COMPUTER-ASSISTED STRUCTURE GENERATION FROM A GROSS FORMULA: II. MULTIPLE BOND UNSATURATED AND CYCLIC COMPOUNDS. EMPLOYMENT OF FRAGMENTS*

I.P. BANGOV

Institute of Organic Chemistry, Bulgarian Academy of Sciences, Sofia 1040, Bulgaria

and

K.D. KANEV

Institute of Mathematics with Computing Center, Bulgarian Academy of Sciences, Sofia 1040, Bulgaria

Received 11 June 1986
(in final form 16 March 1987)

Abstract

A general method for the generation of molecular structures from a gross formula is described. This method allows the construction of different classes of structures: saturated, multiple bond, conjugated, as well as cyclic and acyclic, in a uniform way. The employment of fragments within this approach is provided.

1. Introduction

Several approaches to building generators for chemical structures have been discussed in the literature [2–6]. All of them are based on a graph-theoretical (topological) description of the molecular structure, but they vary according to the objective of the program and in the procedures or algorithms developed in the course of the research. However, owing to the combinatorial nature of structure generation, there are problems common to all approaches. Two such problems are the enormous number of structures corresponding to the molecular formulae of all but the smallest molecules, and the duplicated structure problem which attends the generation process.

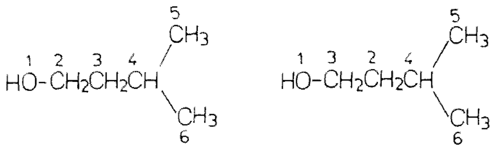
To help solve the first problem, additional spectral and/or chemical information is useful for inferring the presence of particular molecular fragments (called either "superatoms" in ref. [3] or "components" and "macrocomponents" in ref. [4]).

*For part I, see ref. [1].

They are employed, rather than single atoms, as basic elements in the combinatorial process of structure generation. By increasing the size of the fragments, the number of combinatorial operations is drastically reduced. In this paper, we shall not discuss the ways of deriving fragments from spectral information, but their handling in the process of structure generation.

The second problem originates from isomorphisms of different representations of the chemical graph. In general, we divide the duplicated structures into two classes. The structures obtained after permuting two groups or fragments bonded to different valences of the same atom fall in the first class. The duplicated structures of the second class are generated by permuting two equivalent atoms or groups, all other structural fragments being arranged in such a way that structures isomorphic to those previously generated appear.

In most programs, a priori pruning of the duplicated structures is required [7]. In contrast to this, our approach to structure elucidation, based on ranking the chemical structures according to the extent they obey the ^{13}C chemical shift/charge density linear relationship [8], requires all connectivities within a given set of atoms or fragments to be considered. For instance, two structures such as:



which are duplicated in the chemical sense and isomorphic in the graph-theoretical sense, are not considered equivalent in our approach. This is because the chemical shifts of atoms 2 and 3 are different (the influence of the adjacent OH group), and thus the two structures will have different ranking factors. Accordingly, we prune the duplicated structures of the first class in the course of their generation. Those of the second class are generated, and ranked with respect to the ^{13}C chemical shift/charge density linear relationship [8], and then coded through an appropriate unique coding algorithm. Hence, they are perceived by the computer as duplicated structures, and pruned during the stage of their display only in the case that they possess a worse ranking factor than those previously generated. The problem of perceiving and pruning the duplicated structures of the second class will be discussed in detail in subsequent papers dealing with the coding and displaying procedures. Here, we shall outline only the basic ideas of our coding method for completeness of the exposition.

The present work represents our efforts in the development of a general algorithm for the generation of structures of different classes: saturated and unsaturated, cyclic and acyclic.

2. Method

Our approach to structure generation in the case of acyclic saturated compounds has been reported in a previous paper [1]. The general features are outlined here to aid in clarifying the following discussion.

The starting point of the process of structure generation is the gross formula. The notion "gross formula" implies a formula constituted both of single atoms and fragments. Thus, the gross formula Fa1Fb2C4O1H12 indicates that the generated structures consist of a fragment Fa, two fragments Fb (with substructures known to the user and subsequently fed into the computer), four carbon, one oxygen and twelve hydrogen atoms. Since the processing of the fragments will be discussed in sect. 5 of this paper, here the method will be exemplified with a molecular formula, e.g. $C_4N_1O_1H_{11}$ [figs. 1(a) and 1(b)]. As in other approaches, the hydrogen atoms do not participate directly in the structure generation process. The skeleton atoms (C, N, O, S, etc.) of a valency higher than one and the univalent atoms F, Cl, Br, etc., are considered graph vertices with degree equal to their valency.

Let the first atom of the molecular formula be a carbon atom. It has four valences, each one of them being saturated by one valence of any of the other atoms. Thus, a substituted methane is obtained. Hence, the substituting atoms (of valency n) have one saturated valence and $n-1$ unsaturated (free) valences. The latter are saturated on their part by one valence of each of the remaining atoms, which are also saturated in the same way. The valences left unsaturated at the end are filled with hydrogen atoms.

This procedure is illustrated by the generation of two isomers in fig. 1(a). One sees that the valences of the free atoms are marked in different ways: one valence of each of the atoms, except the first, is designated as "$\star$" (we shall call it a $\star$-type valence), and all of the first atom and $n-1$ of the other atom valences are designated as "+" (we shall call them +-type valences). The structure generation process is carried out by generating different transpositions of the $\star$-type valences with respect to the +-type valences, i.e. by generating ${}^{m}P_{L}$ permutations of m valences selected from all L +-type valences without repetition, and saturating them with the m $\star$-type valences. Note that, since all the valences of a given atom are equivalent, the choice of a valence being $\star$-type is arbitrary.

For computer implementation of this approach, the following mathematical representation has been developed. The molecular formula is transformed into two string arrays [see fig. 1(b)]: a vector array SUBS containing the $\star$-type valences with their numbering given in the NSUB array [the digits below in fig. 1(b)], and a two-row matrix GRAPH with the first row consisting of all L +-type valences, the second row elements being filled with "+". The numbers below (contained in the NGRAPH two-row integer matrix) designate the numbering of the corresponding atoms. The latter is fixed in the course of transforming the molecular formula into the SUBS and

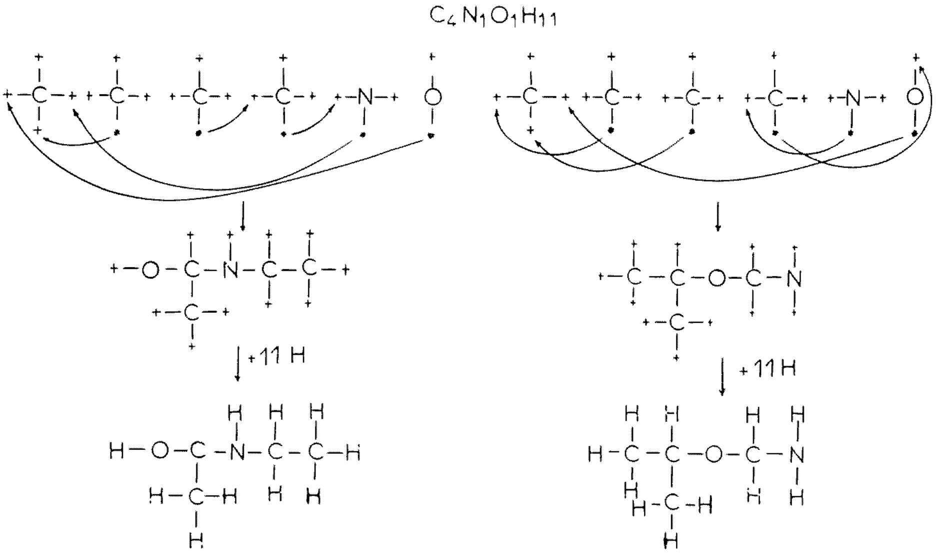


Fig. 1(a). Depiction of the method of structure generation with two acyclic saturated isomers.

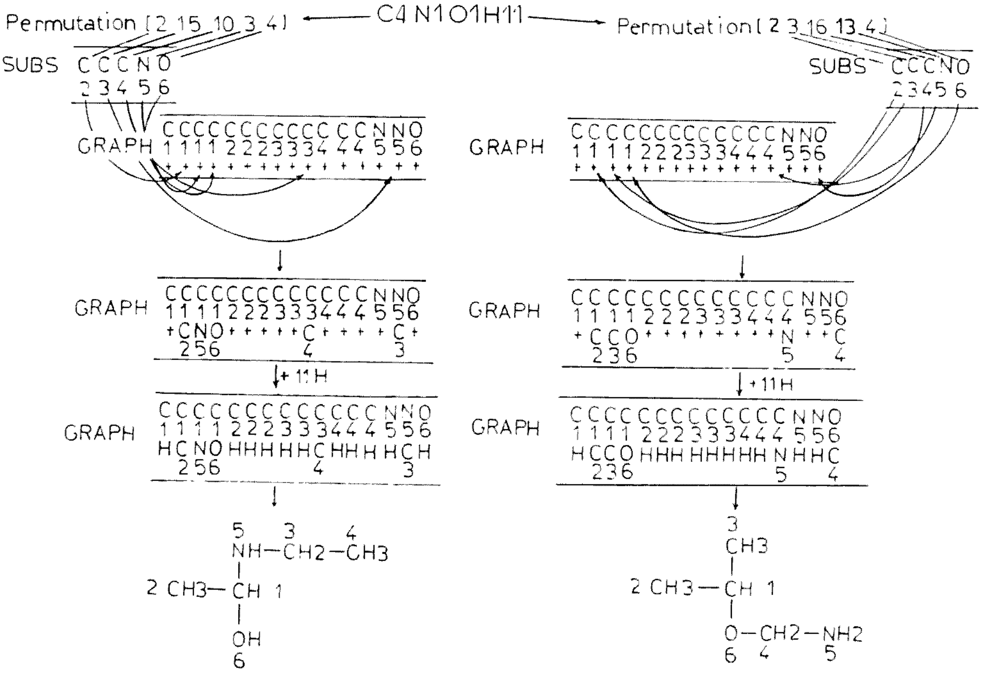


Fig. 1(b). Mathematical representation of the generation process.

GRAPH (NSUB and NGRAPH, respectively) matrices and follows the order of their appearance in the molecular formula. The valences in the SUBS and GRAPH arrays are denoted with the atomic designations commonly accepted (C for carbon atom valence, N for nitrogen atom valence, O for oxygen atom valence, etc.). It is worth mentioning that this algorithm could be developed on the basis of NSUB and NGRAPH matrices only, but we believe that the use of the SUBS and GRAPH representations makes our exposition clearer.

All except the first atoms are represented by $n - 1$ +-type valences in the GRAPH(1) row (n is the atomic valency, as stated above), and by one ⋆-type valence in the SUBS array. The first atom is represented by all of its n +-type valences in the GRAPH(1) row. This representation corresponds to the separation of the free atom valences into + and ⋆-types described earlier [fig. 1(a)].

As already mentioned, structure generation is carried out by generating ${}^{m}P_{L}$ permutations of m selected (from the L) +-type valences of GRAPH(2) row, without repetition, and saturating them with the m ⋆-type valences from the SUBS array [see fig. 1(b)]. The remaining unsaturated valences are filled with hydrogen atoms.

Two controls are built into the program: the first disallows valences corresponding to the same atoms [having the same numbers in NGRAPH(1) and NGRAPH(2)] to be juxtaposed (no atom can be connected with itself); the second one prunes all the structures generated after permuting two valences from the GRAPH(2) row bonded to the same atom. In this way, most duplicated structures of the first class are eliminated. As a consequence, the combinatorial problem is alleviated to some extent, since instead of $(m + L)!$ permutations generated by permuting all the $m + L$ valences (m ⋆-type and L +-type), only

$${}^{m}P_{L} = \frac{L!}{(L-m)!}$$

permutations are generated. Unfortunately, such a reduction is not sufficient for most practical cases.

It is appropriate at this point to discuss some aspects of the mathematical nature of our two-row graph representation. First, it is clear that a chemical bond is represented by juxtaposing two valences of the GRAPH(1) and GRAPH(2) rows. Hence, the number of bonds equals L, the dimension of the GRAPH matrix. Second, according to its definition, the GRAPH array represents a directed graph (digraph) [9] with edge orientation pointing from the GRAPH(1) to the GRAPH(2) row. This explains our separation of the atom valences into ⋆ and + types. As is well known, in digraphs the order of nodes in a pair matters, i.e. one of the nodes is that in which the edge is going into, the other is that from which it is going out. Hence, any bond (edge) is formed by two different types of valences. Third, each permutation of two ⋆-type valences is conducive to cleavage of two bonds of the structure previously generated and forming two new bonds. Hence, a new structure is generated simultaneously with the generation of a permutation. On the one hand, this makes our approach very fast, on the other, however, there is no way to check whether the generated structure will be canonical at the time of its generation, as in other approaches. This check must be done after the structure is generated, as stated earlier.

3. Multiple bond structure generation

The number of multiple bonds and cycles in a molecular formula can be pereceived by calculating the degree of unsaturation (Du) [10] :

$$\begin{aligned} Du &= -\tfrac{1}{2}n_1 + \tfrac{1}{2}n_3 + n_4 \\ &\quad + \tfrac{3}{2}n_5 + 2n_6 + 1, \end{aligned}$$

where n_1, n_3, n_4, n_5, n_6 are the numbers of the univalent, trivalent, tetravalent, pentavalent, and hexavalent atoms, respectively. In this format, Du represents the number of double bonds plus twice the number of triple bonds plus the number of rings. If a double bond (Du = 1) is considered to be a two-member cycle, and a triple bond (Du = 2) two fused two-member cycles, the generation of structures having multiple bonds can be reduced to the generation of cyclic structures discussed in the next section.

However, we favour another approach. Since the number and type of multiple bonds can be derived from a careful study of the spectral information available (^{1}H and ^{13}C nuclei of the saturated and unsaturated parts of the molecule resonate in different regions of the NMR spectrum), we consider the atoms forming single bonds to be different from those forming double and/or triple bonds. In other terms, say, for the carbon atoms, we have five types of atoms: atoms C, forming single bonds only, of valency n = 4, atoms =C, forming a double and two single bonds, of valency n = 3, atoms ⋆C, forming a triple and a single bond of valency n = 2, atoms C:, forming two conjugated bonds and a single one, of valency n = 3, and atoms =C=, forming two double bonds of valency n = 2. In the same way, atoms =O (n = 1), =N (n = 2), ⋆N (n = 1), etc., are introduced. Hence, the gross (or molecular) formula can be formed of each of these atom types, e.g. the molecular formula C2=C3=O1H8 in fig. 2 is formed of two C, three =C, one =O and eight hydrogen atoms. The structure generation process is carried out in the same manner as described in the preceding section, but taking into account the corresponding valences of the atoms introduced here. For instance, the =C atom in fig. 2 has one ⋆-type valence and two +-type valences. A control is built into the program which ensures that each multiple bond atom, e.g. the =C atom, has at least one valence saturated by a multiple bond atom valence, for example, a valence of =C, or =O, etc., and there must be a juxtaposing

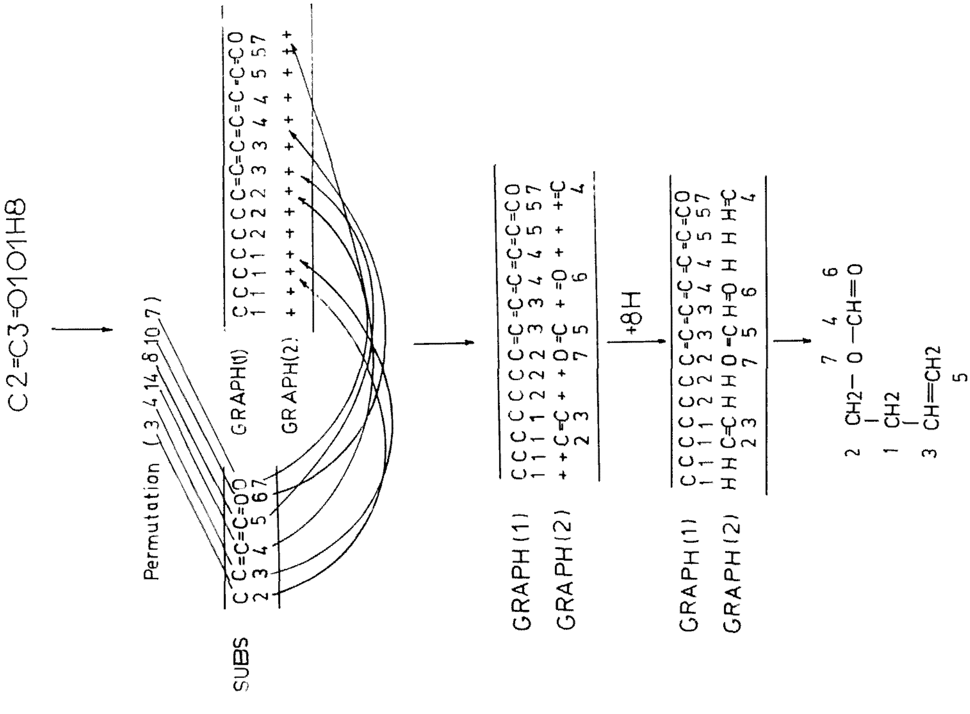


Fig. 2. Generation of multiple bond structures.
(a) Double bond containing structures.

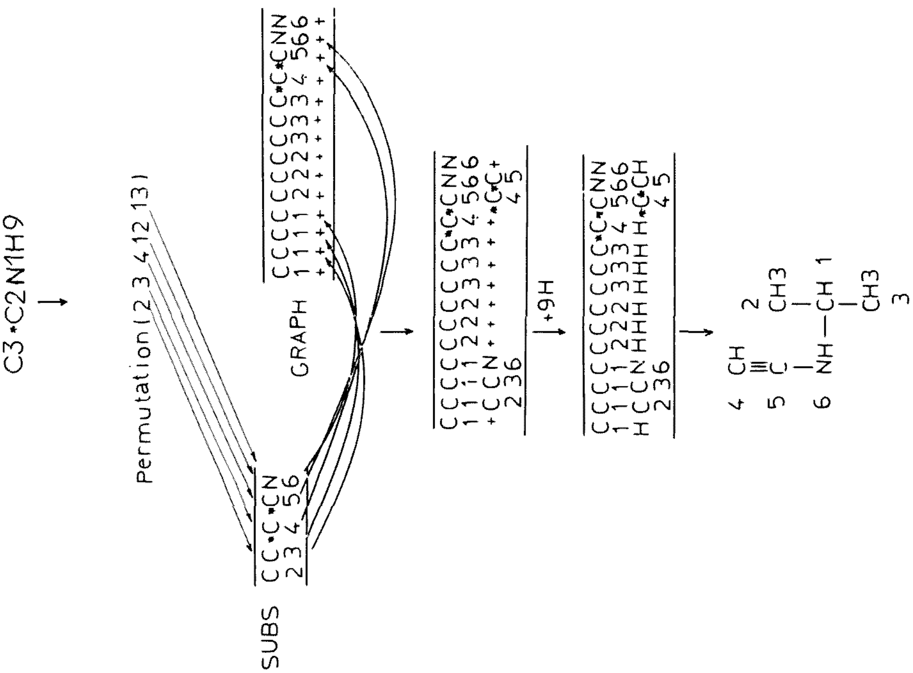


Fig. 2. Generation of multiple bond structures.
(b) Triple bond containing structures.

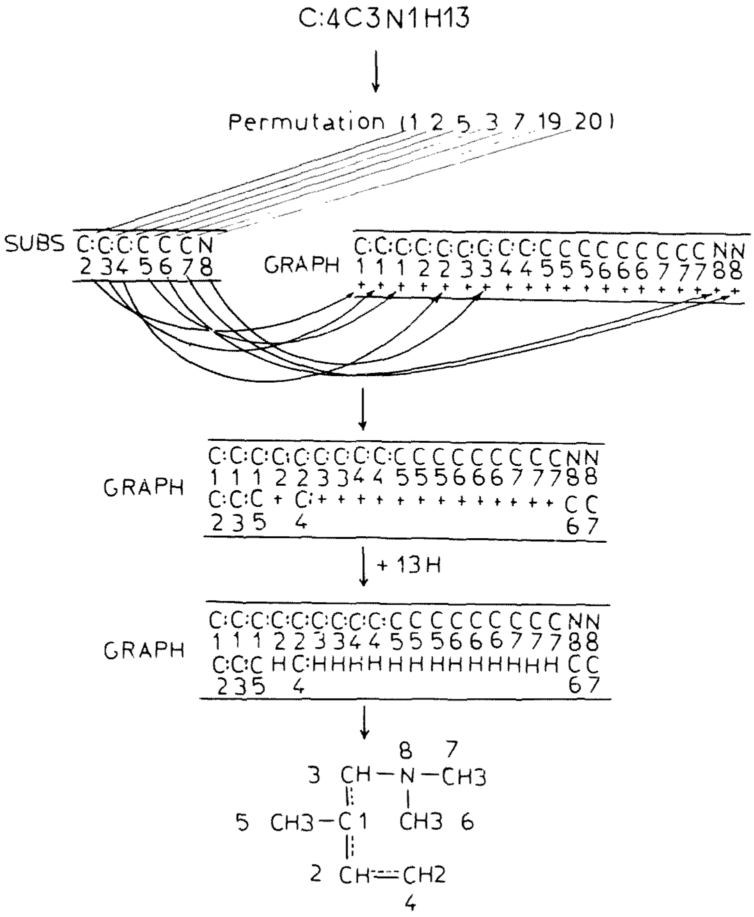

C:4C3N1H13
Permutation (1 2 5 3 7 19 20)
SUBS C:C:C:C C C N
2 3 4 5 6 7 8
GRAPH C:C:C:C:C:C:C:C:C C C C C C C C C N N
1 1 1 2 2 3 3 4 4 5 5 5 6 6 6 7 7 7 8 8
GRAPH C:C:C:C:C:C:C:C:C C C C C C C C C N N
1 1 1 2 2 3 3 4 4 5 5 5 6 6 6 7 7 7 8 8
C:C:C + C:+ + + + + + + + + + + + + + C C
2 3 5 4 6 7
+ 13H
GRAPH C:C:C:C:C:C:C:C:C C C C C C C C C N N
1 1 1 2 2 3 3 4 4 5 5 5 6 6 6 7 7 7 8 8
C:C:C H C:H H H H H H H H H H H H H H C C
2 3 5 4 6 7
8 7
3 CH—N—CH3
5 CH3—C1 CH3 6
2 CH==CH2
4

Fig. 2. Generation of multiple bond structures.
(c) Conjugated bond structures.

of GRAPH(1) to GRAPH(2) rows having double bond atom valences in order to form a double bond. The process of generation of multiple bond structures is illustrated in fig. 2 for (a) double, (b) triple, and (c) conjugated bond structures.

4. Generation of cyclic structures

In this section, a generalization of our approach to generation of cyclic compounds is discussed. In most of the cases, the cyclic structures are considered either derivatives of spanning trees [11,12] : acyclic graphs containing all the vertices (atoms) of the cyclic graph, but not containing all the edges (bonds), or of vertex-graphs, consisting of nodes, representing the points of ring fusion [3,13].

In our program, we use an algorithm for generation of cyclic structures which incorporates some aspects of both approaches. The number of cycles Dc is preliminarily determined according to the equation:

$$Dc = Du - Nd - Nc - 2\,Nt,$$

where Du is the degree of unsaturation, defined in the previous section, and Nd, Nc, and Nt are the number of double, conjugated, and triple bonds, respectively. On the other hand, Dc is the cyclomatic number: $Dc = Lc - N + 1$, where Lc is the number of cyclic structure bonds, i.e. the dimension of the cyclic structure GRAPH array representation, and N is the number of atoms. Since, for an acyclic structure such as the spanning tree, $Dc = L - N - 1 = 0$, where L is the dimension of the spanning tree GRAPH array representation, the cyclic structure GRAPH array dimension equals $Lc = L - Dc$. This decreasing of the dimension of the GRAPH array is carried out at the stage of construction of the GRAPH and SUBS arrays. Dc valences are transferred from GRAPH(1) to the SUBS array, i.e. Dc +-type valences are transformed into ⋆-type valences. In fact, the extra Dc ⋆-type valences in the SUBS array form closure bonds by saturating any one of the +-type valences. We shall designate them as $|@\rangle$. This designation helps us in a subsequent search for closure bonds. Hence, the number of generated permutations becomes ${}^{m+Dc}P_{L-Dc}$. The generation of a cyclic isomer by this algorithm is illustrated in fig. 3.

The problem here is which Dc out of the L +-type valences are to be chosen. They should meet the following requirements: first, to be that of a cyclic atom, and second, to form nodes of a degree higher than 2, i.e. in chemical terms, to have valences higher than 2. Note that in the case of condensed cyclic compounds, these nodes form vertex graphs. Thus, in our method, the spanning trees and the vertex graphs are simultaneously generated, the former by the m ⋆-type valences and the latter by the extra Dc closure valences. On the one hand, we present here a general method allowing generation of all kinds of cyclic structures within a given Dc. On the other hand, all possible connectivities should be traced. Hence, an exception to the second rule is admitted: valences of the first atom, even having valency 2, can also form closure bonds. This then allows the manipulation of one-cycle structures having no nodes of a degree higher than two, e.g. cyclohexane. Accordingly, in the case of a general molecular formula, all the carbon, nitrogen, sulfur, phosphorus, etc. atoms of a valency higher than two are potential candidates. Consider the case of a molecular formula with $Dc = 2$ (fig. 3). The program chooses successively two out of all seven carbons and generates all cyclic structures corresponding to this choice. Then, each carbon atom is combined with the only nitrogen atom, and the generation process is repeated. Obviously, in more complex cases this procedure is conducive to a wasteful number of combinatorial operations. However, as already noted, the use of the molecular formula only is unrealistic for most practical problems. Evidently, additional

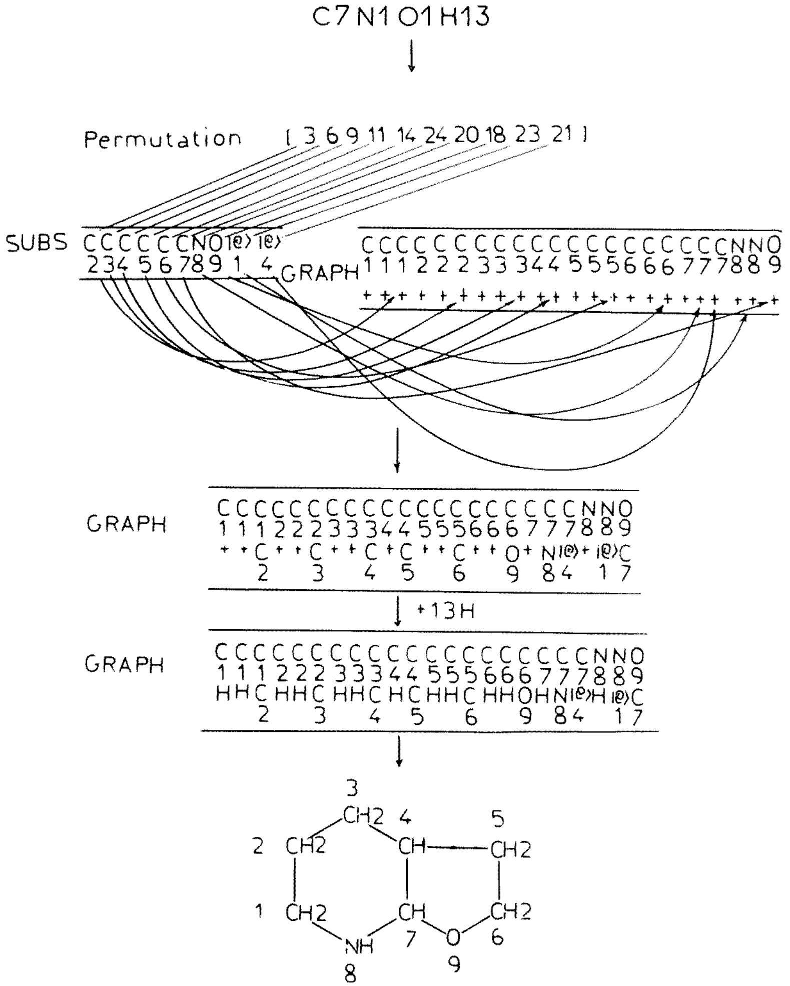
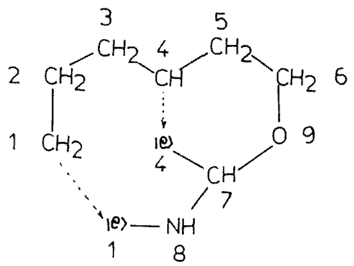


Fig. 3. Generation of cyclic structures.

spectral and/or chemical information is necessary. As a first step, information about the number of adjacent hydrogen atoms can be obtained, say, from off-resonance multiplicity of the ^{13}C NMR spectrum. Thus, in the case of fig. 3, the two CH carbon atoms (4 and 7) and the only nitrogen as well as the CH_2 atom (1), as discussed above, are candidates for the formation of closure bonds.

Figure 3 shows that our approach to generation of cyclic structures is the same as that for acyclic structures. This is based on the fact that we could consider each cyclic structure an acyclic one. For example, the structure from fig. 3 can be represented by the following acyclic structure:



having $Dc = 2$ atoms (1 and 4 in our case) with a valency equal to $n - 1$ and $Dc = 2$ univalent atoms |@⟩ with the same numbering (1 and 4). This allows the processing of the cyclic structures as acyclic ones, i.e. as digraphs. Note that if we had processed them as directed cyclic graphs, non-Euler structures could not be easily treated.

5. Employment of fragments

If we have information regarding the connectivity within substructures of the molecular structure (here called "fragments"), the gross formula can be constituted of both fragments and single atoms. As previously stated, such information could not only be valuable for easing the combinatorial problem, but also increases the reliability of the subsequent inference step. Information of this type can be derived both on chemical grounds and/or by using physical techniques and spectroscopic results. Various ^{13}C NMR techniques are rich sources of such information. As a first step, the ^{13}C-adjacent ^{1}H multiplicity, either from off-resonance ^{13}C NMR spectra or from other NMR techniques such as APT or ^{13}C-^{1}H COSY, can be employed. On the basis of this information, the separate carbon atoms are transformed into a CH group for a doublet, a CH_2 group for a triplet, and a CH_3 group for a quartet. Accordingly, the GRAPH(2) row is modified and the number of +-type valences is substantially decreased, which also decreases the number of permutations to be investigated. Unfortunately, this decrease is not sufficient for most practical cases. Obviously, the number of permutations will sharply decrease with an increase in size of the fragments employed. In the extreme case, one of the fragments could be the molecular skeleton.

Our idea of a fragment is close to the idea of a "superatom" in ref. [3]: "A structure sub-unit may be deemed a superatom possessing any number of free valences". Thus, in the GRAPH representation at the top of fig. 4, the free valences are designated by + in the GRAPH(2) row, i.e. they are of +-type. Here, each fragment is represented as a separate structure with n valences of the first atom and $n - 1$ valences of the other atoms in the GRAPH(1) row. The remaining single valence is in GRAPH(2), thus forming the bonding within the fragment. Further, the same approach is applied to the fragments by considering them "superatoms". The first fragment is represented in the GRAPH(1) row by all n free valences of its atoms, and the other fragments by

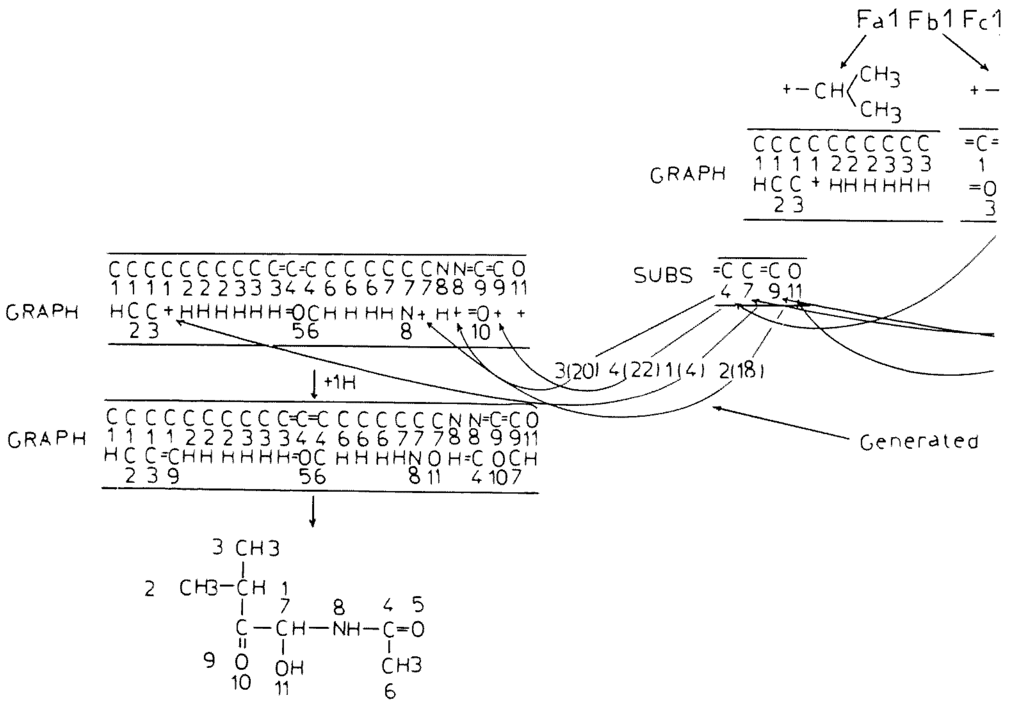
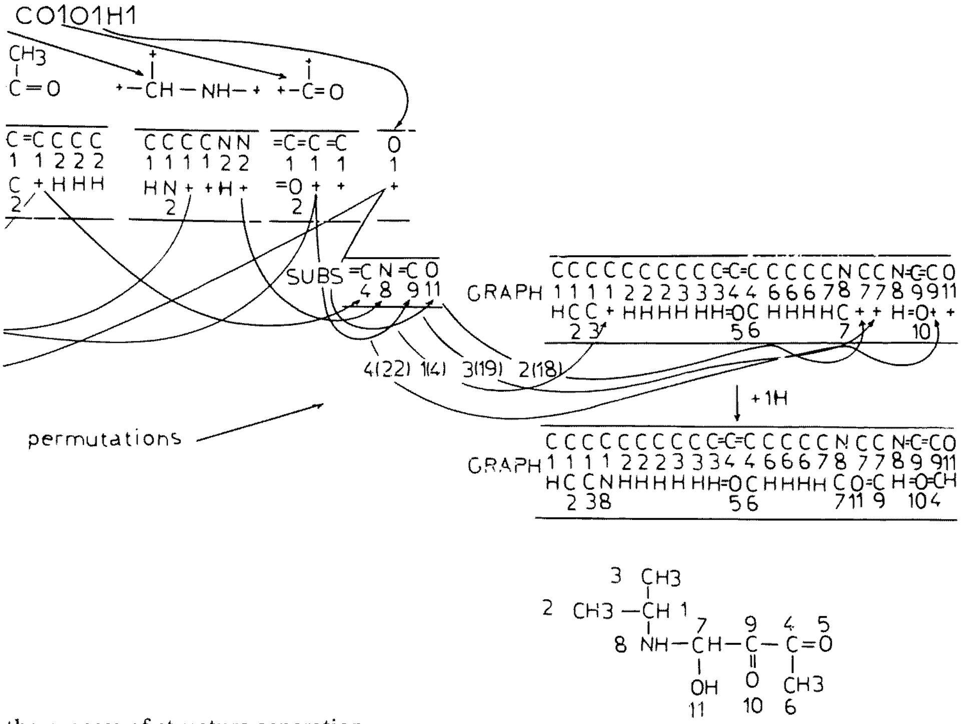


Fig. 4. Employment of fragments in

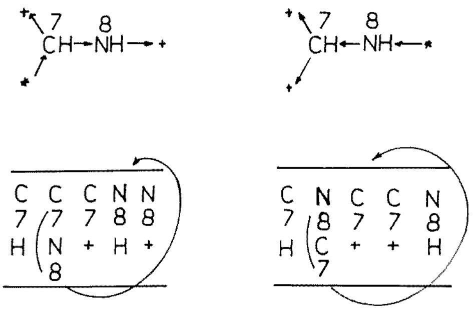
$n - 1$ free valences, one valence being transformed from +-type to ⋆-type, i.e. it is transferred from the GRAPH to the SUBS array. The generation of structures consistent with the fragments in the gross formula is carried out in the same manner, as discussed in the previous section: m out of all L +-type valences are selected (here, L is no longer the GRAPH array dimension) and saturated by m ⋆-type SUBS valences.

However, a problem with the choice of the free valence which is to be transformed from +-type to ⋆-type emerges. As previously mentioned, the single-atom valences are equivalent, but this is no longer the case for the fragment-free valences. For instance, if the only free valences of atoms 4 and 8 from fig. 4 are transformed to ⋆-type, they will never form a bond (remember a bond is formed between +-type and ⋆-type valences). Hence, all the free valences of the second, third, etc. fragments should be selected in turn as ⋆-type valences. This is illustrated in fig. 4, where two isomers are generated, the first one with a ⋆-type valence for the *Fc* fragment taken from atom 7, and the second with a ⋆-type valence for the same fragment taken from atom 8. Note that after choosing a valence as ⋆-type (e.g. that of atom 8), a rearrangement of the GRAPH array is carried out so that it remains a directed graph. Figure 4



the process of structure generation.

indicates that on changing the ★-type valence from atom 7 to atom 8, the corresponding part of the GRAPH array is rearranged as follows:



6. Unique numbering and coding of the structures

Most of the canonical numbering and coding schemes reported in the literature [14,15] could be adapted to our approach. However, we intended to develop a numbering scheme free from permutations. As mentioned at the beginning of this paper, the ^{13}C chemical shift/charge density linear relationship is employed for the subsequent step of chemical inference. Thus, we have atom charge densities computed by a fast semi-empirical method (either the method of Del Re [17] or IPEOE [16] are used in our program). We therefore use those values for ranking the separate atoms within the generated structure. This is carried out by assigning a rank (R) to each of the atoms, calculated according to the equation:

$$R = R_0 + N_H - Qa,$$

where R_0 are initial ranks, listed in table 1 for different atoms, N_H is the number of hydrogen atoms attached to the corresponding atom, and Qa is its charge density. This approach is based on the fact that the methods used by us for charge calculations are dependent upon the toplogy, where the charges and ranking values are the same for isomorphic structures. On the other hand, they are not integer values, and the reliability of the numbering scheme depends on the precision of their evaluation (we found a threshold precision value of 0.00001 to be appropriate). It should be admitted that there is no proof that this method is reliable in all cases, but for the aims of our program, it gives satisfactory results. The reliability and precision problems require wider discussion, which will be done in a subsequent paper.

Table 1

Initial ranks (R_0) for the different atoms employed in the structure generation program

Atom	Rank
C	1
=C	5
⋆C	8
N	10
=N	13
⋆N	15
O	16
=O	18
S	19
=S	21

The use of charge densities allows an easy detection of the constitutional symmetry: equivalent atoms have the same charges within the given precision. However, an additional canonization procedure had to be developed for their discrimination and numbering. The process of numbering is illustrated in fig. 5. One can see that

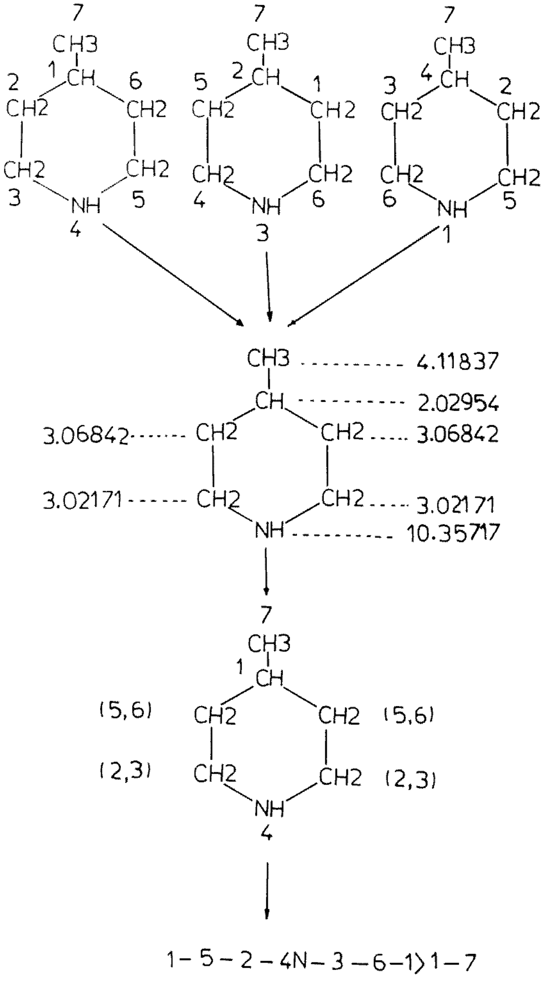


Fig. 5. Numbering and coding of the generated structures (Del Re [17] charges used).

there are two pairs of equivalent atoms (2, 3) and (5, 6). Instead of a single number, a list of all equivalent atom numbers, ordered in a lexicographical order, is assigned to each of the equivalent atoms. The allocation of these numbers is carried out by constructing different walks in the course of the transformation of our two-row array representation into a linear structural code (search with backtracking procedure used).

The following rule is applied: the earlier an equivalent atom is encountered along the walk, the smaller the unused number from the list, that is assigned to it, is. The procedure is developed in such a way that the walks are constructed in a uniform way for all the structures.

Finally, the structure is transformed into a linear notation code, shown in fig. 5. We use the OGRA-30 [18] structure presentation language developed in Novosibirsk (USSR). In forming the code, C, CH, CH_2 and CH_3 are represented by their numbers only, and the other atoms and groups by numbers and a chemical description. The only differences are that the symbols "=", "$\star$" and ":" are used for double, triple and conjugated bonds, respectively ("−" is used for a single bond as it is in OGRA-30), and the symbol > is used for a new branch of the structure.

7. Further development

As stated at the beginning of this paper, our approach to structure elucidation imposes on the generator the need for tracing all the connectivities within the set of atoms given by the gross formula. Consequently, this is conducive to redundancy of structures. Evidently, such an approach will be less attractive for other applications. Our algorithm, however, allows a further modification to substantially lower this redundancy or completely eliminate it.

The duplicated structures of the first class as classified in the beginning are easily detectable, since they appear after permuting two atoms or groups bonded to one and the same atom. A common feature of the duplicated structures of the second class is that they are due to permutations of equivalent groups. Note that the resulting structure after such a permutation is not in all cases a duplicate of the preceding one. Since the ${}^mP_L$ permutations, in our program, are generated by generating ${}^mC_L$ combinations of m taken from L elements, and for each combination $m!$ permutations are generated, a single permutation, either of the first or of the second class, leads to $(m-2)!$ permutations resulting in duplicated structures. Hence, after detecting such a permutation, all $(m-2)!$ permutations should be simply skipped. Thus, the number of permutations could be sharply reduced.

A more straightforward procedure for eliminating the redundancy of the second class is the prohibition of permutations between equivalent atoms or groups. The atoms in the SUBS array are ranked according to the initial ranks R_0 provided in table 1, and to their hydrogen deficiency (obtained from the ^{13}C-H multiplicity of the ^{13}C NMR spectra). The fragment valences in the SUBS array are ranked separately, taking into account their connectivities within the fragment. Thus, only permutations between $\star$-type valences of atoms having different ranks are carried out.

The development of this version of our approach is in progress, and the details of the algorithm will be reported in a forthcoming paper.

8. The program

This algorithm is implemented in the program STRGEN (as part of the ASEC13 structure elucidation program), written in BASIC for a Hewlett-Packard computer. Translation of the program into FORTRAN 77 is in progress.

References

[1] I.P. Bangov, MATCH 14(1983)235.

[2] D.B. Nielson, M.E. Munk, K.B. Gash and D.B. Herald, Jr., J. Organ. Chem. 34(1969)3800.

[3] L.M. Masinter, N.S. Sridharn, J. Lederberg and D.H. Smith, J. Amer. Chem. Soc. 96(1974) 7702;
R.E. Carhart, D.H. Smith, H. Brown and N. Sridharn, J. Chem. Inf. Comp. Sci. 15(1975)124;
R.E. Carhart, D.H. Smith, H. Brown and C. Djerassi, J. Amer. Chem. Soc. 97(1975)5755.

[4] Y. Kudo and S. Sasaki, J. Chem. Inf. Comp. Sci. 16(1976)43;
H. Abe, T. Okuyama, I. Fujiwara and S. Sasaki, J. Chem. Inf. Comp. Sci., in press;
S. Sasaki and Y. Kudo, J. Chem. Inf. Comp. Sci. 25(1985)252.

[5] J.E. Dubois and H. Viellard, Bull. Soc. Chim. Fr. 3(1968)900;
J.E. Dubois and H. Viellard, Bull. Soc. Chim. Fr. 3(1971)839;
J.E. Dubois, H. Viellard and A. Panaye, Bull. Soc. Chim. Fr. 6(1973)1988;
A. Panaye and J.E. Dubois, in: *Proc. 9th Int. CODATA Conf.*, Jerusalem, Israel, ed. P.S. Glaeser (North-Holland, Amsterdam, 1985);
J.E. Dubois and Y. Sobel, J. Chem. Inf. Comp. Sci. 25(1985)326.

[6] J.V. Knop, W.R. Miller, Z. Jericević and N. Trinajstić, J. Chem. Inf. Comp. Sci. 21(1981)91;
J.V. Knop, W.R. Miller, K. Szymanski and N. Trinajstić, Computer generation of certain classes of molecules, SKTH/Kemija u industriji, Zagreb (1985).

[7] Y. Kudo and S. Sasaki, J. Chem. Doc. 14(1974)200;
I.A. Faradjev, in: *Algorithmic Investigations in Combinatorics*, ed. I.A. Faradjev (Nauka, Moscow, 1978) p. 11 (in Russian).

[8] I.P. Bangov, Comm. Dep. Chem. 17(1984)459 (in Bulgarian);
I.P. Bangov, Mikrochim Acta (Wien) (1986) 281.

[9] N. Christofides, *Graph Theory. An Algorithmic Approach* (MIR, Moscow, 1978) (in Russian).

[10] V. Pelligrin, J. Chem. Ed. 60(1983)626.

[11] W.T. Wipke and T.M. Dyott, J. Chem. Inf. Comp. Sci. 15(1975)140.

[12] J.B. Hendrickson, D.L. Grier and A.G. Toczko, J. Chem. Inf. Comp. Sci. 24(1984)195.

[13] A.T. Balaban, Rev. Roumaine Chim. 1(1966)1097;
A.T. Balaban, ibid. 15(1970)463;
A.T. Balaban, ibid. 17(1972)865;
A.T. Balaban, ibid. 18(1973)635.

[14] H.L. Morgan, J. Chem. Doc. 5(1965)107;
C. Jochum and J. Gasteiger, J. Chem. Inf. Comp. Sci. 23(1983)171;
W. Bremser, Anal. Chim. Acta 103(1978)355;
R.C. Read, J. Chem. Inf. Comp. Sci. 23(1983)23;
R.C. Read, ibid. 25(1985)116.

[15] O. Mekenyan, D. Bonchev and A.T. Balaban, J. Comp. Chem. 5(1984)629;
D. Bonchev, O. Mekenyan and A.T. Balaban, MATCH 18(1985)83;
A.T. Balaban, O. Mekenyan and D. Bonchev, J. Comp. Chem. 6(1985)538;
O. Mekenyan, D. Bonchev and A.T. Balaban, ibid. (1985)552;
A.T. Balaban, O. Mekenyan and D. Bonchev, ibid. (1985)562;
N. Ralev, S. Karabunrliev, O. Mekenyan, D. Bonchev and A.T. Balaban, ibid. (1985)578.

[16] J. Gasteiger and M. Marsili, Tetrahedron 36(1980)3219;
M. Marsili and J. Gasteiger, in: *Data Processing in Chemistry,* Summer School in Rzeszow, Poland (1980), ed. Z. Hippe (PWN/Elsevier, Warsaw – Amsterdam, 1981) p. 56.

[17] G. Del Re, J. Chem. Soc. (1958)4031.

[18] A.A. Kochetova, V.A. Skorobogatov and P.V. Hvorstov, V'chislitelnie sistemi 92(1982)70 (in Russian).