.2.1.2]

nodes of zero weight, identify this first node of zero weight encountered along P_i as v_x ($v_x = v_0$) and go to step 5; otherwise go to step 4c.

c. For this new circuit identify $v_0' \neq v_0$ as the other node of zero weight of the circuit. Extend P_i by adjoining v_0 and v_0' by the path having the smaller string in this circuit. If the two paths between v_0 and v_0' are identical, arbitrarily choose one. Although strings may not be of the same length, obtain a linear ordering of them in the sense of "smallest string," according to the following conventions. If two strings are equal in the linear ordering, but are of different lengths, let us agree that the string of smaller length is also smallest in the sense of the linear ordering.

d. Redefine v_0' as v_0 and return to step 4b.

5. Define a branch, B_i, of G^* corresponding to terminal circuit C_i, as consisting of all the nodes in the constructed path P_i and of all edges of G^* whose endpoints occur in this node set. This B_i is a subgraph of G^*. Define P_i as the spanning chain of B_i.

6. Construct a string D_i for branch B_i by beginning at node v_i and traversing path P_i until the terminal node v_x is encountered. At this point, complete D_i by completing the Euler tour from v_x to v_i in B_i. Identify the point in D_i at which v_x was encountered.

7. When steps 4 through 6 have been completed for all terminal circuits in G^* select a "largest string" and consider the corresponding branch, B_i, in step 8. If several branches correspond to largest strings, then apply step 8 to all of these branches, in any order.

8. For convenience in subsequent processing of the spiro graph, the branch, B_i, should be thought of as being contracted to a single "super node." The two edges incident to node v_x which are not in B_i should now be incident to this super node, i.e., the node v_x, of zero weight, which terminated the spanning chain P_i for B_i, is replaced by this super node. This super node is now considered as a node of positive weight and is marked so as to identify it as corresponding to B_i. For further convenience at later stages of the algorithm, we define a new string, L_i, formed as follows. As B_i is contracted, and beginning at the position of string D_i identified as the point at which node v_x occurred, make entries in L_i by reading D_i backward from this point; when the beginning of D_i is reached go to the end of D_i and continue reading D_i backward, making corresponding entries in L_i in a forward manner. This operation will yield an L_i which begins at v_x of B_i, traverses the spanning chain P_i backwards to v_i, and then completes the induced Euler tour, ending with v_x. Note that L_i has exactly as many entries as D_i.

9. As subsequent contractions of branches to super nodes are performed, a sequence of S-graphs ($G_0 = G^*$, G_1, G_2,. . . .) is formed. In determining the linear ordering of strings, let super nodes be ordered by the index of the graph, G_j, in which they first appear. Continue steps 7 and 8 with the new graph containing super nodes. After a branch (or several branches) has (have) been contracted to a single super node (or several super nodes), determine whether a new circuit containing a single node of zero weight is formed. If so, consider a new branch, and develop a new spanning chain and descriptor string as indicated in steps 4 through 6.

10. Terminate this process at the first graph G_j of the sequence which is unbranched. Establish a spanning chain for G_j in the manner described in the Unbranched Labeling Algorithm section and construct a descriptor string D_j for G_j. However, when a super node is encountered in this string, expand this super node, entering the associated string, L_i. Place parentheses around substring L_i within string D. If L_i should contain a super node, replace this entry with its corresponding string, also set off by parentheses. When all super nodes contained in L_i, or in its generated substrings, have been expanded, return to graph G_j and continue tracing the spanning chain to construct D.

11. Continue this construction until all edges of G_j, and hence all edges of the original S-graph, G^*, have been traversed by an Euler tour. The descriptor string D thus generated is unique.

Finally, it can be shown that in spite of the apparent complexity of this algorithm, the growth of the construction is only on the order of N^2, where N is the number of nodes in the S-graph.

It only remains to show that the descriptor string produced by this algorithm is one-to-one with an arbitrary S-graph G^* and therefore with an arbitrary spiro graph G. We here offer only the theorem and not the proof.[7]

Theorem 4: Given a string D, constructed by the generalized labeling algorithm, two spiro graphs possess the same string D if and only if they are isomorphic.

THE RECONSTRUCTION ALGORITHM

Once the unique descriptor string D has been constructed according to the algorithm just described, the original spiro graph G corresponding to this descriptor string can be reconstructed by means of the following algorithm (see also Figure 10). Note that although this algorithm applies only to unbranched spiro graphs, branched spiro graphs can also be unambiguously recon-

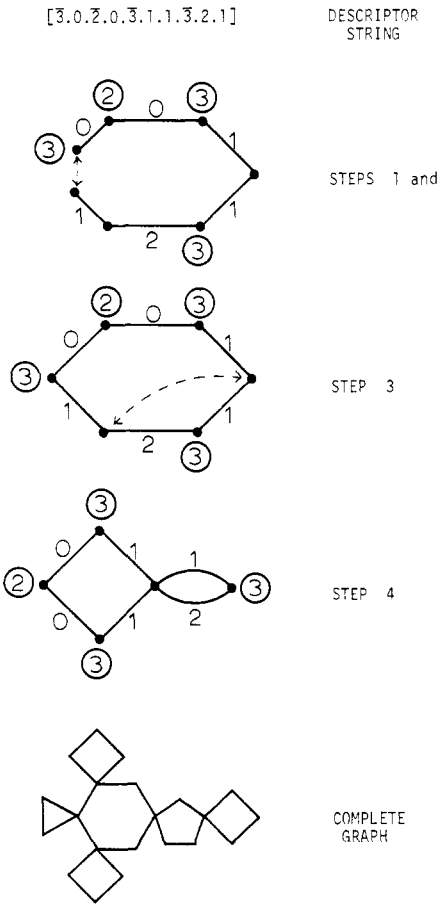


Figure 10. Course of the reconstruction algorithm

structed via this algorithm through the simple mechanism of an n-step procedure in which the branches are treated as single nodes in the first step, leaving an unbranched spiro graph to be reconstructed. The algorithm is then reapplied, independently to each branch. The number of steps n required for the complete reconstruction will be equal to the maximum level of nesting within D.

1. Identify and mark the node corresponding with the first barred entry—i.e., the first entry—of the descriptor string D. Proceed through the string to the right until all $2N$ entries have been considered.

2. Draw a simple path, beginning with this distinguished node weighted with its corresponding barred integer from D. Adjoin this node to the next node in the path by the edge corresponding to the unbarred integer in D and weight the edge with this integer and the new node zero (0), or else weight this new node with the barred integer from D. When a branch is encountered (a portion of D enclosed within parentheses), treat this branch as a single specially marked node in the constructed path and proceed through D.

3. When this simple path is complete (entries of D have been exhausted), identify the initial and terminal nodes of the path as coincident. The form of the drawn structure is now a closed path.

4. Starting at this node (point of coincidence) move along the closed path in the two directions emanating from this node and terminate in each direction when a node of zero weight is encountered in that portion of the closed path. Identify these two nodes of zero weight as being coincident and reestablish their incidence so that the degree of the resultant node is four.

5. Continue step 4 by moving in the two directions, from the newly formed node, along edges not yet examined, making nodes coincident as in step 5. Terminate this process when the two subpaths meet and all edges have been traversed.

6. If any branches were marked in step 2, apply steps 2 through 5 to each such branch (parenthesized list in D) in any order. Otherwise the reconstruction process is completed by expanding node and edge weights into terminal circuits or edges and nonspiro nodes.

STATEMENT OF THE RULE FOR NAMING SPIRO-FUSED RING SYSTEMS

(Replaces Rules A-41.1, A-41.2, A-41.3 and A-41.6)

1. This rule applies to ring systems which have the following characteristics: the ring system is a graph which is finite and connected, in which the atoms are either of degree two or of degree four (in the latter case called spiro atoms) with respect to attachments (bonds) within the ring system, in which two distinct rings (circuits) of the system have no edges in common, and in which the ring system contains at least one spiro atom. Define a terminal ring as a ring which contains only one spiro atom.

2. The names for such spiro ring systems are formed according to the following general formula: {Hetero atom stems}{multiplicative}spiro[{descriptor string}]{name of normal acyclic hydrocarbon}{bond descriptors}{functional suffixes} (variables are enclosed in braces; underscored variables must always be present in the name). (Complete explication of such components as "hetero atoms stems" would, of course, be given in a complete set of nomenclature rules. We cannot enter into a discussion of these components in this paper.)

3. Define an unbranched spiro ring system as one in which there exists a spanning chain. A branched spiro ring system is defined as one in which no such spanning chain exists. To determine whether a spanning chain exists (particularly in complex cases) employ the following procedure.

3.1. Transform the ring system into a reduced spiro ring system (hereinafter called "reduced system") as follows:

3.1.1. Eliminate all terminal rings; assign to the spiro atom of each terminal ring a numerical weight equal to the number of atoms of degree two eliminated in that ring.

3.1.2. Transform all other paths between spiro atoms to single edges ("bonds"), assigning to each such edge a numerical weight (edge weight) equal to the number of atoms eliminated from the original path.

3.1.3. Assign to spiro atoms not in terminal rings a weight of zero.

3.2. If there is no atom of zero weight in the reduced system, the system (and hence its parent system) is unbranched.

3.3. If a ring of the reduced system contains more than two atoms of zero weight, the system (and hence its parent system) is branched.

4. Form the descriptor string according to the following procedure, noting, in the process, the path taken through the system.

4.1. Within the reduced system formed in 3.1, locate all terminal rings (rings which contain only one atom of zero weight).

4.2. Within each terminal ring identify (mark) as preferred an atom adjacent to the atom of zero weight. Call this atom A_p. If there are two such atoms, select one as preferred as follows:

4.2.1. Begin with each atom in turn and construct partial descriptor strings by traversing the terminal ring in such a way that all atoms of positive weight are encountered, entering atom and edge weights into the string alternately, until the atom of zero weight is encountered in each string.

4.2.2. If one string is smaller than the other the starting atom of this string is preferred.

4.2.3. If the two strings are identical, arbitrarily give preference to the starting atom of one of them.

4.3. Perform steps 4.3 through 4.5 for every terminal ring of the reduced system. (If the reduced system is unbranched, omit the steps involving branches, that is, go to 4.9.1.)

4.3.1. Construct a simple path through the reduced system, beginning at the preferred atom, A_p (as identified in 4.2), and traversing the ring until an atom of zero weight is encountered. Call this atom A_1. If A_p is the only positively weighted atom in this ring, traverse the edge of lower weight (in case of a tie, arbitrarily choose an edge). Otherwise traverse the ring in such a way that all nodes of positive weight are included on the path.

4.3.2. If the other ring containing A_1 is a ring having more than two atoms of zero weight, rename A_1 as A_x and go on to step 4.4. If A_1 is the only atom of zero weight in the ring, this identifies the terminus of the spanning chain, so go to step 4.9.2. Otherwise go to step 4.3.3.

4.3.3. For this other ring, call the second atom of zero weight A_2 ($A_2 \neq A_1$). Extend the path along the route from A_1 to A_2 which has the smaller string. If the two routes from A_1 to A_2 are identical, arbitrarily choose one (mark the choice).

4.3.4. Now redefine A_2 as A_1 and return to 4.3.2.

4.4. Define a branch of the reduced system, corresponding to a particular terminal ring, as consisting of all of the atoms in the constructed path together with all bonds between these atoms (both ends of a bond must join atoms on the path). Define the constructed path as the spanning chain P_i of this branch.

4.5. Construct a descriptor string for the branch by beginning with the preferred atom, A_p, in the terminal ring of the branch and entering atom and edge weights alternately as the path P_i is traversed. When A_x is encountered, complete the descriptor string by traversing edges not on P_i, ending at A_p. Mark the position in the descriptor string where A_x occurred.

4.6. Once descriptor strings have been constructed for all branches, select a branch corresponding to the largest string and go to step 4.7. If more than one branch corresponds to the largest string, then apply step 4.7 to all such branches, in any order.

4.7. For the branch(es) having the largest descriptor string(s):

4.7.1. Contract each such branch to a single super node. Assign identical weights to all such super nodes. The value of the weight (1,2,3,. . . .) corresponds with the first, second, third, etc., application of step 4.7.

4.7.2. Transform the descriptor string D for this branch into a new string L as follows: invert the elements of D, from the beginning of D to A_x, and enter them into L; complete L by entering the remaining elements of D in inverted fashion. (A string of k elements is inverted by exchanging the first with the kth, the second with the $(k-1)$th, and so on.)

4.8. After all branches have been contracted to super nodes, determine whether any new ring is formed which contains a single atom of zero weight. If so, consider a new branch, and develop a new spanning chain and descriptor string, as indicated in steps 4.3 through 4.5.

4.9. Terminate this process (4.8) as soon as the reduced system has been contracted to a system which is unbranched (no ring contains more than two nodes of zero weight).

4.9.1. Establish a spanning chain for this unbranched system as prescribed in step 4.3.

4.9.2. Construct a descriptor string for the unbranched system as prescribed in step 4.5, but ignoring references to A_x. However, when a super node is encountered in this string, replace it with its associated string L, enclosing L within parentheses. If L should also contain a super node, replace it with its associated string, also enclosed within parentheses.

4.9.3. When all super nodes contained in L or in its generated substrings have been replaced, return to the unbranched system and continue tracing the spanning chain to complete the descriptor string.

4.10. Continue this construction until all edges of the unbranched system, and hence all edges of the original spiro ring system have been traversed by an Euler tour. The resulting descriptor string uniquely and unambiguously represents the spiro ring system from which it was derived.

5. Number the atoms of the original spiro ring system as follows (these numbers serve as locants for the placement of substituents and other structural features):

5.1. Locate the terminal ring which corresponds the the first barred (i.e., first) entry in the descriptor string constructed in step 4. Select as atom number 1, one of the two atoms adjacent to the spiro atom in this ring. Wherever a choice for the assignment of a number exists, give preference first to noncarbon atoms, then to atoms bearing substituents, then to atoms linked by multiple bonds (within the ring), then to ionic charge, then to isotopic mass, and finally to stereochemical features. (In a complete set of nomenclature rules, full explication of these precedence rules would be given, but such rules are beyond the scope of this paper.) If a choice still exists, break it arbitrarily.

5.2. Number the remaining atoms in this terminal ring sequentially from 1 (i.e., 2,3,4,. . . .), the spiro atom being numbered last.

5.3. Number the atoms in the remainder of the spiro ring system in a like manner, proceeding around the system by following the Euler tour prescribed by the descriptor string, and applying the precedence rules of 5.1 wherever necessary.

Figure 11 illustrates the application of rule 5.

6. Drawing the structure from the name can be readily accomplished as follows:

6.1. Draw an atom corresponding with the first barred entry (i.e., first entry) of the descriptor string and weight it with this barred integer.

6.2. Proceeding to the right through the descriptor string, draw a simple path from the first atom to the next, adjoining these two atoms by an edge corresponding with the unbarred integer in the descriptor string. Weight the edge with this unbarred integer and the new atom zero (0), or else weight this new atom with the barred integer from the descriptor string. When a branch is encountered (signified by a parenthesized list within the descriptor string) treat this branch as a single, specially marked atom in the constructed path. The branch is otherwise ignored until 6.6.

6.3. When the simple path is complete (entries from the descriptor string have been exhausted) make the first and last atoms of the path coincident.

6.4. Starting at this initial point of coincidence, move along the closed path in the two directions from this atom, and stop when an atom of zero weight is encountered in that portion of the closed path. Make these two atoms coincident (result: a single atom of degree four).

6.5. Continue to apply step 6.4 by moving in the two directions from the newly formed atom along edges not yet

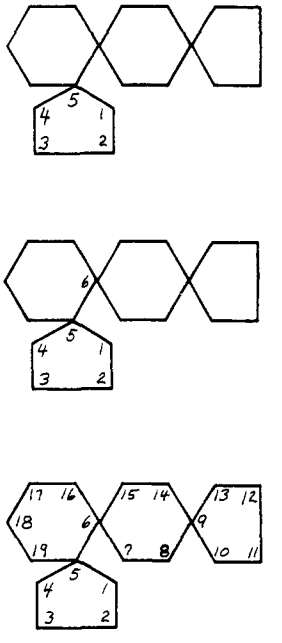


Figure 11. Illustration of the numbering rule (Rule 5)

traversed. Terminate this process when the two subpaths meet and all edges have been traversed.

6.6. If any branches were identified in 6.2, apply steps 6.2 through 6.5 to each such branch in any order.

6.7. Complete the reconstruction process by expanding the atom and edge weights into terminal rings or sequences of edges and nonspiro atoms.

CONCLUSION

We have shown that it is possible to derive nomenclature rules which produce unique and unambiguous names for chemical structures. While the short-range goal of this work was to remove the ambiguities which exist in the nomenclature of spiro ring systems, the work serves as a model for the development of a complete nomenclature of chemical structures. In addition, the contribution to graph theory is clear.[8]

ACKNOWLEDGMENT

We wish to express our appreciation to G. G. Vander Stouw for pointing out the ambiguity in existing nomen clature rules. Support of this work by the National Science Foundation, through a grant (GN 534.1) to the Computer and Information Science Research Center, The Ohio State University, is also gratefully acknowledged.

LITERATURE CITED

(1) Vander Stouw, G. G., I. Naznitsky, and J. E. Rush, "Procedures for Converting Systematic Names of Organic Compounds into Atom-Bond Connection Tables," J. CHEM. Doc. 7, 165–9 (1967).

(2) Smith, E. G., "The Wiswesser Line-Formula Chemical Notation," McGraw-Hill, New York, 1968.

(3) Long, P. L., and J. E. Rush, "An Algorithm for the Identification and Characterization of Cyclic Graphs Contained in Connection Matrixes," Abstracts of Papers, CHLT-21, 157th Meeting, ACS, Minneapolis, Minnesota, April 1969.

(4) Petrarca, A. E., J. E. Rush, and J. E. Blackwood, "Stereonomenclature IV. A Simple Method for Unambiguous Specification of Configurational Stereoisomerism of Ring Systems Based on a Group Theoretical Approach," Abstracts of Papers, CHLT-23, 157th Meeting, ACS, Minneapolis, Minnesota, April 1969.

(5) International Union of Pure and Applied Chemistry (I.U.P.A.C.), "Nomenclature of Organic Chemistry," 2nd ed., Butterworth, 1966.

(6) Busacker, R. G., and T. L. Saaty, "Finite Graphs and Networks: An Introduction with Applications," McGraw-Hill, New York, 1965.

(7) White, L. J., and J. E. Rush, "Linear Lists for Spiro Graphs," Tech. Rep. No. 69-6, The Computer and Information Science Research Center, The Ohio State University, Columbus, Ohio, 1969.

(8) White, L. J., and J. E. Rush, "Linear Lists for Spiro Graphs," Proceedings of the Calgary International Conference on Combinatorial Structures and their Applications, Calgary, Alberta, June 1969 in "Combinatorial Structures and Their Applications," R. Guy, H. Hanani, N. Sauer, and J. Schonheim, Eds., pp. 473–75, Gordon and Breach, New York, 1970.

An Optical Coincidence System for Personal Literature References

A. F. M. BARTON
Chemistry Department, Victoria University of Wellington, Wellington, New Zealand

Received February 11, 1970

A reference and indexing feature card system suitable for individual literature research files is described.

Abstract cards are frequently used for personal research references, but often they are not capable of being used to full advantage for reviewing a subject or writing a paper. As the number of cards increases, the system becomes more unwieldy, leading to neglect and eventual chaos. Presented here is a description of a simple system of indexing which overcomes many of the difficulties. For a personal reference system it is not necessary to go to the lengths involved in a full literature survey facility. The articles have all been read, or at least seen, and a system is adequate if it enables a vaguely remembered paper to be found.

CODE INDEXING SYSTEM

The "Code Numbers" define the subject matter in terms of a feature card system such as an optical coincidence method.[1,2,3] The Butterworths Research and Indexing Kit (B.R.I.K.) peep-hole index cards (Figure 1) enable 3000 reference numbers to be indexed. Each subject category has an index card, in which holes corresponding to the reference or card numbers are punched. The peep-hole system enables specific subjects to be defined by a combination of index cards. For example, to find articles dealing with pressure vessel design, the index cards for high pressure and for apparatus would be selected, and the common abstract reference numbers found by superimposing the index cards and holding them up to the light. A careful choice of subject categories is necessary to allow expansion of any subject as one's research activities deepen and the emphasis changes. The aim is to have the minimum number of subject categories for each particular article, while restricting the total number of subject categories. One method of achieving this is to eliminate