(3) Sussenguth, E. H. "A Graph-Theoretic Algorithm for Matching Chemical Structures". *J. Chem. Doc.* **1965**, *5*, 36-43.
(4) Figueras, J. "Substructure Search by Set Reduction". *J. Chem. Doc.* **1972**, *12*, 237-244.
(5) Dittmar, P. G.; Farmer, N. A.; Fisanick, W.; Haines, R. C.; Mockus, J. "The CAS ONLINE Search System. 1. General System Design and Selection, Generation and Use of Search Screens". *J. Chem. Inf. Comput. Sci.* **1983**, *13*, 93-102.
(6) Wipke, W. T.; Rogers, D. "Artificial Intelligence in Organic Synthesis. SST: Starting Material Selection Strategies. An Application of Superstructure Search". *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 71-81.
(7) Willett, P. "The Calculation of Inter-Molecular Similarity Coefficients Using an Inverted File Algorithm". *Anal. Chim. Acta* **1982**, *138*, 339-342.
(8) Adamson, G. W.; Bush, J. A. "A Comparison of the Performance of Some Similarity and Dissimilarity Measures in the Automatic Classification of Chemical Structures". *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 55-58.
(9) Feldman, A.; Hodes, L. "An Efficient Design for Chemical Structure Searching. I. The Screens". *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 147-152.
(10) Willett, P. "The Effect of Screen Set Size on Retrieval from Chemical Substructure Search Systems". *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 253-255.

A New System for the Designation of Chemical Compounds. 2. Coding of Cyclic Compounds[†]

RONALD C. READ

Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Canada N2L 3G1

Received September 1, 1983

In this paper a procedure is given for coding cyclic compounds, that is, deriving a unique designation for any such compound. The method described in part 1 for coding acyclic compounds is first used to detect and remove all side chains from the molecule, their codes being recorded for later use. The remainder of the molecule, which, by definition, is taken to be the ring structure, is then coded by first classifying its atoms and then constructing a special "walk" within the structure. This leads to a unique and concise designation for the ring structure and a standard numbering of its atoms. This numbering enables the locations of the side chains to be precisely specified. The resulting designation for the whole compound has something in common with some existing systems of nomenclature in so far as it specifies first the side chains and then the ring structure to which they are attached. Unlike other systems, however, the coding process does not require the use of lists of ring structures, such as those in the *Parent Compound Handbook*; the designation can be computed in full from the structural formula of the compound or equivalent information. The procedure is very amenable to automatic computation and has already been implemented by a Fortran program of no great length.

(1) INTRODUCTION

Part 1 of this paper[8] described a system for coding acyclic chemical compounds, the main feature of which was an algorithm that computed a unique code or "name" for any acyclic compound by means of operations performed on the structural formula of the molecule being coded (or on anything equivalent to a structural formula, such as a connection table). These operations were purely graph theoretical, making no call on chemical knowledge or intuition, and the resulting code was made up of the customary symbols used in organic chemistry (atom symbols, symbols for bonds, etc.) carrying their usual meanings. Moreover, the code was in a form that was meaningful to a chemist, either immediately or with only a small amount of paperwork.

Typical examples of codes produced by the algorithm are $CH(CH_3)_2(CH_2.OH)$, $C(CH_3)_3((CH_2)_2.CH_3)$, and $C(CH{:}CH_2)(=NH).CH_2.C(CI_3){:}CH.CH(CH_3)_2$

In this paper the more difficult problem of coding cyclic compounds is tackled. The main objectives are as before: to produce a coding algorithm that is automatic, that does not require chemical intuition (and which, therefore, can be easily programmed on a computer), and that produces a code that is at least partly intelligible at sight and is easily decoded in full, even by hand. Naturally, it must meet the basic requirement of any system of nomenclature that is to be used for information retrieval, namely, that each structural formula must give rise to a unique code, no matter in what form the formula is originally presented to the algorithm. These requirements have been met in the present system.

Although the question was discussed in part 1, it would be well to reiterate here the two main ways in which this system is an improvement on existing systems. First, it gives a finite set of rules from which the designation corresponding to *any* structural formula can be derived, without the need to consult any book of reference such as the *Parent Compound Handbook*. This desirable property is, to be sure, shared by a few existing systems—the one due to Morgan[7] is a good example. But these systems are designed for use in a purely computerized environment, and the designations that they produce are not readily intelligible to the chemist. By contrast, designations produced by the present system give a great deal of information about the compound in a readily visible form. The nature of the side chains is clear from the first part of the designation. The tail of the designation, which expresses the ring structure, is more opaque but can easily be decoded, by hand, to yield the ring structure and its canonical numbering.

The general idea just described, that of producing a "name" made up of components representing the ring structure of the molecule and the side chains attached to it, is one that is found in certain other nomenclature systems. In these systems, however, the ring structure is designated by a trivial name, which has to be looked up in, say, the *Parent Compound Handbook*.[2] In the present system, the name for the ring structure can be computed automatically from the structural formula without any lookup process. (Needless to say, the name that this automatic procedure produces is not a name in the sense of something that can be pronounced; it is a string

[†] The research for this paper was supported by Grant A8142 from the National Science and Engineering Research Council of Canada.

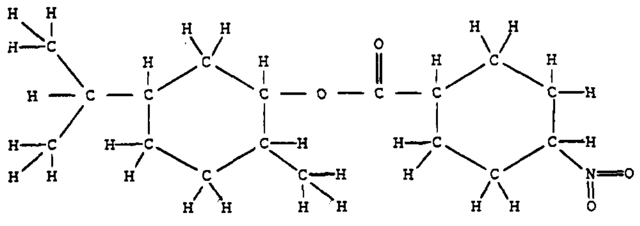


Figure 1.

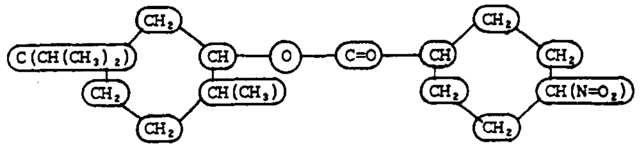


Figure 2.

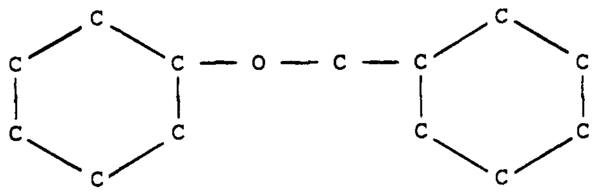


Figure 3.

of symbols—letters, digits, and some punctuation. For this reason the term "code" will henceforth be used for the designation that the system produces.)

The system presented here can be completely automated. As was mentioned in part 1, it has been programmed in Fortran for an IBM 370 at the University of Waterloo and has also been implemented on a PDP11/20 computer. The program handles both cyclic and acyclic compounds, and because of its automatic nature, it can cope with *any* structural formula, no matter how unusual, and thus will not need to be updated or modified as more recondite structures are encountered.

It will be assumed that the reader is already familiar with the coding procedure for acyclic compounds, as given in part 1. I have attempted to use a vocabulary congenial to chemists, but since the problem of chemical nomenclature is fundamentally a graph-theoretical problem, some graph theory terminology has inevitable intruded. In particular, "adjacent" atoms are atoms between which there is a bond (of any kind), and the terms "path" and "walk" are used in their graph-theoretical senses (though with a modification that is explained in the case of walk).

(2) DETERMINATION OF THE RING STRUCTURE

We now turn to the coding of cyclic compounds. We can expect this to be a difficult problem, since it is a variation on the problem of graph isomorphism, which is notoriously intractable[4] (see Discussion in part 1). Before tackling the difficult part of this problem, let us first see how far we can get by applying the procedure for coding acyclic compounds, as given in part 1. To this end, consider the structural formula of Figure 1, in which no abbreviations have been used.

Applying the algorithm for coding acyclic compounds to such a formula, we see that the various side chains of the molecule will be coded to become what were called clusters in part 1. When all the side chains have thus been dealt with, the algorithm will be unable to continue, since there will be no clusters of degree 1. The compound of Figure 1 will then appear as in Figure 2 where each cluster has been circled. If we now omit from each of these clusters all but the first symbol, we obtain the "ring structure" for the molecule, as shown in Figure 3. Now the ring structure of a molecule is of considerable importance since molecules with the same ring structure form families of compounds tending to have some properties in common. Thus, it is certainly advantageous, even if not absolutely necessary, for a coding system to have the property that one can immediately tell from the codes of two compounds whether or not they have the same ring structure. Effectively, this means that a good coding system needs to display the ring structure as an explicit part of the code, and this, in turn, implies that the rest of the code gives information about the nature of the side chains that are attached to the atoms that make up the ring structure.

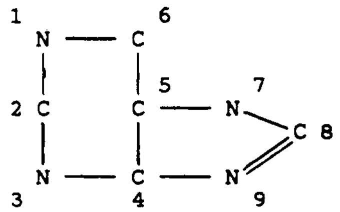


Figure 4.

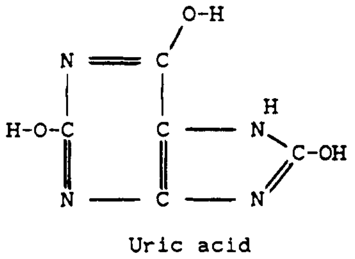


Figure 5.

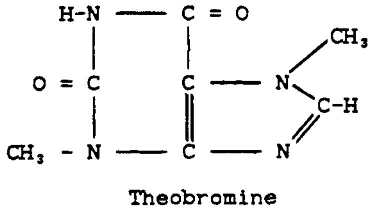


Figure 6.

This is already the practice in many systems of chemical nomenclature. For example, the compounds referred to as the purines have the ring structure shown in Figure 4. We can regard the suffix "-purine" as being a code (albeit an arbitrary one) that denotes this ring structure, with the atoms numbered as shown. Note that this numbering of the atoms, though one which, by convention, is widely accepted, is nevertheless an arbitrary numbering—one that is not deducible from the ring structure alone.

The compound with the trivial name "uric acid" can be referred to as 2,6,8-trihydroxypurine, a name that contains information on the nature of the side chains (three OH groups) and where they are located in terms of the standard numbering of the atoms. Thus a chemist, given the ring structure corresponding to the suffix -purine and knowing the standard numbering for it, can deduce from the above name the structural formula for the compound (Figure 5). Similarly, the compound "theobromine" can be described as 3,7-dimethyl-2,6-dihydroxypurine. A more complete designation is 3,7-dihydro-3,7-dimethyl-1*H*-purine-2,6-dione.[1] In either case, the name gives enough information to determine the structural formula (Figure 6) once the ring structure and its numbering are known. The common element -purine in these names make it immediately apparent that the compounds have the same ring structure.

This method of naming compounds is in frequent use and is well-known. It is reviewed here to draw attention to the fact that it provides a convenient basis for a completely general system of chemical nomenclature. In fact, the whole purpose of this paper is to develop this basis and its general principles into a system that can be applied to any compound, but with the difference that the names and numbering are not arbitrary but are unambiguously deduced from the structural formulas themselves.

There are four requirements for such a system: (1) The ring structure of the compound must be identified. (2) The ring structure must be given a unique code, to serve as the ring parent name at the end of the code of the compound. (3) The atoms in the ring structure must be numbered in some unique way. (4) For each atom of the ring structure the nature of the side chain rooted on that atom must be expressed or implied.

In order to identify the ring structure of a compound we must have a precise definition of what "ring structure" means. The term is not always precisely defined in existing nomenclature systems. Thus, the compound theobromine is sometimes referred to as 3,7-dimethylxanthine (see, for example, p 287 of reference 3), thereby relating it to another structure, "xanthine", rather than to purine as before. This association of two ring structures with a single compound is not conducive to an unambiguous system of nomenclature. Hence, a unique ring structure must be determined for each compound, and we have already specified a procedure for doing this. The coding algorithm for acyclic compounds is applied and terminates when all side chains have become clusters. The structure obtained by keeping only the roots of these clusters and the bonds between them will, *by definition*, be taken as the ring structure of the compound. It will often differ from what is at present regarded as the ring structure.

There is thus no great difficulty in identifying the ring structure of a compound, in the sense that we shall use this term, and since this identification is performed concurrently with the coding of the side chains, the unique codes of these side chains are available when required, as in point 4 above. Thus, the only remaining task is that of deriving a unique code for the ring structure and a unique standard numbering of its atoms.

In the system of nomenclature described above, the name given to a ring structure is arbitrary and can be found only by looking up the structure in the *Parent Compound Handbook*[2] or some similar work of reference. There is no direct way of *computing* that the structure of Figure 4 is called purine. In the same way, the numbering of the atoms can be determined only by looking up the structure in a book of reference. Now apart from the fact that this look-up process is tedious to do by hand and difficult to program for a computer, there is also the unfortunate fact that the *Parent Compound Handbook* lists only what it lists; if one comes across a compound whose ring structure is not in the handbook (and this is something that can be expected to happen more and more as the range of chemical compounds widens), then the handbook is of no help in naming the compound.

By contrast, the procedure given in the following section will associate a unique code with any ring structure whatsoever and also gives a unique method of numbering its atoms. Once this is done, the code for the whole compound can be constructed.

The general appearance of the code of a cyclic compound is as follows: $X^{(1)}$; $X^{(2)}$; ...; $X^{(k)}$; R, where R denotes the string of symbols that is the computed code of the ring structure. Each "prefix" $X^{(i)}$ to the ring structure code is a symbol string of the form: n_1, n_2, ..., n_r–S, where S is the code for a side chain and n_1, n_2, ..., n_r are the numbers of the atoms at which that side chain occurs.

An example will illustrate the general idea. Take the compound theobromine again (Figure 6). In the system being described the code displays all side chains consisting of more than one atom (i.e., consisting of more than the root alone). For this compound therefore the side chains are NH, C=O, CH, and $N(CH_3)$. The ring structure is as given in Figure 7 (where it will be noticed that a double bond has been drawn single, for reasons given in the next section). The algorithm that will be given in section 5 gives, as the code for this ring structure, the symbol string "C3NCN-1NCN-2", and this code implies the atom numbering shown in Figure 7. The complete computed name for the compound is therefore 8-CH;4-NH;3,5-C=O;6,9-$N(CH_3)$;C3NCN-1NCN-2.

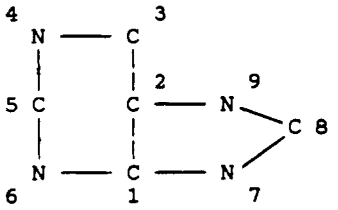


Figure 7.

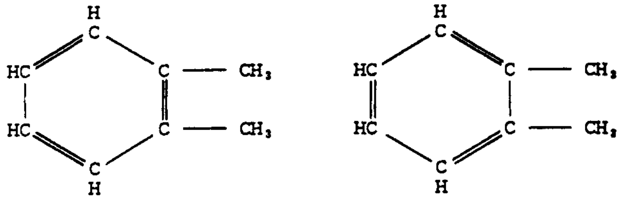


Figure 8.

Comparing this with a standard name for the compound, for example, 3,7-dimethyl-2,6-dihydroxypurine, we see that despite many differences in coding and punctuation the component parts in the two names are similar and are used in similar ways. Naturally, it is a consequence of a fully computerized system of this kind that the names of these component parts, whether of the side chains or of the ring structure, are mere symbol strings, rather than pronounceable words replete with etymological and other associations; but, it could hardly be otherwise. In so far as the proposed system is intended to be used largely for computerized information retrieval, however, this is of little consequence; no doubt to a computer C3NCN-1NCN-2 is just as euphonious as purine!

(3) QUESTION OF MULTIPLE BONDS

Two important matters must be discussed before we consider the method for coding the ring structure of a molecule. The first is that of what to do about double or triple bonds occurring in the ring structure.

In the coding of acyclic compounds the distinction between bonds of different multiplicities was retained throughout the coding process, but this approach leads to problems when dealing with cyclic compounds because of "resonance". This is a well-known phenomenon, exemplified by the two structural formulas in Figure 8. As depicted there, these two formulas are *different* and would be coded differently by any system that took account of the distinction between single and double bonds at all stages of the coding process, for the two carbon atoms having an attached methyl group are joined by a double bond in one case and a single bond in the other. Yet chemical theory asserts that there is, in fact, no difference between them, that the six bonds in a benzene ring are all equivalent. How are we to cope with this very common situation?

One method would be to introduce extra types of bonds, for example a $^2/_3$-fold bond, so to speak, for benzene rings. This would be a substantial step in the direction of greater complexity and, moreover, would be only the first such step, as has been observed elsewhere. Thus, the following types of bonds would all need to be recognized if we take this path to

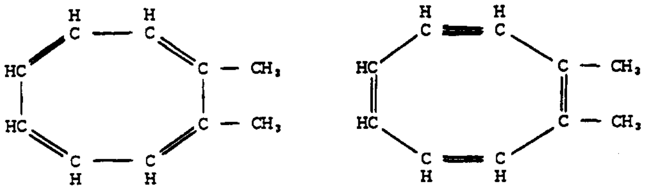


Figure 9.

a solution of the problem (I quote from Figueras in reference 5): (1) single, (2) double, (3) triple, (4) benzenoid, (5) tautomer, and (6) charge delocalized.

It is quite possible that a system of nomenclature could be evolved that made all these distinctions, but it would certainly be complicated and difficult to implement. Instead, we shall follow a path that leads in the direction of lesser complexity and cut the Gordian knot by stipulating that we shall regard *all* bonds occurring in the ring structure as if they were single bonds. This may seem drastic, but there are precedents for it. Thus, the ring structures in Figures 5 and 6 differ in the arrangement of single and double bonds, yet both compounds are regarded as derivatives of purine. This indicates that the distinction between single and double bonds, if not entirely ignored, is not always regarded as of paramount importance in defining the ring structure. The usual way of depicting the benzene ring shows a similar, though not identical, convention.

An obvious consequence of this "single-bond" convention is that when the code for a compound is decoded (as described in section 7), the nature of the bonds in the ring structure is not determined. Frequently, however, the actual bonding can be precisely deduced from the normal valencies of the atoms. Moreover, in most cases where two or more bondings are possible, as with benzene for example, the bondings can be regarded as equally valid, or we can assert that the division of bonds into single or double is not meaningful. In either case, the compound has been properly identified.

Note however the two compounds in Figure 9. Because the phenomenon of resonance is different for eight-membered rings, these two compounds are chemically distinct, but the present method of coding will not distinguish between them. This shows that under certain exception circumstances a code may identify a small group of closely related formulas instead of designating a unique compound. This is at most a minor inconvenience. The decoding process (as will be seen) provides a unique numbering for the atoms of the ring structure, and this numbering can be used to specify unambiguously any further information needed to distinguish between the compounds in such a group.

In summary, the adoption of the single-bond convention leads to a radical simplification of the whole coding process. This more than offsets its one slight disadvantage, which is easily coped with.

(4) MATTER OF RANKS

The other matter that must be settled before describing the coding procedure is that of the "rank" that will be assigned to each atom in a formula being coded. In order to discuss this we must anticipate slightly the contents of the next section, but it will be better to do this than to embark on a lengthy diversion in the middle of the description of the coding process.

The general principle of the coding process is similar to that described in an early paper by Gordon, Kendall, and Davison[6] in that it proceeds by constructing a walk that covers the ring structure. The precise way in which this is done differs greatly from that described in reference 6, but an important feature of both methods is the use of an ordering that is defined over the atoms. In reference 6 this concept is referred to as "seniority" or "priority"; we shall use the shorter term rank. The concept of the rank of an atom, although not a deep one, is important in the coding process and needs to be explained.

The constructiong of a walk in the ring structure is carried out in a step by step manner; we start at some atom, go to an adjacent atom, then go to another atom, and so on, going usually to an atom adjacent to the last atom in the walk so far constructed. If there are several adjacent atoms, then we go to the one with lowest rank. If there are two or more adjacent atoms of lowest rank, then we have a choice of how to continue the walk, and until such time as the ambiguity can be resolved, we have to allow for all the possible continuations. Thus at each stage of the coding process we shall normally have in hand a number of possible walks. Each of these walks will be extended by a further step, if possible, and since there may be several choices for the next atom in the walk, the number of walks in hand may well increase.

We naturally wish to keep this proliferation of walks to a minimum or at least within reasonable bounds, and a careful choice of how the ranks are allocated can be instrumental in achieving this aim. Gordon, Kendall, and Davison,[6] allocate ranks mainly on the basis of coordination number and atomic number. This is a static scheme, fixed once and for all. A more effective way to minimize the number of walks is to use a dynamic scheme, that is, one that depends on the ring structure being coded, thereby taking advantage of some of the statistical characteristics of the ring structure.

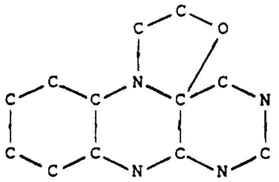
As an illustration of this, consider the method of starting the walks that are constructed. To do this we choose an atom with the higest priority (lowest rank, i.e., rank 1); if there are several atoms of lowest rank, then we have to start a walk at each one of them. Now it may well happen that a static scheme for allocating ranks will give the lowest rank to an atom that is quite common in the ring structure, in which case we have a multiplicity of walks right from the start. On the other hand, the method given below for determining ranks operates on the principle of giving the lowest ranks to atoms that occur least frequently in the particular ring structure being coded. Thus, if there is some kind of atom that occurs only once in the ring structure, there will be only one atom with rank 1, guaranteeing that we start off with a unique walk, whatever may happen later.

To rank the atoms of the ring structure, we therefore divide them up into a number of classes and allocate the lower ranks to the atoms in the smaller classes. Here we must avoid two extremes. Too broad a classification of the atoms (say by merely distinguishing different elements) will not cut down walk proliferation very much. On the other hand, to achieve a division into a large number of small classes might require so much more complex programming as to outweigh the advantages accruing from the finer classification. Clearly, some compromise is called for, and the rank allocation procedure adopted here is one that seems to fall reasonably between the two extremes.

Allocating the Ranks. Ranks are calculated on the basis of just two properties of the atoms in the ring structure: chemical nature, i.e., what element they are atoms of, and "ring degree". The first needs no explanation; the second needs to be defined.

The term "degree" was introduced in part 1 to denote the number of other atoms to which an atom is adjacent. In other words, the degree is what the valency becomes if we treat all bonds as if they were single bonds. The ring degree is a similar concept within the ring structure, being defined as the number of other atoms of the ring structure to which an atom of the ring structure is adjacent. Thus in the benzene molecule, each carbon atom has valency 4, but its degree is 3, and its ring degree is 2.

To calculate the ranks of the atoms of the ring structure, we first classify them by ring degree and element and form a table that shows how many atoms there are in each of the



Ring degree	C	N	O
2	8	3	1
3	3	1	0
4	1	0	0

Element

Figure 10.

resulting classes. Figure 10 shows a typical example, arising from the ring structure shown. The rows in the table correspond to ring degrees in increasing order (note that a ring degree cannot be 1) and the columns to the various elements represented. The columns are arranged in alphabetical order of the customary chemical symbols for the elements.

The classes of atoms corresponding to the nonzero entries in this table are then ranked according to the following rules: (Ranking rule 1) Small classes have lower rank than larger classes. (Ranking rule 2) Among classes of the same size, those with smaller ring degree have lower rank than those with larger ring degree. (Ranking rule 3) Among classes of the same size and same ring degree, the ranking is by alphabetical order of the atom symbol.

There is a simple method of implementing these rules when the table has been constructed. We allocate ranks 1, 2, 3, ... to the classes, working from the smallest to the largest classes, and when we have several classes of the same size, we allocate the ranks to them in the order in which we come across them if we "read" the table in the usual manner, i.e., left to right across the rows and from the top row to the bottom row.

Thus, the classes for the table given above will be ranked as follows, the classes being specified by their row and column symbols:

class	2,O	3,N	4,C	2,N	3,C	2,C
rank	1	2	3	4	5	6

Every atom of the ring structure will belong to exactly one class and, hence, will be given a rank—the rank of its class. This ranking of the atoms plays an important part in the coding process.

(5) CODING PROCESS

We now consider informally the algorithm for obtaining the code of the ring structure of a compound. We start with a drawing of the ring structure or something equivalent to it, such as a connection table (or "adjacency matrix", in graph-theoretical terminology), and we assume that the atoms of the ring structure have been labeled in some arbitrary way for purposes of reference. If the algorithm is being carried out on a computer, then some such labeling will be implicit in the way in which the ring structure is stored in the computer. We can assume, without losing generality, that these arbitrary labels are positive integers. We shall also assume that the ranks of the atoms have been evaluated as outlined in the last section. In general, there will be several atoms in each rank class.

We now construct in the ring structure a number of what we shall call walks, although the sense of that word will be a little different from what one would normally understand by it. A walk in this sense starts at a certain atom, goes to an adjacent atom, then goes to a third atom adjacent to the second, and so on; it may come back to an atom that has already appeared in the walk but will never make the same step twice; i.e., it never goes over a bond more than once. Where these walks differ from walks in the graph-theoretical sense (or even in the colloquial sense) is that they will occasionally break off and continue by "jumping" to some quite different atom, i.e., by continuing at an atom that is not adjacent to the previous one. Thus, a more accurate term for them would be something like "interrupted walks", but we shall need to refer to them so often that the shorter term is preferable.

The algorithm begins by constructing walks in this sense. In general, several walks will be under construction at any given time; they will be called "the walks in hand". Some of these will be abandoned as the algorithm progresses, while others will persist until the algorithm finishes. Each walk that is still extant at the end will "cover" all the bonds in the ring structure; i.e., every bond will occur exactly once as a step from one atom in the walk to its successor.

We first choose an atom at which to start the walk, and for this we choose an atom of lowest rank (rank 1). If there is more than one such atom, then we start several walks, one for each, since there is no way at this stage of choosing one of these atoms in preference to the others.

We continue each walk by making a step to an adjacent atom—one of lowest rank possible. Again, if there is more than one such atom, we must consider each possibility and thus increase the number of walks currently in hand. At a general stage we shall have a number of walks, each starting at an "initial atom" and ending at a "terminal atom", i.e., the one last added to the walk. The usual way to continue each walk is to extend it to an atom of lowest rank adjacent to the terminal atom, as just described. This will be called a "normal extension" of the walk.

Under some circumstances we depart from the process of normal extension. If it is possible to extend the walk by going to an atom already on the walk, then we do so. In other words, if the terminal atom is adjacent to an atom on the walk, then we step to this atom, thus "closing" a portion of the walk to form a circuit or ring. If the terminal atom is adjacent to several atoms on the walk, then the step is made to the one most recently added, i.e., the one that would be encountered first if we retraced our steps along the walk. This has the effect of making the ring that is closed as small as possible. Note that the circuit formed by this operation of ring closure may not be what a chemist would regard as a ring in the usual sense, but it will be convenient to use the shorter term in what follows.

This "closure rule", as it will be called, takes precedence over the rule for normal extension. Note that this manner of extending the walk does not depend on any consideration of ranks and that the extension is unique, so that there is no increase in the number of walks.

When a walk has been extended by closure, we have to determine how to continue it. It may be possible to continue normally from the terminal atom—the atom that was already on the walk; if so, then we do so. Frequently this is not possible, since at least two bonds (more usually three) at the terminal atom have now been used in the walk and there may not be any more (a bond may not be used more than once). In that case, we invoke the "backtrack rule" whereby we backtrack along the walk from the terminal atom until we first come to an atom from which it *is* possible to go along an unused bond. This atom becomes the new terminal atom, and the walk continues from it. This step is, so to speak, a jump and is not regarded as using up a bond, even if the ends of the jump happen to be adjacent (which will be so if the backtracking takes us back exactly one step).

We have seen that we usually need to handle several walks simultaneously and that their number may increase. Fortunately, there is provision in the algorithm, the "conformity rule", which reduces the number of walks whenever a suitable opportunity arises. Informally, this is done by retaining only these walks that are "best", in a sense to be defined.

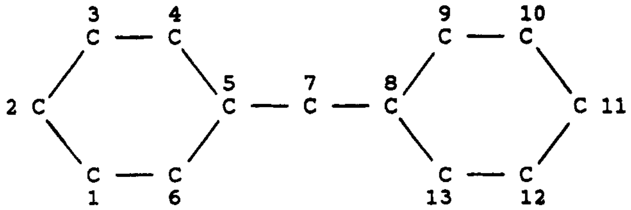


Figure 11.

A walk can be completely described by specifying the labels of the successive atoms that make up the walk, and we shall define the length of a walk to be the number of these labels. Note that, because of closure and backtracking, the same label may well occur several times on a walk. Suppose that we have in hand several walks all of the same length, say k, and we extend all of them by normal extension. In some cases we may end up with two or more walks where we had only one before, but whatever happens all the resulting walks will have length $k + 1$. It helps to imagine that the algorithm proceeds in "stages" and that at each stage all the walks in hand are extended simultaneously (though in practice, of course, they are treated one at a time). It then follows that the walks that we have in hand at the end of each stage will be of the same length.

Associated with a walk there is a sequence made up of the ranks of the successive atoms in the walk. This "rank sequence" must start with "1", since the initial atom of the walk, by definition, has rank 1, but how the sequence continues after that will depend on the walk. Thus, the rank sequence for a walk might be "1, 2, 3, 3, 2, 4".

Now suppose that, when we extend the first walk, we find that we can step to an atom with rank 2, while the second walk can only be extended to an atom of rank 3. We can then distuishing between these two walks; we say that the first walk is "better" than the second and reject the latter. More generally, after all the walks have been extended, we reject any walks whose (new) terminal atom has higher rank than that of the terminal atom of some other walk. Thus, for the two walks mentioned above, the new rank sequences will be "1, 2, 3, 3, 2, 4, 2" and "1, 2, 3, 3, 2, 4, 3". The difference in the last rank enables us to reject the second walk.

In a similar vein, a walk that is extended by ring closure is better than one extended by normal extension, and ring closure to an atom near to the terminal atom is better than that to an atom further back along the walk. In all these case the confirmity rule requires that all but the best walks be rejected.

It is now easy to verify that, after each extension of the walk, whether by normal extension, ring closure, or backtracking, the walks that survive this rejection procedure must all have the same rank sequence. For if two sequences were different, one of the corresponding walks would be rejected.

Examples of Walk Construction. To see how the algorithm works, consider the ring structure of Figure 11 in which the labels attached to the atoms are allocated arbitrarily and are purely for reference purposes. It will be well to digress first to consider the "philosophy" behind the present coding system. Confronted with a structure such as Figure 11, a chemist might well be tempted to decompose it still further and say that what we really have here is a pair of six-membered rings connected by an acyclic component. This temptation is a natural one but should be resisted. The reason is that the ring structure, as defined in section 2, is *easy* to compute. To break down a ring structure further may well give fragments that are more easily analyzed individually, but the computer program that does this (and which will also have to combine the information about the fragments into a single code) will certainly be much more complex. There is a grave danger that this increase in complexity may well outweight any advantages gained. In my opinion even the simple and natural decomposition of Figure 11 into three components would be, at best, a marginal improvement, and to keep the coding process simple, nothing of this kind has been attempted. Moreover, in retrospect, after experimentation with the computer program, it does not appear to have been necessary.

Thus, Figure 11 is something which, by definition, is a ring structure. This means, for example, that atom 7, again by definition, is a ring atom, even though it belongs to nothing that the chemist would normally call a ring. In particular, its ring degree is 2.

Since there is only one kind of atom in Figure 11 the ranks will depend only on the ring degrees. In this compound atoms 5 and 8 have ring degree 3 as distinct from the others that have ring degree 2. Since they are less numerous, atoms 5 and 8 are given rank 1 while the others have rank 2.

Thus, we start at these lowest rank atoms and construct two walks with initial atoms 5 and 8. From neither of these can we go along a bond to an atom of rank 1, so the walks extend to atoms of rank 2—three possibilities in each case. Thus, we have six walks of length 2, namely, "5 7", "5 4", "5 6", "8 7", "8 9", and "8 13".

We now extend these six walks. The walk "5 7" will extend to atom 8 since this has rank 1. The walk "5 4", on the other hand, can extend only to atom number 3, which has rank 2. Thus, the resulting walk "5 4 3" has rank sequence "1, 2, 2" and is therefore less good than the "5 7 8" walk already constructed, which has rank sequence "1, 2, 1". Thus, it is rejected by the conformity rule. So is the walk "5 6 1", which is the only possible extension of "5 6". Continuing in this way we see that only two walks survive, namely, "5 7 8" and "8 7 5", each of which has rank sequence "1, 2, 1".

The next iteration gives four walks since each walk has two continuations. They are "5 7 8 9", "5 7 8 13", "8 7 5 4", and "8 7 5 6" with rank sequence "1, 2, 1, 2".

Note. Looking at Figure 11, anyone can "see" that these four walks are essentially the same, because of the symmetry of the ring structure. A person using the algorithm by hand might well take a natural short cut by dropping three of the walks and continuing with one only. To achieve this economy on a computer, however, we would have to program it to perceive any symmetry in the ring structure and take advantage of it, and this would be difficult. Hence, we continue to work with all the walks that cannot be distinguished by their rank sequences. We shall see that the algorithm does eventually get round to "perceiving" the symmetries of the ring structure, though in its own, rather roundabout, way.

The next four steps of the algorithm are normal extensions and give the following four walks: "5 7 8 9 10 11 12 13", "5 7 8 13 12 11 10 9", "8 7 5 4 3 2 1 6", and "8 7 5 6 1 2 3 4".

We now meet the possibility of ring closure. From atom 13 in the first walk we can close a ring by going to atom 8. In this example there are no other possible extensions from atom 13, but if there were, we would use the conformity rule to reject any that did not result in ring closure. Similarly, we would reject any other walk that could not be extended by ring closure, but in the present case each of the walks can be so extended.

Instead of a space, we shall use a hyphen between the labels when ring closure takes place. Thus, the first walk above extends, by ring closure, to the walk written "5 7 8 9 10 11 12 13 - 8".

How will this walk continue? It cannot continue from atom 8, since the three bonds there are now used up. We therefore backtrack to find an atom with an unused bond, and this takes us right back to the initial atom, number 5, which becomes the new terminal atom. We shall use a comma to indicate this

```
5  7  8  9 10 11 12 13 - 8, 5  4  3  2  1  6 - 5
5  7  8  9 10 11 12 13 - 8, 5  6  1  2  3  4 - 5
5  7  8 13 12 11 10  9 - 8, 5  4  3  2  1  6 - 5
5  7  8 13 12 11 10  9 - 8, 5  6  1  2  3  4 - 5
8  7  5  4  3  2  1  6 - 5, 8  9 10 11 12 13 - 8
8  7  5  4  3  2  1  6 - 5, 8 13 12 11 10  9 - 8
8  7  5  6  1  2  3  4 - 5, 8  9 10 11 12 13 - 8
8  7  5  6  1  2  3  4 - 5, 8 13 12 11 10  9 - 8
```

Figure 12.

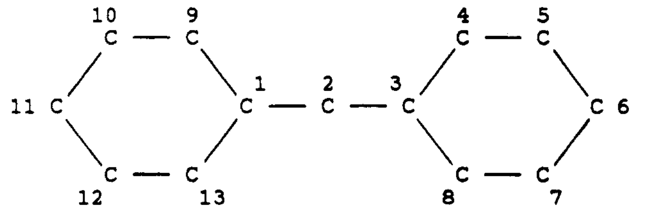


Figure 13.

jump to a new terminal atom, so that the extended walk is written "5 7 8 9 10 11 12 13 - 8, 5". The other three walks are "5 7 8 13 12 11 10 9 - 8, 5", "8 7 5 4 3 2 1 6 - 5, 8", and "8 7 5 6 1 2 3 4 - 5, 8".

From the new terminal atom two continuations are possible in each case, and they are of equal rank. Hence, we now have 8 walks. When we again get ring closure, we find that all bonds have been used up; i.e., the backtrack rule sends us right back to the beginning of the walk, and even there we find no continuation. This, incidentally, is how we know that the walk construction part of the algorithm is finished. We now have the eight walks shown in Figure 12.

All these walks have the same rank sequence, namely, "1, 2, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 1". The significance of there being eight walks is that the algorithm has, as promised, finally recognised the 8-fold symmetry of this ring structure (formed from combinations of the three two-way operations of rotating each ring by itself and of interchanging these two rings). This symmetry is not important in forming the code for the ring structure but will be required later when we reattach the side chains.

The Rest of the Ring Structure Coding Algorithm. To complete the algorithm for coding the ring structure, we carry on from the stage where we have constructed all the walks. Originally, the atoms were labeled arbitrarily, but now we can relabel them in a unique manner with the numbers 1, 2, 3 .... We take any walk, traverse if from beginning to end, and number the atoms in succession. Thus, we relabel the initial atom "1", and thereafter, whenever we come to an atom that has not been relabeled already, we give it the next available number.

Thus, *from a given walk* we get a unique numbering. But the important point is that the conformity rule ensures that this numbering is the same for every walk, in the sense that, if we rewrite the description of each walk, using the new labels, the result will always be the same.

Thus, we now have a unique numbering for the atoms. It will be called the "canonical numbering" of the ring structure. Moreover, the new numbering gives a unique description of the walk that specifies the ring structure completely except for the nature of the atoms. Thus each walk in Figure 12 gives the same walk under its new numbering, namely, 1 2 3 4 5 6 7 8 - 3, 1 9 10 11 12 13 - 1, as the reader can verify. If we insert these new numbers into the ring structure, we obtain Figure 13.

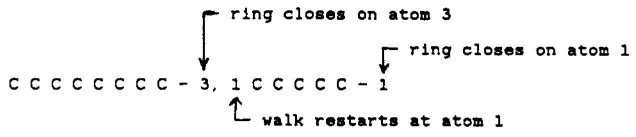


Figure 14.

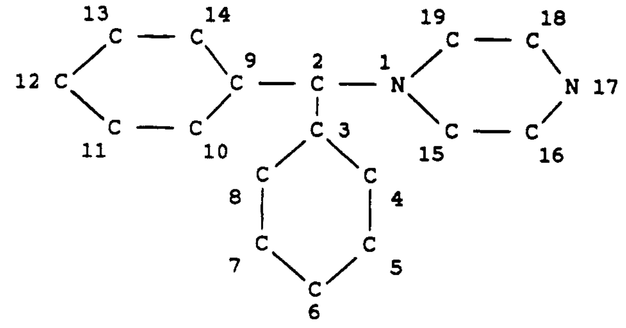


Figure 15.

```
17 16 15  1  2  3  4  5  6  7  8 - 3
17 18 19  1  2  3  4  5  6  7  8 - 3
17 16 15  1  2  9 10 11 12 13 14 - 9
17 18 19  1  2  3  8  7  6  5  4 - 3
17 16 15  1  2  3  8  7  6  5  4 - 3
17 18 19  1  2  9 10 11 12 13 14 - 9
17 16 15  1  2  9 14 13 12 11 10 - 9
17 18 19  1  2  9 14 13 12 11 10 - 9
```

Figure 16.

To obtain a complete description of the ring structure—its code—it suffices to indicate the nature of the atoms. This is done by replacing the *first* occurrence of each label in the renumbered walk by the chemical symbol of the atom that it represents. For subsequent occurrences of a label (which must denote ring closure or the new starting atom after a jump), the new label is left as it is. Thus for the walk above, we obtain Figure 14. We see that the hyphen and comma continue to indicate ring closure and the resumption of the walk after backtracking. Finally, we abbreviate the code by writing, for example, "C8" in place of the string of eight C's. The final code for this ring structure is therefore C8-3,1C5-1.

A Further Example. Let us consider another example—the ring structure of Figure 15 with the atoms labeled arbitrarily. First, we calculate the ranks, by constructing the table for these atoms, as described under Allocating the Ranks. We obtain the following table:

	C	N
2	14	1
3	3	1

The classes rank in the order (2,N), (3,N), (3,C), and (2,C). Thus, atom 17 has rank 1, atom 1 has rank 5, and atoms 2, 3, and 9 have rank 3, while the remaining atoms have rank 4.

Since there is only one atom of rank 1, we have a unique initial atom (atom 17) for the walks. The first extension is possible in two ways, to atoms 16 and 18, and later choices give us eight paths in hand by the time we get to the first ring closure. They are the ones shown in Figure 16, each of which has the rank sequence "1, 4, 4, 2, 3, 3, 2, 2, 2, 2, 2, 3".

Further continuations from the terminal atoms being impossible, we invoke the backtrack rule, which takes us back to atom 2. A further choice occurs at the following step (atom

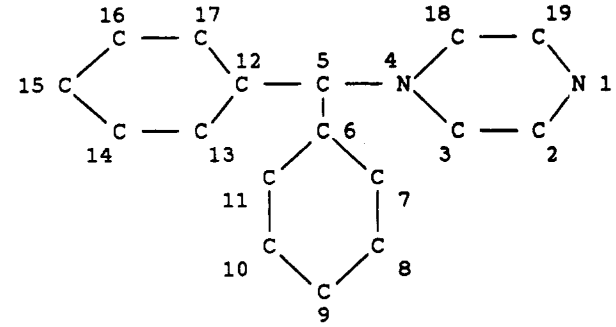


Figure 17.

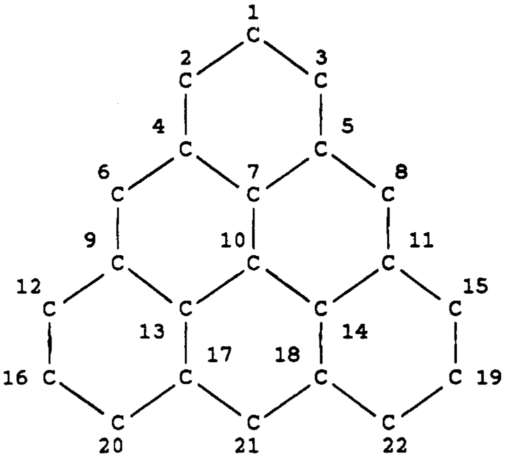


Figure 18.

3 or 9 as the case may be) and doubles the number of walks to 16, one of which is "17 16 15 1 2 3 4 5 6 7 8 - 3, 2 9 10 11 12 13 14 - 9" at the next ring closure. The other 15 walks can be easily written down but need not be given here.

Again, we invoke the backtrack rule, which takes us back to the fourth atom in the walk (atom 1). The continuation from here is unique up to the next, and final, ring closure. We thus end up with 16 paths of which a typical example is "17 16 15 1 2 3 4 5 6 7 8 - 3, 2 9 10 11 12 13 14 - 9, 1 19 18-17".

If we now renumber the atoms in the order in which they are encountered along the walk, we get "1 2 3 4 5 6 7 8 9 10 11 - 6, 5 12 13 14 15 16 17 - 12, 4 18 19 - 1".

We can do this with any of the 16 walks—they will all give the same result. In terms of the ring structure itself, this canonical numbering appears as in Figure 17. If we now insert the atom symbols into the walk description above and make the appropriate abbreviations, we obtain the following code for the ring structure: NC2NC7-6,5C6-12,4C2-1.

Two Special Examples. The two examples considered above illustrate most of what needs to be known about ring structure coding. Two more examples, which will be treated quite briefly, illustrate two small points not yet covered.

Consider the ring structure of Figure 18. There are two kinds of carbon atoms: 10 with ring degree 3 and 12 with ring degree 2. The former, being less numerous, are given rank 1; the others, rank 2.

We therefore start walks simultaneously at each of the 10 atoms with rank 1. A typical walk begins "17 13 10 14 18 21 - 17" up to the first ring closure. Note that there will be quite a few walks that are rejected early on by the conformity rule, including, for example, those that start "17 13 10 7 ...", which are rejected when they fail to close a ring at the same time as the walk given above.

We now have an example of a walk continuing from the same atom (17) at which the ring closed. The walk continues "17 13 10 14 18 21 - 17 20 16 12 9 ...". For this ring structure we get six walks, a typical one being "17 13 10 14 18 21 - 17 20 16 12 9 - 13, 9 6 4 7 - 10, 7 5 8 11 - 14, 11 15 19 22 - 18, 5 3 1 2 - 4". The renumbering procedure converts this to "1 2 3 4 5 6 - 1 7 8 9 10 - 2, 10 11 12 13 -3, 13 14 15 16 - 4, 16 17 18 19 - 5, 14 20 21 22 - 12", which gives the code C6-1C4-2,10C3-3,13C3-4,16C3-5,14C3-12. Note the sequence "... -1C4 ..." in which the absence of a comma and a following integer shows that we have continued the walk without backtracking.

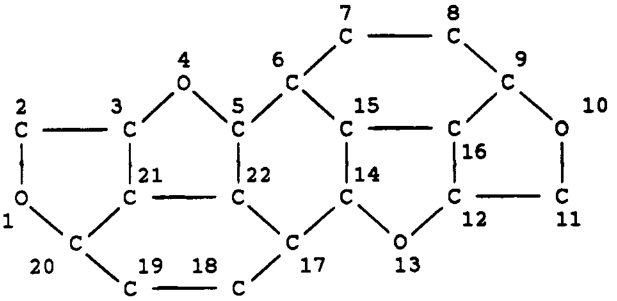


Figure 19.

As a final example consider the ring structure of Figure 19 in which, to save time, the displayed labeling is, in fact, the canonical numbering. We shall not trace the derivation of this numbering or of the code but merely quote the code, which is OC2OC5OC2OC2-6,15C-12,16-9,14C4-1,20C-3,21C-17,22-5 and arises from the walk "1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 - 6, 15, 16 - 12, 16 -9, 14 17 18 19 20 - 1, 20 21 - 3, 21 22 - 17, 22 - 5".

There are several points of interest in this code. We have seen that the algorithm completes a ring whenever it can do so by a single step; but it does not, so to speak, go out of its way to close a ring. Thus, it may happen, as here, that the walk wanders through the ring structure for some while (here 14 steps) before a ring is closed (with the bond 15–6). This walk restarts at atom 15 and, after the very next step (15 16) is able to close a ring again. In fact, it can close two rings, so the closure is made to the most recent atom in the walk, atom 12, in accordance with the closure rule. The walk now resumes, this time at atom 16, and here we have something new. It is now possible to close a ring without extending the walk through any intervening atoms; i.e., the next part of the walk consists of the bond 16–9 alone.

Although this possibility is worth noting, no special provision has to be made for it—the coding rules already given are able to cope with the situation. All that happens is that the symbols indicating the atoms that come between the start and end of that portion of the walk will be absent. This can be seen in the code for this ring structure, where the sequence "... - 12, 16 - 9, ..." shows that, having closed a ring at atom 12, and backtracked to atom 16, the walk immediately closes another ring by means of the bond 16–9. The end of the code, "... – 17, 22 - 5", provides another example of this.

This completes the informal description of the algorithm for coding the ring structure. Eventually we shall need to show that the codes that have been obtained can, in fact, be decoded, i.e., that from the code we can reconstruct both the ring structure itself and the canonical numbering associated with it; but first, we shall complete the description of the overall coding process by giving the method of specifying the way in which the side chains are attached to the ring structure.

(6) LOCATING THE SIDE CHAINS

In the previous section we obtained a single canonical numbering for the atoms in the ring structure, and this was possible because, although in general we constructed several walks covering the ring structure, these walks were all equivalent under the symmetries of the ring structure. This is an important point and needs to be clarified.

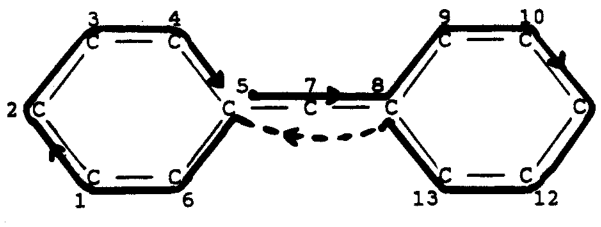


Figure 20.

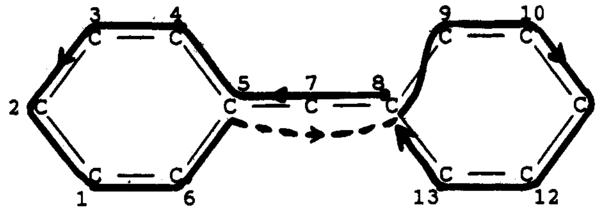


Figure 21.

Consider the ring structure of Figure 11, with the original arbitrary numbering of the atoms, and consider the two walks "5 7 8 9 10 11 12 13 - 8, 5 6 1 2 3 4 - 5" and "8 7 5 4 3 2 1 6 - 5, 8 9 10 11 12 13 - 8". In Figures 20 and 21 these walks have been superposed on the ring structure in a self-explanatory manner. (The dotted line shows the jump after a ring closure.) If the ring structure is regarded as a rigid structure, immovably fixed in the plane of the paper, then these two walks are quite distinct. But the ring structure is not rigid. For example, if we rotate the left-hand ring, by a swivelling action about atom 5, and then flip the whole structure over—interchanging left and right—the ring structure is not changed at all. Moreover, this particular symmetry will convert each of the above walks into the other, thus showing that Figures 20 and 21 are merely two different drawings of the same walk in the same structure. This is what is meant by saying that the walks are equivalent, and this is why the various walks all give the same canonical numbering.

When we return to the whole molecule, however, we have to contend with the fact that the molecule will not, in general, have as much symmetry as the ring structure, so that, although our walks are equivalent in the ring structure, they may not be equivalent in the molecule as a whole. To take an extreme case, if every atom in the ring structure of Figure 11 had an attached side chain and if these side chains were all different, then every atom of the ring structure would be distinguishable from every other atom. The molecule would then have no symmetry, and all the eight walks that we found for this ring structure would be different in the molecule. Under these circumstances the several walks would give us conflicting advice on how to describe where the side chains are attached; according to the numbering for one walk, a particular side chain might be attached at atom 8 of the ring structure, whereas if we were to use another walk, the same "atom of attachment" might be labeled as atom 12.

This problem is easily resolved. We first arrange the side chains in order in much the same way that we ordered clusters when coding acyclic compounds (see part 1). The method is actually simpler in this instance, and consists of two rules: (Side chain rule 1) Shorter side chains precede longer side chains. (Side chain rule 2) If two side chains are the same length, compare them, symbol by symbol, from left to right (i.e., starting at the root) until they differ for the first time. The side chain having the earlier symbol precedes the other.

Here "earlier symbol" relates to the ordering of the symbols that make up the side chain codes, as given in part 1. In other words, side chain rule 2 specifies dictionary order for strings of the same length.

We now give a method for picking out one particular walk and its associated numbering as being the one that we shall use to identify the atoms at which the side chains are attached. First, we list all the *different* side chains that occur in the molecule. We arrange these side chains in order, using the rules just given, and number them consecutively: 1, 2, 3, ... We shall call these numbers the "serial numbers" of the side chains.

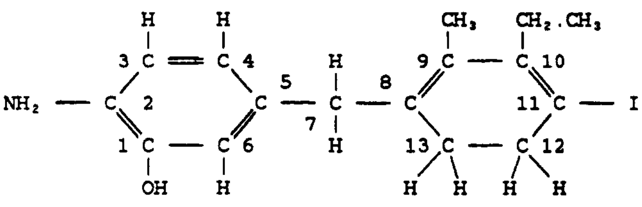


Figure 22.

Now for each walk we have a corresponding numbering of the atoms of the ring structure, and we can therefore write down a sequence of the form $x_1, x_2, x_3, ..., x_n$, where n is the number of atoms in the ring structure and x_i is the serial number of the side chain that is attached at atom number i in the ring structure.

It is now a simple matter to pick out the required walk. It will be the one whose sequence is first in lexicographical or dictionary order. This is the ordering in which the sequence $x_1, x_2, x_3, ..., x_n$ precedes the sequence $y_1, y_2, y_3, ..., y_n$ if, and only if, for some k, $x_i = y_i$ for $i < k$ and $x_k < y_k$. (In other words, basically the same ordering as that given by side-chain rule 2 above.)

An Example. Consider the molecule shown in Figure 22. This has the ring structure that was studied in section 5, and the atoms of the ring structure have been given the same (noncanonical) labeling that we had before (see Figure 11).

First we list all the different side chains, even the trivial ones that in many other systems of nomenclature would be omitted and merely implied. Recall, too, that our definition of a side chain is slightly unusual in that we include the root. Thus, a side chain consisting of a single hydrogen atom attached to a carbon atom is coded as "CH", and this is included in the list of side chains. Even atoms having no bonds to atoms outside the ring structure will be included here, even though they will not be listed as side chains in the final code for the molecule.

In the present molecule, the side chains, correctly coded but not in order, are C, CH, C(NH$_2$), C(OH), C(CH$_2$.CH$_3$), C(CH$_3$), CH$_2$, and CI. Arranging these side chains in order and numbering them serially, we get (1) C, (2) CH, (3) CI, (4) CH$_2$, (5) C(OH), (6) C(CH$_3$), (7) C(NH$_2$), and (8) C(CH$_2$.CH$_3$).

Refer now to section 5 and to the first walk that we obtained for this ring structure (Figure 12), namely, "5 7 8 9 10 11 12 13 - 8, 5 4 3 2 1 6 - 5", which, with the new labeling, gives "1 2 3 4 5 6 7 8 - 3, 1 9 10 11 12 13 - 1". If we now look up in the diagram what side chains are attached at the atoms in this numbering, we obtain the sequence "1, 4, 1, 6, 8, 3, 4, 4, 2, 2, 7, 5, 2" in which, for example, the fact that $x_5 = 8$ means that atom 5 (in the new numbering) has attached to it the side chain with serial number 8.

We can take a short cut here. There is no point in bringing in the new numbering at this stage, since we have to refer back to the old numbering anyway to see which side chain is attached. A shorter method is to trace the walk in the old numbering and drop all but the first occurrence of each number. For the above walk, we get "5 7 8 9 10 11 12 13 4 3 2 1 6".

We now merely write down the serial numbers of the side chains attached at these atoms (this information is directly available from the first part of the coding process when we coded the side chains and removed them to get the ring structure). The result is the same sequence as before.

```
1, 4, 1, 4, 4, 3, 8, 6, 2, 2, 7, 5, 2

1, 4, 1, 4, 4, 3, 8, 6, 2, 5, 7, 2, 2

1, 4, 1, 4, 4, 7, 5, 2, 6, 8, 3, 4, 4

1, 4, 1, 2, 2, 7, 5, 2, 4, 4, 3, 8, 6   *

1, 4, 1, 2, 5, 7, 2, 2, 6, 8, 3, 4, 4

1, 4, 1, 2, 5, 7, 2, 2, 4, 4, 3, 8, 6
```

Figure 23.

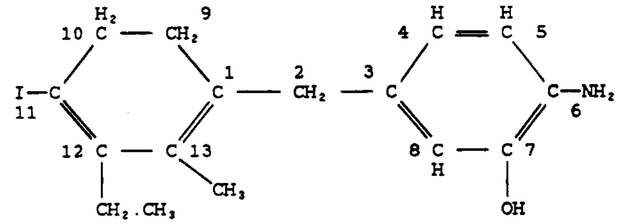


Figure 24.

The second walk in Figure 12, after we have dropped later occurrences of labels, becomes "5 7 8 9 10 11 12 13 6 1 2 3 4" and gives the sequence "1, 4, 1, 6, 8, 3, 4, 4, 2, 5, 7, 2, 2". This sequence is later in lexicographical order than the other, since it has 5 instead of 2 in the first place where they differ (x_{10}).

It is readily verified that the remaining walks give the sequences shown in Figure 23 and that the sixth walk (starred) gives the best sequence. We shall call this walk the "preferred walk". In general, there will be several sequences that tie for being best. If so, we choose any one of the corresponding walks as the preferred walk. It can be shown that it makes no difference which one we choose, since they must all be equivalent under the symmetries of the molecule.

By comparing this walk in the original labeling, namely, "8 7 5 4 3 2 1 6 - 5, 8 13 12 11 10 9 - 8", and the labeled version, namely, "1 2 3 4 5 6 7 8 - 3, 1 9 10 11 12 13 - 1", we obtain the correspondence between the original arbitrary labeling and the canonical numbering. This enables us to attach the various side chains at the correct places in the canonical version of the ring structure as given in Figure 13. In this way we obtain the drawing of the molecule as depicted in Figure 24.

Finally, we build up the full code for this compound. We take the side chains in their serial order (omitting those consisting of a single atom symbol), and for each one, we construct a "prefix" to the ring structure code. As explained in section 2, this prefix consists of the code for the side chain, preceded by the labels (in ascending order) of the atoms at which this side chain occurs. For the above compound, whose ring structure code is C8-3,1C5-1, the prefixes are 4,5,8-CH, 11-CI, $2,9,10\text{-}CH_2$, 7-C(OH), $13\text{-}C(CH_3)$, $6\text{-}C(NH_2)$, and $12\text{-}C\text{-}(CH_2.CH_3)$. Thus, the full code for the compound is $4,5,8\text{-}CH;11\text{-}CI;2,9,10\text{-}CH_2;7\text{-}C(OH);13\text{-}C(CH_3);6\text{-}C(NH_2);12\text{-}C\text{-}(CH_2.CH_3);C8\text{-}3,1C5\text{-}1$.

Note that in the designation of a compound each ring atom is usually referred to twice (explicitly or implicitly): once as the root of a side chain and once as a constituent of the ring structure. Thus, the code has some built-in redundancy. This is a useful feature; it does no harm apart from making the codes slightly longer, and it serves as a check against some errors of decoding.

(7) DECODING PROCESS

As with acyclic compounds the decoding process—that of obtaining the structural formula from the code—is much simpler than the coding process; in fact, the decoding process for even quite complicated molecules can be carried out manually with little trouble. The decoding algorithm is described in full below, although (as will be seen) only the decoding of the ring structure is really new.

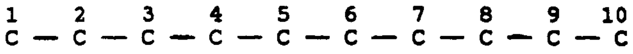


Figure 25.

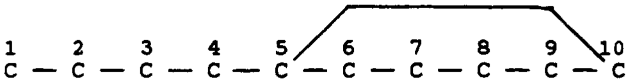


Figure 26.

Note that codes for cyclic compounds are recognizable from the fact that they end with a hyphen followed by an integer, whereas a hyphen never appears in the code of an acyclic compound. Other distinguishing marks are the presence of semicolons and commas.

The code of a cyclic compound is divided up by its semicolons into a number of substrings. Each of these substrings, except for the last, is a prefix defining a type of side chain and the places where it occurs. The initial portion of each prefix, up to the hyphen (which must be present), is a list of integers, separated by commas, each of which indicates an atom of the ring structure at which a side chain of this type occurs. The particular type of side chain is given by the code that follows the hyphen, and this is decoded by the procedure for decoding acyclic compounds, as given in part 1. Note that the first symbol of this code must be an atom symbol, denoting the root atom, which is part of the ring structure. Thus, from these prefixes we can construct a list of side chains, together with their locations. Once the ring structure is decoded, we can then insert these side chains in their proper places.

Thus all that remains is to determine the ring structure and its numbering. We therefore look at its code, which is the final substring defined by the semicolons.

A ring structure code is divided up at the commas into a number of portions, which can be called "code segments". Each code segment represents a portion of the walk, usually made up of a "path" (in the strict graph-theoretical sense, i.e., not going through any atom more than once) followed by a ring closure. In this case, a segment contains exactly one hyphen. Occasionally, as we have seen, we may continue the walk from the same atom at which closure took place, in which case a segment can be divided into subsegments, each containing one hyphen and each representing a path followed by ring closure.

Consider the following code: C10-5,8C4-7,6C2-3,4C2-1C4-2. The first code segment is C10-5 and indicates a path of 10 carbon atoms followed by closure at atom number 5. We therefore start to construct the ring structure by drawing this path, numbering the atoms as we go. This gives us Figure 25.

Following the hyphen is the number of that atom at which the ring closes. We therefore draw a bond from the last atom drawn to the atom bearing the number. It does not matter for now *how* we draw the diagram provided the connections are right. Since the ring closes at atom 5, we obtain Figure 26.

After the closing of a ring, the next symbol will usually be a comma followed by an atom number, which denotes the atom at which the next path starts. We can therefore start to draw the next path. In our example we have 8C4-7, meaning that we start a path at atom 8, extend it via four carbon atoms, and close a ring by going to atom 7. Remembering to number the atoms in the order in which they are drawn, we obtain Figure 27.

The next code segment is 6C2-3, describing a path going from atom 6 via two new carbon atoms to close a ring on atom 3. This gives Figure 28.

The next segment is 4C2-1C4-2, which has two hyphens. We first deal with the subsegment 4C2-1 and get Figure 29.

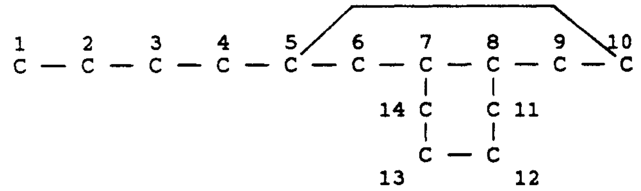


Figure 27.

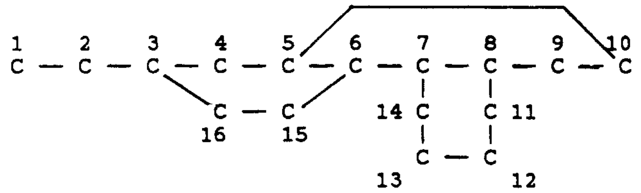


Figure 28.

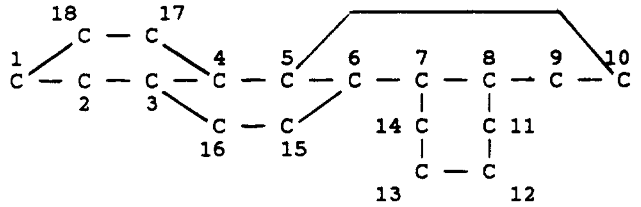


Figure 29.

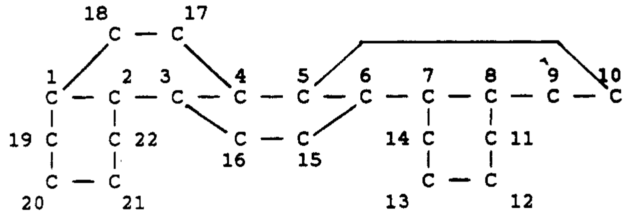


Figure 30.

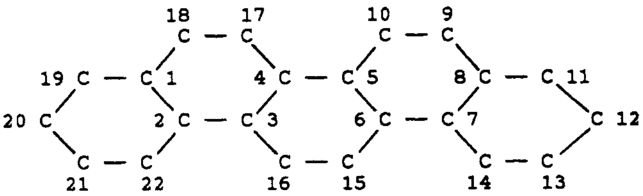


Figure 31.

We now encounter the other way in which the beginning of the next path is denoted. Since there is no comma following the number of the atom at which the last ring closed, the next path starts at that same atom. Hence, the final subsegment, 1C4-2, starts at atom 1 and continues via four new carbon atoms to close a ring at atom 2. When this is drawn, we obtain Figure 30.

We now have a crude picture of the ring structure, but the connections are all specified, so that the ring structure is properly defined. It is easily redrawn in a more conventional way as in Figure 31.

This is the ring structure of picene. The numbering given here for this structure is quite different from that given in the *Parent Compound Handbook*, as one might expect, but it is just as useful for the purposes of information retrieval and has the advantage that it is determined directly from the code of the ring structure, without consulting the *Parent Compound Handbook* or any other document.

One further example will illustrate the complete decoding process, whereby the whole molecule is reconstructed, not just the ring structure. The code for the molecule is 7,8,9,10,11,12-CH;13-CI;3-CH_2;4-N(CH_3);SC2NCNC4-5,2C3-1. From this, by breaking at the semicolons, we deduce that at atoms 7, 8, 9, 10, 11, and 12 of the ring structure we have the side chain CH; at atom 13, there is the side chain CI; at atom 3, there is the side chain CH_2; at atom 4, there is the side chain N(CH_3). The code for the ring structure is SC2NCNC4-5,2C3-1, which is made up of the two segments SC2NCNC4-5 and 2C3-1. These give the walk "S C C N C N C C C C", starting at atom 1 and closing at atom 5, and the walk "C C C", starting at atom 2 and closing at atom 1. Drawing these walks in order we obtain Figure 32.

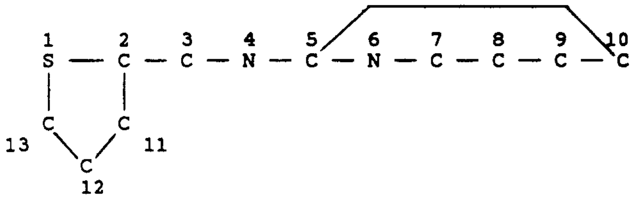


Figure 32.

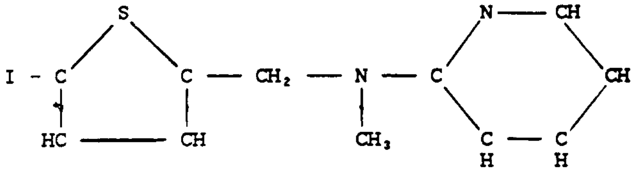


Figure 33.

If we now put in the side chains in the indicated position, we get the full structural formula. It will be somewhat misshapen if we leave the ring structure as drawn in Figure 32, but with the obvious tidying up, we get Figure 33.

According to our convention, all bonds in the ring structure are regarded as if they were single, and to this extent, the above formula is not quite complete. So far we have not assumed any knowledge of what the valences of the various atoms ought to be. Given this information, it is possible to locate the missing multiple bonds, except in so far as resonance and other phenomena give rise to ambiguity of the the kind that we have already noted in section 3. The important thing, however, is that the compound has been adequately identified for all practical purposes. Moreover, since each atom in the ring structure has been numbered, it would be easy to list further information about any particular bonds to whatever extent was considered desirable, identifying the bonds by the numbers of the atoms that they join.

Finally, we note how the redundancy referred to at the end of the last section can sometimes detect errors of decoding. When the ring structure is decoded, a unique number is given to each ring atom. This number will usually occur also as the root of a side chain, and the two atom symbols thus referred to should agree. For example, the nitrogen atom in the ring structure of Figure 33 received the number 4. If the side chain listed for atom 4 had been C(CH_3), then it would be clear that an error had occurred somewhere. Naturally, not all possible errors can be detected in this way. But since even a correct computer program for decoding would still be sensitive to faulty input, a degree of error-correcting capability is worth retaining.

(8) SUMMARY OF THE CODING PROCESS

This section presents a summary of the coding algorithm in a step by step form. A few changes have been made to the timing of the various steps compared with the description given earlier. Thus, the allocation of serial numbers to the side chains has been moved forward, and the determination of the canonical numbering and the code of the ring structure has been postponed until after the preferred walk has been found.

These changes represent slight programming advantages and make the algorithm easier to set out.

Note also that although it was necessary, in section 5, to introduce the rank sequence, in order to show that we obtain the same ring code from any walk, in practice only the final element of that sequence is important at each stage. Thus, the rank sequence, as such, does not feature in the algorithm.

The Coding Algorithm. **(Step 1)** Apply the algorithm for coding an acyclic compound, as given in part 1. If this yields a closed cluster or a closed chain, then the compound is acyclic and its code has been found.

(Step 2) Allocation of Serial Numbers to Side Chains. The atoms in the molecules have been partitioned into clusters (side chains). Order these clusters (a) by increasing length and (b) lexicographically for clusters of the same length. Allocate serial numbers to the clusters in this order, assigning equal serial numbers to identical clusters. This procedure is described in section 6.

(Step 3) Identifying the Ring Structure. Form the ring structure of the compound by taking just the root atoms of the side chains and the bonds between them. These bonds will now all be treated as single bonds no matter what they were originally.

(Step 4) Ranking the Ring Atoms. Classify the atoms of the ring structure by ring degree and atom name and allocate ranks to the various classes obtained according to the three criteria (a) smallness of size, (b) ring degree, and (c) atom name, in that order of priority. This rank allocation procedure is described in section 4. Every ring atom now has a rank.

(Step 5) Starting the Walks. The atoms of lowest rank are taken as the starting points of a set of walks. (Steps 6–11 that follow form a procedure that is performed several times until the final set of walks is obtained. At each stage there is a set of walks in hand each going from an initial atom to a terminal atom. Each iteration of the procedure extends these walks by adding a bond at the terminal atom, maybe in more than one way. At the same time, some of the walks may be eliminated. The bonds already in a walk will be said to be "used" bonds; the remainder are "unused".)

(Step 6) Ring Closure. Examine each of the walks in hand and determine if ring closure is possible, that is, if the terminal atom T is adjacent by an unused bond to some atom A already in the walk. If T is adjacent in this way to more than one atom in the walk, choose A to be the one first encountered on moving back along the walk from T. If ring closure is not possible for any of the walks in hand, go to step 10.

(Step 7) Elimination of Walks during Ring Closure. Eliminate any of the walks in hand that do not allow ring closure. Also, eliminate any of the remaining walks in hand for which atom A is further back along the walk than it is in some other walk in hand. The surviving walks now all have ring closure at atoms at the same distance back along the walk.

(Step 8) Walk Extension on Ring Closure. Extend each of the walks in hand by adding the bond TA, which now becomes a used bond. The fact that this is a closure bond is noted. If there is at least one unused bond at atom A (this will either be true for all walks or false for all of them), then A becomes the new terminal atom. Go to step 6.

(Step 9) Backtracking. All bonds at atom A are used. In each walk, retrace the walk back from atom A (via T) until an atom B is first reached at which there is an unused bond. If there is such an atom, then a jump is recorded from A to B; B becomes the new terminal atom. When this has been done for all the walks, the algorithm continues from step 6. If there is no such atom B, that is, if even the initial atom of the walk has no unused bonds, then the construction of walks is completed, and the algorithm continues from step 12.

(Step 10) Normal Extension. For each of the walks in hand find in the ring structure the atom of lowest rank that is adjacent by an unused bond to the terminal atom. If there are *n* such atoms, replace the walk by *n* copies of itself and extend each copy by the bond to one of these atoms (a different one for each copy). This atom becomes the new terminal atom for the walk, and the bond by which it was reached is now used.

(Step 11) Conformity Rule. Examine the walks obtained in step 10 and eliminate any whose terminal atom has higher rank than the terminal atom of some other walk. The remaining walks now end in atoms of the same rank. Go to step 6.

(Step 12) Finding the Preferred Walk. Take each of the walks in turn and consider the ring atoms in the order in which they first appear in the walk. Construct the corresponding sequence of serial numbers of the side chains rooted at these atoms, in the same order. Compare these sequences lexicographically and eliminate any walk for which the sequence comes after the sequence of some other walk. The remaining walks will all have the same sequence, and will be equivalent under the symmetries of the molecule. Choose any one of them to be the preferred walk.

(Step 13) Obtaining a Canonical Numbering. Renumber the ring atoms with the labels 1, 2, 3, ... in the order that they first appear in the preferred walk.

(Step 14) Obtaining the Ring Structure Code. In what follows, "number" refers to the canonical numbering obtained in step 13.

(a) The required code is a string of atoms and other symbols. Start with the initial atom of the preferred walk and record its atom symbol.

(b) Continue along the walk, recording the symbol of each atom as it is encountered, until ring closure takes place.

(c) When ring closure takes place, add to the string of symbols so far recorded a hyphen and the number of the atom on which closure takes place.

(d) If the end of the walk has been reached, go to (e). If no jump took place after this ring closure, continue from (b). If a jump took place, add to the string of symbols a comma and the number of the atom to which the jump was made. Go to (b).

(e) Replace any atom symbol, X, that occurs *n* times consecutively in the recorded string of symbols ($n > 1$) by X followed by the integer *n*.

The symbol string that results is the code for the ring structure.

(Step 15) Constructing the Code for the Molecule. Take the side chains consisting of more than one atom, in the order of their serial numbers, and for each one list the (canonical) numbers of the ring atoms that are roots of that kind of side chain. Construct a code prefix by listing these numbers, in ascending order and separated by commas, followed by a hyphen and the code of the side chain. The code for the whole molecule then consists of these code prefixes, each followed by a semicolon, listed in the serial order, followed by the code of the ring structure that was found in step 14.

APPENDIX

As in part 1, I give below some practice examples for the benefit of the reader. Since the coding of cyclic compounds is more difficult than that of acyclic compounds, it would not be fair to include any large or complex molecules among these examples. The five compounds in Figure 34 are sufficiently simple that their coding by hand should be quite easy.

Since decoding, even for cyclic compounds, is relatively easy to accomplish by hand, it is possible to give quite complicated molecules for decoding practice. The following codes are for

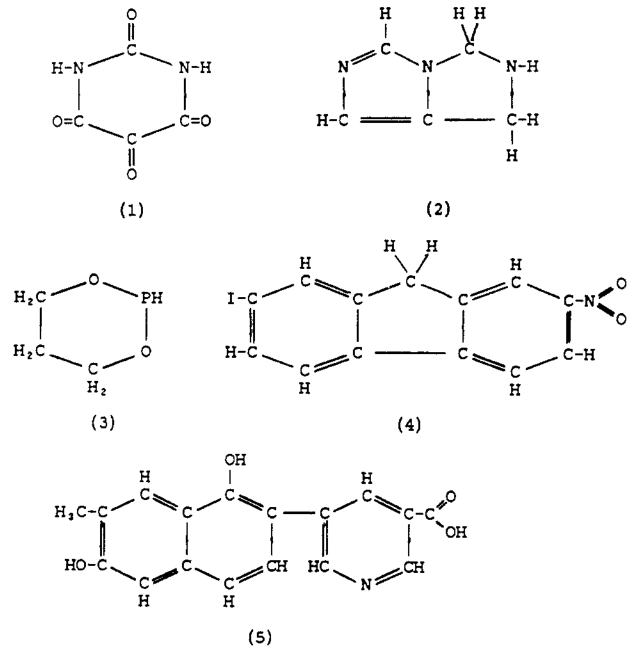

(1) (2) (3) (4) (5)

Figure 34.

some compounds that are identified in the solutions by their trivial names or other designations:

(6) 8,9,10,11,12,13,14,15-CH;2,7-CH_2;1-N(CH_2.CH:CH_2);NC6-1,6C4-5,4C4-3.

(7) 3,11,12,13-CH;7,15-C=O;5,6,14,17-C(OH);16-C(C=O.NH_2);18-C(N(CH_3)$_2$);10-C(OH)(CH_3);C6-1C4-2,9C4-8,5C4-4.

(8) 3,4,6,7,9,10-CH_2;1-P=S;PNC2-2,1NC2-5,1NC2-8.

(9) 3,7-CH;4,6-NH;2-CH_2;5-C=O;8-CH((CH_2)$_4$.C=O.OH);SC2NCNC-3,7C-1.

(10) 7,8,10,11,12,14,15,16,17,19,20,21,23,24-CH; 3,5,27,29-CH_2;4,28-CH(OH);2,9,13,18,22,26-C(CH_3);6,30-C(CH_3)$_2$;C6-1C24-25.

The following is an artificial example giving just the ring structure code of a hypothetical (but possibly feasible) compound for which the ring structure is the one-dimensional skeleton of one of the regular Archimedean solids. The reader is invited to determine which regular solid it is.

(11) C3-1C3-4,6C3-7,9C2-3,11C-10,12C3-13,15C2-8,17C-16,18C2-5,20C-19,21C2-14,23C-22,24-2.

The following is also a hypothetical example. It gives the code of a compound for which the ring structure is not planar, and it is included to illustrate the fact that the coding system does not assume that the structural formula of the molecule can be drawn in the plane with no crossing of bonds.

(12) 1,3,5,7,9,11-CH;2,4,6,8,10,12,13,14,15,16,17,18-CH_2;C12-1C2-7,11C2-5,9C2-3.

Solutions.

(1) 1,3-NH;2,3,5,6-C=O;NCNC3-1.

(2) 3,5-CH;7-NH;6,8-CH_2;CNCNC-1CNC-2.

(3) 1-PH;3,4,5-CH_2;POC3O-1.

(4) 6,8,9,10,12,13-CH;7-CI;5-CH_2;11-C(N=O_2);C5-1C4-2,4C4-3.

(5) 2,8,9,10,13,14,16-CH;5,11-C(OH);12-C(CH_3);15-C (C=O.OH);NC8-4,7C4-6,3C3-1.

(6) Azapetine (see page 120D of reference 10).

(7) Tetracycline (see page 69H of reference 3).

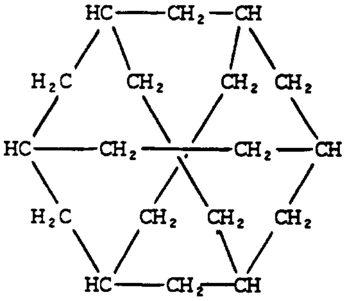
(8) Triethylenethiophosphoramide.

(9) Biotin (see page 82E of reference 3).

(10) Zeaxanthine (see page 21B of reference 3).

(11) The regular solid is the truncated cube.

(12) The compound is



REFERENCES AND NOTES

(1) Buckingham, J., Executive Ed. "Dictionary of Organic Compounds"; Chapman and Hall: New York, London, and Toronto, 1982.

(2) Chemical Abstracts Service, "Parent Compound Handbook"; American Chemical Society: Washington, DC, 1976.

(3) Chemical Abstracts Service, "SOCMA Handbook. Commercial Organic Chemical Names"; American Chemical Society: Washington, DC, 1976.

(4) Corneil, D. G.; Read, R. C. "The Graph Isomorphism Disease". *J. Graph Theory* **1977**, *1*, 339-363.

(5) Figueras, J. "Substructure Search by Set Reduction". *J. Chem. Doc.* **1972**, *12*, 273-244.

(6) Gordon, M.; Kendall, C. E.; Davison, W. H. T. "Chemical Ciphering. A Universal Code as an Aid to Chemical Systematics"; The Royal Institute of Chemistry of Great Britain and Northern Ireland: London, 1948.

(7) Morgan, H. L. "The Generation of a Unique Machine Description for Chemical Structures—a Technique Developed at Chemical Abstracts Service". *J. Chem. Doc.* **1965**, *5*, 107-113.

(8) Read, R. C. "A New System for the Designation of Chemical Compounds. 1. Theoretical Preliminaries and the Coding of Acyclic Compounds". *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 135-149.