Structure Generation of Constitutional Isomers from Structural Fragments

SIMONA BOHANEC* and JURE ZUPAN
Boris Kidrič Institute of Chemistry, Hajdrihova 19, YU61001 Ljubljana, Yugoslavia

Received April 26, 1991

A structure generator for the generation of all possible chemical structures from any set of structural fragments is described and discussed. The generator is exhaustive, efficient, and eliminates the duplicate structures. It considers structural constraints. The input fragments can be of any size. Our algorithm is compared to some other generators[1-15] with respect to the resulting output structures and required generation times. The generator can be used on IBM PC XT/AT personal computers and compatibles with Hercules/EGA/VGA graphic card. The generator is implemented in Turbo Pascal version 5.5.

INTRODUCTION

Chemical information and expert systems[1-11,16,17] must handle large numbers of spectroscopic and structural chemical data. The structure elucidation process determines the structure of an unknown chemical compound, "the sample", by using the information stored in the knowledge base and spectral data from the sample. Usually, at the beginning the chemist guesses or predicts only a few structural fragments and tries to connect them together. Linking fragments together in a number of different ways is a combinatorial process, the result of which is a set of all possible structures generated from a given set of fragments. The combinatorial process is very time-consuming because all possible combinations must be tried **and** checked against all constraints very carefully. This can be done efficiently only by computer programs called structure generators.

A good structure generator must be **exhaustive** (all possible structures should be generated from a given set of fragments), **irredundant** (the generation of identical structures should be avoided), and **effective** (the generation should be fast and should perceive the connections that lead to chemically or topologically impossible, identical, or nonconnected structures as soon as possible during the generation process).

The generator should consider all structural constraints before, during, and after the generation process. Due to the fact that during a **fast** generation process the generation of identical structures cannot be completely avoided, elimination of duplicates must be included at the end of the process.

The number of output structures and the speed of the generation process depend on the number and type of input fragments. In the case of completely different and **nonsymmetrical** fragments each new combination of connections (bonds) among fragments leads to a new output structure. On the other hand, if the input set consists of many identical and/or highly symmetrical fragments, many identical structures will be generated. Therefore, a test of the symmetry of new fragments which are generated during the generation process should be performed on **each** new generated fragment (i.e., after each new bond is established).

The more complex (unique or nonsymmetric) the input fragments are the easier the problem of symmetry is and vice versa. For example, the presence of a number of small identical input fragments (like -H) which can be 'distributed' on carbons and/or on the heteroatoms in many different ways leading to many identical structures causes a so called "hydrogen allocation problem". In order to deal with this problem, the following approaches have been devised in various chemical systems:

> Checking all fragments for consistency with the composition and various spectral data (^{13}C NMR, IR, mass spectra). CHEMICS[3-5] and ISOGEN,[12] for example, consider **all sets** of fragments having hydrogens bonded to carbons and heteroatoms in **all possible ways**.
>
> Hydrogens are not explicitly represented (ASSEMBLE[6,7]). Free bonds of fragments are reserved for connections with hydrogens **after** initial symmetry analysis has been done.
>
> Hydrogens are connected to the free bonds in the **final** structure (CONGEN[13,14]).
>
> All monovalent atoms (not only hydrogens) are connected to higher valent atoms in the **preliminary** step (GENOA[15]).

Another very complex structural problem involves generation of cyclic structures from large numbers of identical fragments. In this case, the analysis of cyclic skeletons (vertex graphs[17,18]) is used to avoid the generation of duplicate structures.

The authors of DENDRAL,[2,13-15,17] CHEMICS,[3-5] CASE,[6-10] and Robien's system[11] have constantly improved structure generators together with their systems. For example, the first structure generator in the DENDRAL was the "Dendral algorithm".[2] It was used for the generation of acyclic structures only. The next generator called StrGen[2] has linked the "Dendral algorithm" and vertex graph catalogues for generation of cyclic and acyclic organic compounds. Dendral's structure elucidation system CONGEN[13-15] based on the concept of superatoms (input fragments of different sizes are regarded as large multivalent "atoms" called superatoms) has used the StrGen[2] generator for generation of structures from superatoms and an "embedding" algorithm for the connection of remaining free bonds on the same superatom. The most recent structure generator of the DENDRAL called GENOA[15] is based upon a standard graph matching technique.

From the CASE system two structure generators, the earlier ASSEMBLE[6,7] and the later COCOA,[10] are worth mentioning. The ASSEMBLE generates structures on the basis of molecular formula (MF). The generation starts with a set of nonconnected atoms defined by the MF and combines them together into larger and larger fragments using spectral data as constraints. These larger fragments are taken from the sets of standard fragments. All fragments have additional restrictions about their incorporation into the final structure. To put the fragments together the generator uses a connection table **and** the valence constraints. Besides, the structural constraints are used to reduce the number of possible combinations of fragments. Similar to ASSEMBLE is the generator in the CHEMICS[3-5] system, which uses fragments with quite broadly defined restrictions.

Compared to ASSEMBLE, the generator COCOA[10] uses a different strategy for the generation. In addition to the standard set of nonoverlapping structural fragments, the input may consist of overlapping fragments as well. In the first step of the generation process, the generator builds up a hyperstructure. The hyperstructure is a structure in which all fragments are connected with all connections that could be considered in building a structure. In the second step a re-

0095-2338/91/1631-0531$02.50/0

duction (removal) of bonds takes place. After the reduction of each connection, the hyperstructure is tested if it satisfies the structural constraints.

Still another algorithm for the generation of structures from a MF is elaborated in the ISOGEN[12] system. The structures are generated from the set of one-atom fragments with allocated hydrogens. All possible connections of these fragments are described with a connection table having specifically ordered connections. Any chemically meaningful set of connections among fragments yields a new possible final structure. This system does not use structural constraints during the generation to reduce the number of possible structures.

Our generator GEN generates structures from arbitrary fragments (not only from predefined fragments like CHEMICS) of any size (not only one-atom fragments as ISOGEN does). GEN uses many different types of input data (molecular formula—MF, molecular weight, fragments of different sizes defined using spectral data, etc.) which can be regarded as constraints. It has to be emphasized that GEN is designed for use on personal computers (not on large computer systems or mini/microcomputers like the generators and systems DENDRAL, CHEMICS, ASSEMBLE, and COCOA).

The GEN's concept for the generation of structures tries to incorporate as many different features as possible from all contemporary programs that we regarded as beneficial to this process and at the same time to exclude some of the drawbacks of these generators. The common features are

Hyperstructure in COCOA corresponds to "basic connection matrix" in GEN

Controlled combinations of connections in the connection matrix (ISOGEN and GEN)

Determination of fragments from molecular formula (ISOGEN and GEN)

Similar description of structural constraints used in CHEMICS and in GEN

GENERATOR

GEN generates structures from any set of fragments of different size by consideration of chemical and topological constraints. For example, the generation of all structures having the molecular formula C_3H_7OBr starts with three >C< fragments, seven -H, one -O-, and one -Br. There are four possible structures:

(a) CH_3-CH_2-CH_2-O-Br
(b) CH_3-CH_2-O-CH_2-Br
(c) CH_3-O-CH_2-CH_2-Br
(d) OH-CH_2-CH_2-CH_2-Br

If the generation is constrained by the fragment -CH_2-CH_2- (not allowed in the final structure), then the result should be structure b only. If the fragment -CH_2-Br is supposed to be present, then the resulting structures are b, c, and d. If the fragment -CH_3 must be present while -O-Br must be absent in the final structure, the resulting structures are b and c.

The generation is based on the **connection matrix**. The generated structures (final or resulting structures) must consist of **all** fragments from the set. If no structures can be generated from a given set, then smaller subsets with less fragments can be used. The fragments and generated structures are written in a form of **connection table**[19] which is used to describe the topological connectivity of atoms in the whole or partial structures.

Connection Matrix. The connection matrix E (Figure 2) used in GEN will be described with respect to the adjacency matrix M (Figure 1) used in the generators ISOGEN, CHEMICS, and ASSEMBLE.

In the adjacency matrix M, the ith row and the ith column correspond to the ith fragment (Figure 1). The element $e_{i,j}$ represents the connection linking the ith and jth fragments.

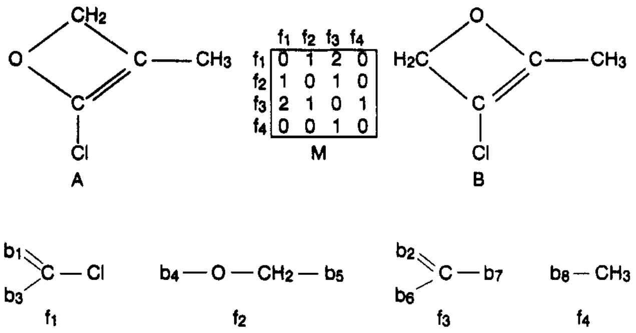


Figure 1. From four fragments labeled f_1, f_2, f_3, and f_4, two structures (A and B) represented with the same adjacency matrix M can be generated.

Single, double, and triplet **connections** (bonds) are represented by the values 1, 2, and 3 of $e_{i,j}$, respectively. The value 0 means no bond. The connections among N fragments (N = 4 in Figure 1) can be represented by the $N \times N$ adjacency matrix.

Normally, each fragment has **more** than one bond with which it can be connected to other fragments. They are called **free bonds**. Free bonds can be single, double, or triple and should be connected to non-hydrogen atoms that are on the **same** or **different** fragments. From Figure 1 it can be seen that single free bonds b_6 and b_7 are connected to the same atom of the fragment f_3, while b_4 and b_5 are connected to two different atoms (C and O) of the fragment f_2. Single free bonds b_4 and b_5 in fragment f_2 are **not** equivalent (are not in the equal environment **within** the fragment). Fragments f_2 and f_3 can be linked in two different ways obtaining two different substructures (larger fragments). By linking free bonds b_5 and b_6 a part of the structure A is obtained, while by linking free bonds b_4 and b_6 a part of the structure B is obtained. The same substructures are obtained by linking free bond b_7 instead of b_6 because b_6 and b_7 are equivalent.

From Figure 1 it can be seen that **one** adjacency matrix M represents **two** different structures (A and B). To establish how many different structures can be obtained from the same adjacency matrix **M**, some additional tests of the equivalency of the free bonds b_i and about the symmetry of fragments must be performed.

In order to enable a uniform representation of structures by the adjacency matrices and to avoid symmetry tests, matrices of a different type are used in the GEN generator (Figure 2). In these matrices, called the connection matrices E, each row and each column correspond to exactly one free bond. The connection matrix E can be divided into matrices E_1, E_2, and E_3 containing the single, double, and triple free bonds, respectively. The size of the matrix E_1 is $B_1 \times B_1$ where B_1 is a number of single free bonds (six on Figure 2), while B_2 and B_3 are the numbers of double and triple free bonds in all input fragments, respectively. All matrices are symmetrical across the main diagonal. The elements $e_{i,j}$ of the matrices E_1, E_2, and E_3 have values T (true) and F (false). If the bond between the bonds in the ith row and jth column is possible, the element $e_{i,j}$ is T, otherwise it is F. If all elements of the connection matrix have T values, the matrix is called a **full-connection matrix**. An **empty connection matrix** is a matrix with all $e_{i,j}$ having F values. We prefere labeling of bonds with T and F over labeling with 1 and 0, respectively. The label '1' can easily be mixed up with the representation for a single bond.

Before the generation process, the pairs of connection matrices (E_{01}, E_1), (E_{02}, E_2), and (E_{03}, E_3) for single, double, and triple bonds, respectively, must be set up. The matrices E_{0i} are called **basic connection matrices** containing **all** chemically and topologically possible connections (Figure 4), while E_i are empty matrices.

structure A $e_{1,2} = e_{3,4} = e_{5,6} = e_{7,8} = T$

$$\underset{E(A)}{\begin{array}{c|cccccccc|} & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 \\ \hline b_1 & F & T & & & & & & \\ b_2 & T & F & & & & & & \\ b_3 & & & F & T & F & F & F & F \\ b_4 & & & T & F & F & F & F & F \\ b_5 & & & F & F & F & T & F & F \\ b_6 & & & F & F & T & F & F & F \\ b_7 & & & F & F & F & F & F & T \\ b_8 & & & F & F & F & F & T & F \\ \hline \end{array}} \longrightarrow \underset{E_2(A)}{\begin{array}{c|cc|} & b_1 & b_2 \\ \hline b_1 & F & T \\ b_2 & T & F \\ \hline \end{array}} \quad \underset{E_1(A)}{\begin{array}{c|cccccc|} & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 \\ \hline b_3 & F & T & F & F & F & F \\ b_4 & T & F & F & F & F & F \\ b_5 & F & F & F & T & F & F \\ b_6 & F & F & T & F & F & F \\ b_7 & F & F & F & F & F & T \\ b_8 & F & F & F & F & T & F \\ \hline \end{array}}$$

structure B $e_{1,2} = e_{3,5} = e_{4,6} = e_{7,8} = T$

$$\underset{E(B)}{\begin{array}{c|cccccccc|} & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 \\ \hline b_1 & F & T & & & & & & \\ b_2 & T & F & & & & & & \\ b_3 & & & F & F & T & F & F & F \\ b_4 & & & F & F & F & T & F & F \\ b_5 & & & T & F & F & F & F & F \\ b_6 & & & F & T & F & F & F & F \\ b_7 & & & F & F & F & F & F & T \\ b_8 & & & F & F & F & F & T & F \\ \hline \end{array}} \longrightarrow \underset{E_2(B)}{\begin{array}{c|cc|} & b_1 & b_2 \\ \hline b_1 & F & T \\ b_2 & T & F \\ \hline \end{array}} \quad \underset{E_1(B)}{\begin{array}{c|cccccc|} & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 \\ \hline b_3 & F & F & T & F & F & F \\ b_4 & F & F & F & T & F & F \\ b_5 & T & F & F & F & F & F \\ b_6 & F & T & F & F & F & F \\ b_7 & F & F & F & F & F & T \\ b_8 & F & F & F & F & T & F \\ \hline \end{array}}$$

Figure 2. Representation of the structures A and B generated from fragments f_1, f_2, f_3, and f_4 (see Figure 1) using the connection matrices $E(A)$ and $E(B)$, respectively. The connection matrices $E(A)$ and $E(B)$ are divided into two pairs of connection matrices $E_1(A)$, $E_2(A)$, and $E_1(B)$, $E_2(B)$, each pair representing single, E_1, and double bonds, E_2.

During the generation the empty matrices E_i are filled with connections defined in basic matrices. Each new connection, adding a T element into E_i, means a generation of a new link between two free bonds from the set. The generation is completed when there is exactly **one** T element in **every** row and column of the matrices E_i.

The matrices E_i with exactly one T element in each row and column are called the **connection matrices containing the structure**. All possible structures are generated from a set of fragments after all possible valid distributions of T elements defined in E_{0i} are examined. For the fragments in Figure 1 two different combinations of T elements and hence two different pairs of connection matrices [$E_1(A)$, $E_2(A)$] and [$E_1(B)$, $E_2(B)$] containing structures A and B are obtained (Figure 2). To form a structure in this example, eight free bonds (two double and six single ones) must be linked together making four connections (for each connection two free bonds are required).

The advantages of the use of connection matrices E_i in GEN, compared to the adjacency matrix M used in other generators, are the following:

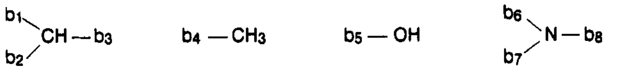
(a) The size of the input fragments is not important. Only the number of fragment's free bonds has an effect on the generation time.

(b) The filling of the matrix E_i with T elements during the generation process is very fast because no additional tests are needed.

However, there are two disadvantages with the use of matrices E_i in GEN compared to the use of matrices M in other generators:

(a) Up to three matrices (E_1, E_2, and E_3) must be used in GEN for determining only one structure. The number of matrices depends on the free bond types of fragments.

(b) Matrices E_i are normally larger (particularly E_1), compared to the matrices M, because each fragment can have several free bonds.



1 CH	b_1 1	b_2 1	b_3 1	3
2 CH_3	b_4 1			1
3 OH	b_5 1			1
4 N	b_6 1	b_7 1	b_8 1	3

(a)

1 CH	2 1	b_2 1	b_3 1	3
2 CH_3	1 1			1
3 OH	b_5 1			1
4 N	b_6 1	b_7 1	b_8 1	3

(b)

1 CH	2 1	b_1 1	b_2 1	3
2 CH_3	1 1			1
3 OH	b_3 1			1
4 N	b_4 1	b_5 1	b_6 1	3

(c)

Figure 3. Connection table for four fragments -CH<, -CH_3, -OH, and -N< (a), connection table after the MPSF >CH-CH_3 is considered (b), and after the renumbering of free bonds (c).

Structural Constraints. The generator GEN distinguishes between structural features that **must be present** and those that **must be absent** in the resulting structure. *The mandatory present structural features* (MPSF) are not allowed to overlap, while *the mandatory absent structural features* (MASF) can. Both types of structural features are used as constraints before, during, and after the generation process. Both types of constraints are represented as structural fragments.

The generator can consider up to eight MPSF and up to eight MASF. The MPSFs reduce the connection matrix, i.e., the number of possible connections among the fragments. As a consequence, the generation time is also reduced.

The MASF can contain one, two, or more **non-hydrogen** atoms. If MASF is detected **within** any fragment in the set, the generation process is immediately suppressed. If present, the MASF containing only one non-hydrogen atom can **always** be identified in the initial fragments. The MASFs with **two** non-hydrogen atoms can (additionally beside being in the initial fragment) reduce the connection matrix by prohibiting the connections that could produce the structures leading to MASF. The presence of more complex MASF not detected in initial fragments (rings for example) is examined in the final output structures.

The inclusion of structural constraints has considerable effect on the reduction of generation time, on the number of possible connections, and on the number of output structures. Finally, the reduction of the number of output structures causes the reduction of the time needed for the elimination of the duplicate structures.

Generation. The structures are generated from the set of fragments. The resulting structures must consist of **all** fragments from a given set and must comply with all constraints. The generation process consists of the following four steps:

(1) making the connection table[19] from the fragments of a given set and consideration of MPSF and small MASF

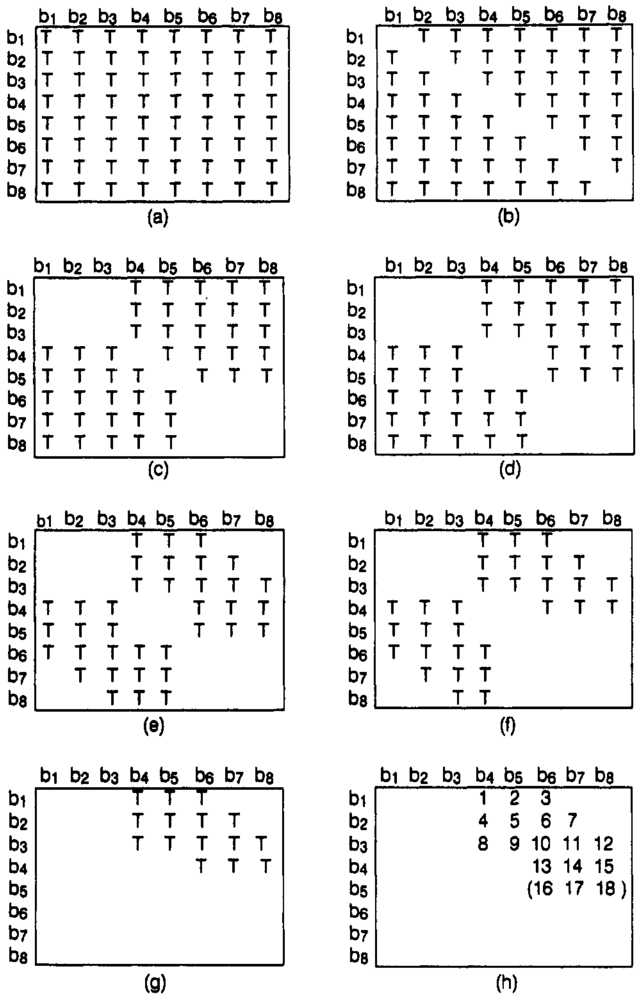
(2) determination of basic connection matrices E_{0i}, the elimination of forbidden and/or chemically nonreasonable connections, and connections that lead to MASF

(3) finding identical connections in the basic connection matrices E_{0i}

(4) finding matrices containing new structures and elimination of structures with MASF

In the first three steps the data for the fourth step are prepared. In the fourth step the real generation is performed. In the following paragraphs all four steps will be explained in detail.

(ad 1) All input fragments with all free bonds are written in the form of a connection table (see Figure 3a for the case of



	b1	b2	b3	b4	b5	b6	b7	b8
b1	T	T	T	T	T	T	T	T
b2	T	T	T	T	T	T	T	T
b3	T	T	T	T	T	T	T	T
b4	T	T	T	T	T	T	T	T
b5	T	T	T	T	T	T	T	T
b6	T	T	T	T	T	T	T	T
b7	T	T	T	T	T	T	T	T
b8	T	T	T	T	T	T	T	T

(a)

	b1	b2	b3	b4	b5	b6	b7	b8
b1		T	T	T	T	T	T	T
b2	T		T	T	T	T	T	T
b3	T	T		T	T	T	T	T
b4	T	T	T		T	T	T	T
b5	T	T	T	T		T	T	T
b6	T	T	T	T	T		T	T
b7	T	T	T	T	T	T		T
b8	T	T	T	T	T	T	T	

(b)

	b1	b2	b3	b4	b5	b6	b7	b8
b1				T	T	T	T	T
b2				T	T	T	T	T
b3				T	T	T	T	T
b4	T	T	T		T	T	T	T
b5	T	T	T	T		T	T	T
b6	T	T	T	T	T			
b7	T	T	T	T	T			
b8	T	T	T	T	T			

(c)

	b1	b2	b3	b4	b5	b6	b7	b8
b1				T	T	T	T	T
b2				T	T	T	T	T
b3				T	T	T	T	T
b4	T	T	T			T	T	T
b5	T	T	T			T	T	T
b6	T	T	T	T	T			
b7	T	T	T	T	T			
b8	T	T	T	T	T			

(d)

	b1	b2	b3	b4	b5	b6	b7	b8
b1				T	T	T		
b2				T	T	T	T	
b3				T	T	T	T	T
b4	T	T	T			T	T	T
b5	T	T	T			T	T	T
b6	T	T	T	T	T			
b7		T	T	T	T			
b8			T	T	T			

(e)

	b1	b2	b3	b4	b5	b6	b7	b8
b1				T	T	T		
b2				T	T	T	T	
b3				T	T	T	T	T
b4	T	T	T			T	T	T
b5	T	T	T					
b6	T	T	T	T				
b7		T	T	T				
b8			T	T				

(f)

	b1	b2	b3	b4	b5	b6	b7	b8
b1				T	T	T		
b2				T	T	T	T	
b3				T	T	T	T	T
b4						T	T	T
b5								
b6								
b7								
b8								

(g)

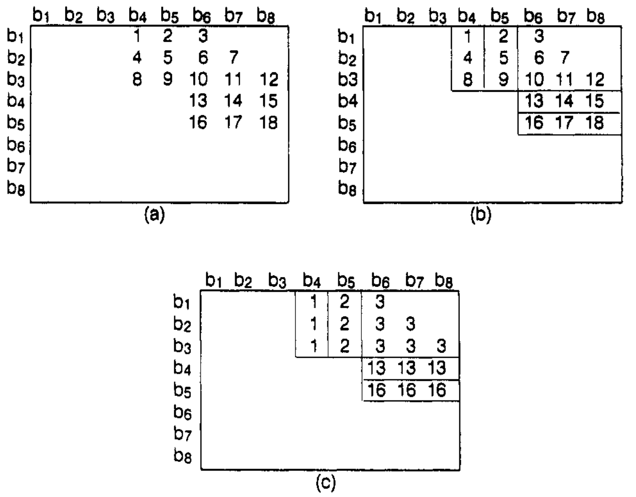
	b1	b2	b3	b4	b5	b6	b7	b8
b1				1	2	3		
b2				4	5	6	7	
b3				8	9	10	11	12
b4						13	14	15
b5						(16	17	18)
b6								
b7								
b8								

(h)

Figure 4. Determination of the basic connection matrix E_{01} for fragments from Figure 3 and the elimination of meaningless or forbidden connections: the full connection matrix (a), the connection matrix without connections between the same bonds $e_{i,j}$ (b), the matrix without the connections between the free bonds on the same atom (c), the matrix without the connections which lead to the small molecules (the element $e_{4,5}$) (d), the matrix without the meaningless connections between the first and all other atoms (e), the connection matrix after the consideration of the MASF)N-OH (f), the matrix with the connections above the diagonal only (g), and the connection matrix (without the consideration of MASF) with enumerated connections (h). The false values, F, (representing no bonds) are omitted from display.

fragments -CH<, -CH$_3$, -OH, and -N<). Each row of the connection table contains full information about one non-hydrogen atom: its identification number (ID), chemical symbol of the atom, number of attached hydrogens, number of free bonds, and number of non-hydrogen neighbors. Each free bond is presented together with the label and with its type. Only the atoms with **at least one free bond** are used in the generation process.

Next, the free bonds of atoms leading to MPSF are connected. For example: if a fragment >CH-CH$_3$ is a MPSF, then the free bonds b_1 and b_4 are connected to form this fragment. If there is more than one equivalent free bond on the atom, always the **first** adequate one is used to form the MPSF substructure. Therefore, the bond b_1 on the C atom (ID = 1) is used and not b_2 or b_3 on the same atom. The fixed link between the atoms 1C and 2C is shown in Figure 3b. The labels are substituted by the ID numbers of atoms, which are attached to the bonds (labels b_1 and b_4 become ID numbers 2 and 1, respectively). As a consequence of this operation, the size of the connection matrix E_1 is reduced and the remaining free bonds are renumbered (Figure 3c).



	b1	b2	b3	b4	b5	b6	b7	b8
b1				1	2	3		
b2				4	5	6	7	
b3				8	9	10	11	12
b4						13	14	15
b5						16	17	18
b6								
b7								
b8								

(a)

	b1	b2	b3	b4	b5	b6	b7	b8
b1				1	2	3		
b2				4	5	6	7	
b3				8	9	10	11	12
b4						13	14	15
b5						16	17	18
b6								
b7								
b8								

(b)

	b1	b2	b3	b4	b5	b6	b7	b8
b1				1	2	3		
b2				1	2	3	3	
b3				1	2	3	3	3
b4						13	13	13
b5						16	16	16
b6								
b7								
b8								

(c)

Figure 5. Basic connection matrix, E_{01} with numbered connections (a), with grouped identical connections (b), and the connection matrix with EQ values (c).

(ad 2) As mentioned, for each free bond of type i a separate basic connection matrix E_{0i} is made. Once formed, the matrices E_{0i} are used in step 4 of the generation process.

The formation of the basic connection matrix E_{0i} starts as a full matrix (Figure 4a). First, the forbidden and chemically unreasonable connections are 'removed'. The removal is done by turning the matrix elements corresponding to the forbidden connections to the 'false' state F. Such elimination, in the case of the four fragments from Figure 3, is shown in Figure 4. The meaningless connections are

Free bonds connnected to themselves (the main diagonal) (Figure 4b).

Free bonds connected on the same atom (the connections $e_{1,2}$, $e_{1,3}$, $e_{2,3}$, $e_{6,7}$, $e_{6,8}$, and $e_{7,8}$ in Figure 4c that connect the free bonds b_1, b_2, and b_3 on atom 1C and free bonds b_6, b_7, and b_8 on atom 1N).

Bonds yielding structures that do not contain **all** fragments from the set [the connection $e_{4,5}$ are eliminated, because they disconnect two fragments from the graph leading to the molecule of methanole (Figure 4d)].

All connections $e_{i,j}$ of any equivalent bond b_i on the first atom to all equivalent bonds b_j on the other atom that have larger sequence number than bond b_i. The sequence numbers for bonds b_1, b_2, and b_3 on atom 1C are equivalent to bonds b_6, b_7, and b_8 on atom 4N and have the sequence numbers 1, 2, and 3, respectively. So, the bond b_1 (with sequence number 1) can be connected only to the bond b_6 (with sequence number 1), bond b_2 (sequence number 2) to b_6 or b_7 (sequence number 2), and bond b_3 (sequence number 3) to b_6, b_7, or b_8 (sequence number 3). Therefore, the connections $e_{1,7}$, $e_{1,8}$, and $e_{2,8}$ are removed (Figure 4e).

If MASFs are given, the program eliminates the connections that are leading to the generation of structures with MASFs. For example, if the fragment >N-OH is given as an MASF, the connections $e_{5,6}$, $e_{5,7}$, and $e_{5,8}$ are removed (Figure 4f). The connection matrix is symmetrical across the main diagonal. During the generation only the connections above the diagonal are considered (Figure 4g). After the elimination of the meaningless connections the **basic connection matrix** E_{0i} is obtained (Figure 4g). The remaining connections are numbered row by row (Figures 4h and 5a). The numbers are the identification numbers (ID) of these connections.

(ad 3) The connections of a given free bond to any equivalent bond of **another** atom are treated as identical and grouped together (Figure 5b). The identical connections are labeled with a unique number (identity parameter EQ). The identity parameter EQ corresponds to the smallest ID number from

Table I. Procedure Called 'filling_of_matrix' Which Describes the Algorithm for Filling the Matrix with Connections

```
procedure filling_of_matrix;

  var
    p : integer;                        (* pointer to the last added connection *)
    a : array (1..Bo) of integer;                (* ID of added connections *)

  begin
   initialization(p, a);
    repeat
      while not ok_connec(a(p)) then n_connec(a(p));
      if not ok_cond then begin p_pointer(p); n_connec(a(p)) end
      else
      if new_comb then
        begin
          if new_struc then save_struc;
          p_pointer(p); n_connec(a(p))
        end
      else
        begin n_pointer(p); n_connec(a(p)) end;
    until all_comb
  end;
```

all connections in the observed group (Figure 5c).

(**ad 4**) The number of connections, B_o, required for linking all fragments into the final structure depends on the number of all free bonds on fragments from the entire set, B_{all}. B_{all} and B_o are calculated as follows:

$$\sum_{i=1}^{N} b_i(\text{single}) + \sum_{i=1}^{N} b_i(\text{double}) + \sum_{i=1}^{N} b_i(\text{triple}) = B_1 + B_2 + B_3 = B_{all} \quad (1)$$

$$B_o = 1/2 B_{all} = 1/2(B_1 + B_2 + B_3)$$

where b_i(single), b_i(double), and b_i(triple) are the numbers of single, double, and triple free bonds of the fragment i; N is the number of all fragments in the set; B_1, B_2, and B_3 are the total numbers of single, double, and triple free bonds in all fragments, respectively.

During the generation the empty matrix E_i of the same size as the basic connection matrix E_{oi} is filled with the connections that give new structures. Each bond type i is processed within its own basic connection matrix E_{oi}. At the end, the connection matrices E_1, E_2, and E_3 should contain exactly $B_1/2$, $B_2/2$, and $B_3/2$ connections, respectively.

As an example, the set of four fragments shown in Figure 3 and 5 is worked out in detail. These four fragments have eight single free bonds ($B_1 = 8$) and no double or triple bonds. These fragments require four single connections ($B_1/2 = 4$) to form a structure. In the basic connection matrix E_{o1} (Figure 5a) there are 18 possible connections. This means that out of 18 possible connections only four must be distributed into the **empty** matrix E_1 to obtain all valid final structures. To find **all** final structures 3060 combinations (combinations of the fourth order of 18 elements) without any constraints or any control of intermediate results are possible and should be tried.

In order to avoid combinatorial explosion the **controlled combinatorial process** is used. The algorithm for filling the matrix is described with the procedure named 'filling_of_matrix' (Table I). The procedure contains several procedures (initialization, n_connec, p_connec, p_pointer, save_struc) and functions (ok_connec, ok_cond, new_comb, new_struc, all_comb) to control the combinatorial process. The IDs of connections are saved into array a during the generation. The number of connections that must be added to form the complete structures is B_o. The entire generation process is repeated in loops from **repeat** to **until**. One loop is called a pass.

Each new connection $e_{i,j}$ (associated with a pointer p) is tested in the function named 'ok_connec'. This function returns TRUE value if the connection $e_{i,j}$ satisfies the following **conditions**:

(1) No element in the ith row and the jth column of the connection matrix has the T value.

(2) The ith row must be the first row with no T elements (starting from the upper to the lower position over the connection matrix).

(3) $e_{i,j}$ must have the **same or larger EQ** than the previously established connection (with pointer $p - 1$).

If all three of the above conditions are satisfied, the tested connection ID is saved in the pth place of the array a(p), otherwise the next connection ID + 1 (procedure n_connec) for the pointer p is considered for testing.

After a new connection is made, a new fragment is generated. The atoms in the new fragment and in the remaining ones are tested (function ok_cond) for three conditions:

(1) If the number of remaining free bonds on one atom is smaller or equal to the number of connections that should be added to obtain the final structure.

(2) If the number of atoms having one free bond is smaller or equal to the number of connections that should be added to obtain the final structure.

(3) If a structure contains all fragments from the set.

If the fragment has passed the above three tests, at the pointer p a new connection is established. If then, in each row and column of filling matrices E_i **exactly one** T element exists and p is equal to B_o, a new structure is obtained (function new_comb is TRUE). The resulting structure is tested by the function new_struc for connectedness, presence of all MPSF, and absence of all MASF and is compared with already generated structures to avoid duplications.

The test for connectedness can be regarded as a walk through the graph. The structure is connected if all atoms starting from any atom can be visited. To test the presence of MPSF and the absence of MASF and to eliminate the duplicate structures, the eliminating routine based on the substructure search[19,20] is used. The new structures are put (save_struc) on the list of generated structures.

The next combination of connections (p_pointer and n_connec) is examined after each structure is found.

An example for the generation of structures from fragments (Figure 3) using the connections from Figure 5 is shown in Table II. The entire generation process consists of 13 passes through the algorithm given in Table I. In order to form a final structure four connections must be made linking these fragments ($B_o = 4$). After the second pass, **three** free bonds (b_6, b_7, and b_8) on one atom (atom 4N) and only **two** connections to form a new structure remain, consequently, condition 1 for stopping the pass is met. In the third pass, the second connection, ID 5, in the connection matrix is exchanged with ID 6. The same procedure as in the second pass is repeated at the fourth one. During the 6th and 10th pass, two structures are generated. In the 13th pass a small molecule HCN is generated leaving the number of fragments with one free bond (-CH_3 and -OH) larger than the number of needed connections (2 vs 1, respectively). The conditions 2 and 3 for stopping the generation are fulfilled and the process stops. Because no other connections satisfy the rules, the generation is completed. Each free bond already used in the generation process is labeled with an 'X'.

ENVIRONMENT OF THE GENERATOR GEN

The environment of the generator GEN consists of the routines for data input, for the location of hydrogen atoms on carbons and other atoms, for the selection of sets of fragments, and for the elimination of duplicates. The input fragments for the generation are determined and/or selected by using different input data. At the moment, two types of input data

Table II. Structure Generation Process from Fragments >CH–, –CH$_3$, –OH, and >N– (Figure 3)

pass	ID of added connections	used free bonds b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	function	results
1	1	X			X					ok_cond	T
2	1 5	X	X		X	X				ok_cond	F (cond. 1)
3	1 6	X	X		X		X			ok_cond	T
4	1 6 9	X	X	X	X	X	X			ok_cond	F (cond. 1)
5	1 6 11	X	X	X	X		X	X		ok_cond	T
6	1 6 11 18	X	X	X	X	X	X	X	X	ok_cond	T
										new_struc	HO–N=CH—CH$_3$
7	2	X				X				ok_cond	T
8	2 6	X	X			X	X			ok_cond	T
9	2 6 11	X	X	X		X	X	X		ok_cond	T
10	2 6 11 15	X	X	X	X	X	X	X	X	ok_cond	T
										new_struc	HO—CH=N—CH$_3$
11	3	X					X			ok_cond	T
12	3 7	X	X				X	X		ok_cond	T
13	3 7 12	X	X	X			X	X	X	ok_cond	F (cond. 2,3)

Table III. Lists of Built-in-Fragments[a]

list A				list B	list C		
–CH$_3$	>C<	=O	–NH$_2$	–F	C8-ring	8-ring	=
–CH$_2$–	=C<	–CO–	–NH–	–Cl	C7-ring	7-ring	≡
=CH$_2$	=C=	–CHO	–N<	–Br	C6-ring	6-ring	
=CH–	≡C–	–COO–	–N=	–I	C5-ring	5-ring	
>CH–	–OH	–COOH	≡N		C4-ring	4-ring	
≡CH	–O–	–CN	Ph-ring		C3-ring	3-ring	

[a] Fragments from lists A and C can be chosen as structural constraints, while the fragments from lists A and B, called also 'standard fragments', can be chosen as the input fragments for the generation.

are used: molecular formula (MF) and fragments obtained from the spectroscopic and analytical data. The sequence in which the above routines are used depends on the type of the input. The fragments determined from the MF are less narrowly defined than the fragments obtained from various other sources (spectroscopic and analytical data). The fragments determined from the MF do not have a specified number of hydrogen atoms, exactly defined free bonds, and bond types. On the other hand, the fragments determined from the spectroscopic and analytical data are usually larger, have a specified number of hydrogen atoms, and free bonds. Because of these differences the input fragments must be treated differently before they entered the generation process in GEN. The following paragraphs describe two sequences of routines for preparing the fragments and for the generation of structures based on the two previously mentioned types of input data.

If the MF is used as the main input, the structural constraints should be given as well. Otherwise, the combinatorial explosion prevents exhaustive generation for larger MFs.

MPSF and MASF structural constraints can be input by the structure editor or selected from a list of build-in fragments (Table III, lists A and C). The selected fragment can be additionally changed by using the structure editor. For example: if the C5-ring (cyclopentane) fragment is selected from list C (Table III), it could simply be changed to a pyrrole ring by changing one of the carbon atoms into the nitrogen (see Figure 6).

For enhancing the description of structural constraints, two different labels for free bonds are available. Free bonds labeled 'A' can be connected to any atom **including** hydrogen while free bonds labeled 'X' can be connected to any atom **except** to the atom (or fragment) from which it starts and to the hydrogen atom. Two free bonds labeled 'X' from **one** fragment are not allowed to be linked to the **same** atom on another fragment. Single, double, or triple free bonds that are not allowed to change their character must be labeled with an 'X'.

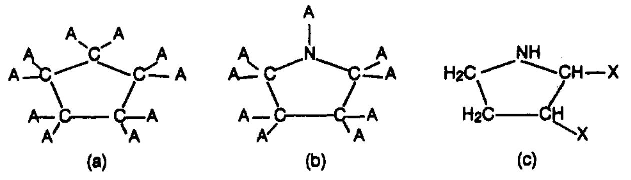


Figure 6. Using the structure editor, the C5-ring (a) selected from list C (Table III) is changed into the pyrrole ring (b) and further into the 2,3-disubstituted pyrrole ring (c). Lables 'A' and 'X' mean the substitution of any atom and any atom except hydrogen are allowed, respectively.

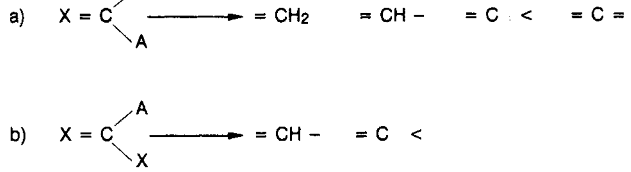


Figure 7. All possible appearances of the MPSF fragments in the resulting structure. If both single bonds are labeled with 'A's (a) the MPSF is defined very general, while the case of one single bond labeled with 'A' and another with 'X' (b) represents more strictly defined MPSF.

Otherwise the single bonds can be linked to any other atom (including hydrogen) and/or merged to form double or triple bonds. For example: carbon atom with two double bonds connected to two non-hydrogen atoms is defined as fragment X=C=X. Two fragments from Figure 7 consist of carbon atoms with one defined double bond ('X') and two single bonds (in the first fragment both are 'A', but in the second one is 'A' and another is 'X') that can be linked to any atom or changed. The possible parts of the resulting structures defined with fragments on the left are shown in Figure 7.

Structural constraints are used in the selection of fragments for determining sets and in the generation. The fragments that constitute a set are used directly in the generator GEN. From these fragments all possible structures are generated. In the generation process the structural constraints are used three times:

(1) When making the connection table where MPSF with well-defined single bonds (free bonds labeled with 'X') are considered (step 1).

(2) When setting up the basic connection matrices where the connections which lead to the MASF fragments are eliminated (step 2).

(3) When the testing of every new structure if it has all MPSF and no MASF fragments. The tests are made on constraints which are not considered at points 1 and 2 above (step 4).

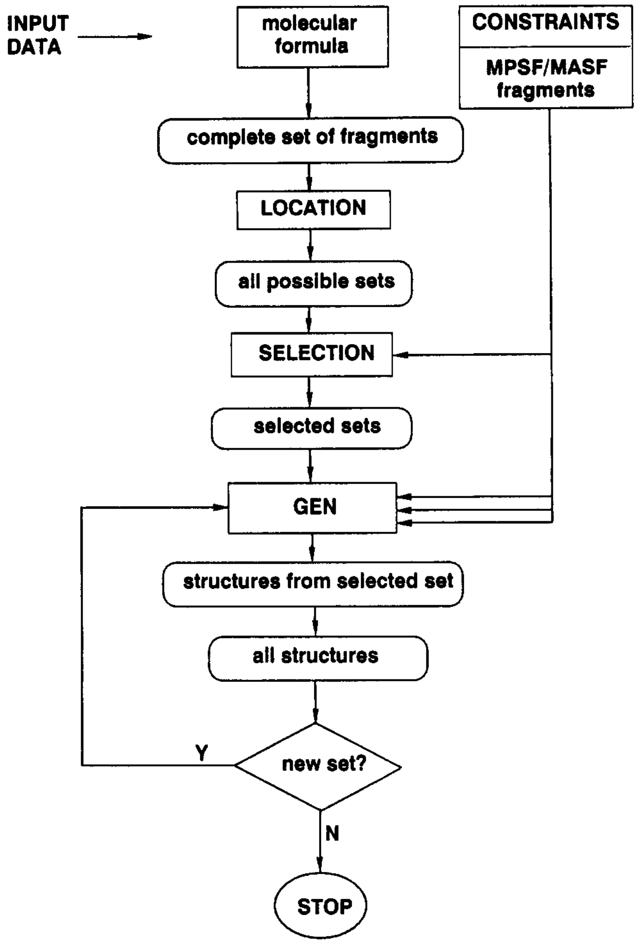


Figure 8. Procedure sequence in which the program obtains fragments for GEN using the molecular formula.

The sequence of the routines for the generation of structures from the MF using structural constraints is given in Figure 8.

After the MF is input and atomic centered fragments (fragments in the complete set in Figure 9) are formed, the substitution of hydrogens on the free bonds takes place. The hydrogens are distributed in **all** possible ways hence producing **all** possible sets. Figure 9 show how four hydrogens can be placed on the three carbons and on one oxygen in six different ways. Therefore, six different possible sets are obtained. Among these sets, the selection of sets for the generation is made by the program taking into account the MPSF and MASF. To be able to generate at least one structure from the selected set of fragments, the sum of free bonds of equal type must be even.

If the sought structures are radicals (i.e., they are supposed to have one or more free bonds), then the sum of free bonds minus the sum of radicals must be even.

According to the structural constraints and the above condition, only two sets, 4 and 6 (Figure 9), are selected. From each of these sets two structures are generated. The structures generated from different sets of fragments are **always** different because they consist of different atomic fragments: different number of hydrogens are linked on the equivalent central atoms. Hence, no additional elimination of duplicate structures is necessary.

If the MF is given as the input, only the fragments with single free bonds can be determined at the beginning (Figure 9). However, during the process of linking free bonds into the connection table of final structure the connections on the same atoms are merged into double or further into triple bonds.

For a generation of structures from a set of arbitrary fragments a different sequence of routines (compared to that used for generation from MF) as shown in Figure 10 is applied. The input are fragments and constraints. For the fragments in the complete set different data sources can be used:

standard fragments built into the system (Table III, lists A and B)

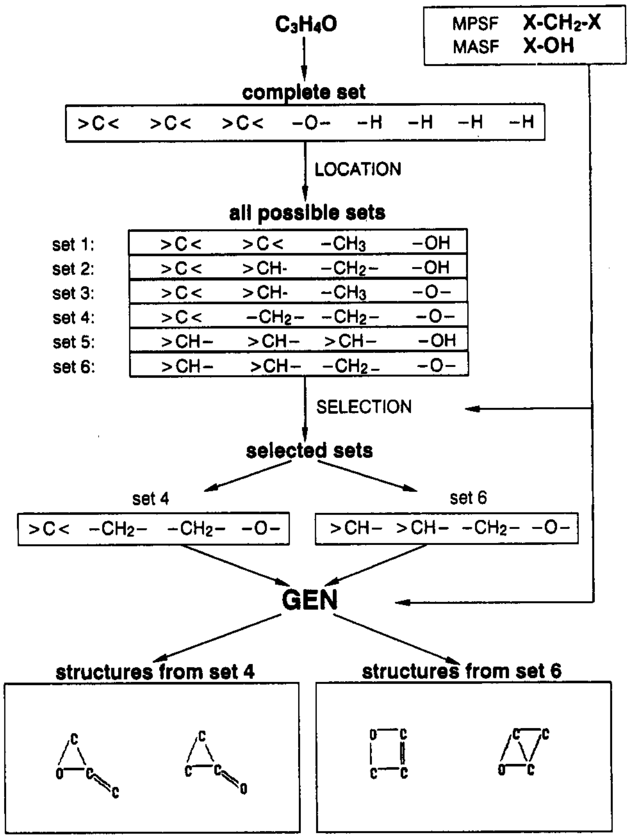


Figure 9. Example of structure generation from the molecular formula C_3H_4O using X-CH_2-X and X-OH as MPSF and MASF, respectively.

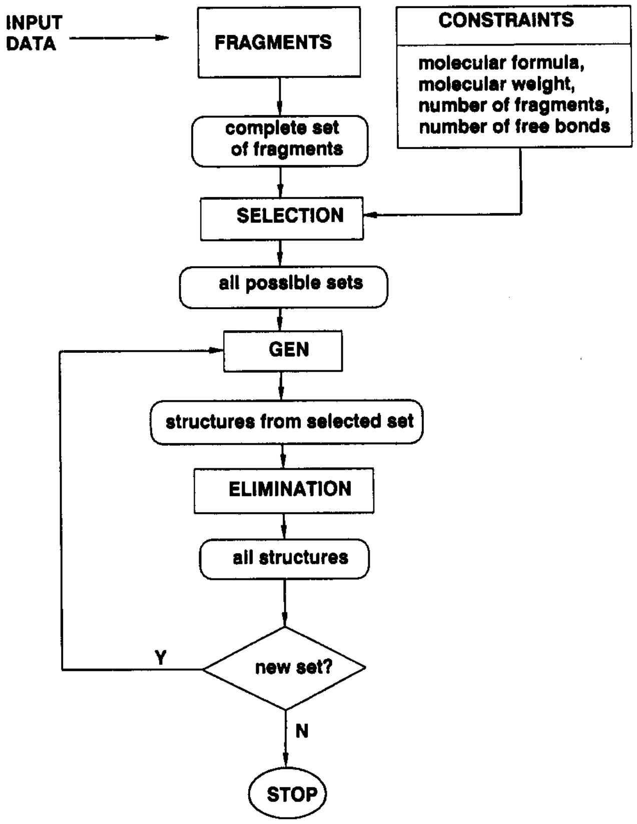

INPUT DATA
FRAGMENTS
CONSTRAINTS
molecular formula, molecular weight, number of fragments, number of free bonds
complete set of fragments
SELECTION
all possible sets
GEN
structures from selected set
ELIMINATION
all structures
Y
new set?
N
STOP

Figure 10. Sequence in which the structures are generated from exactly defined fragments.

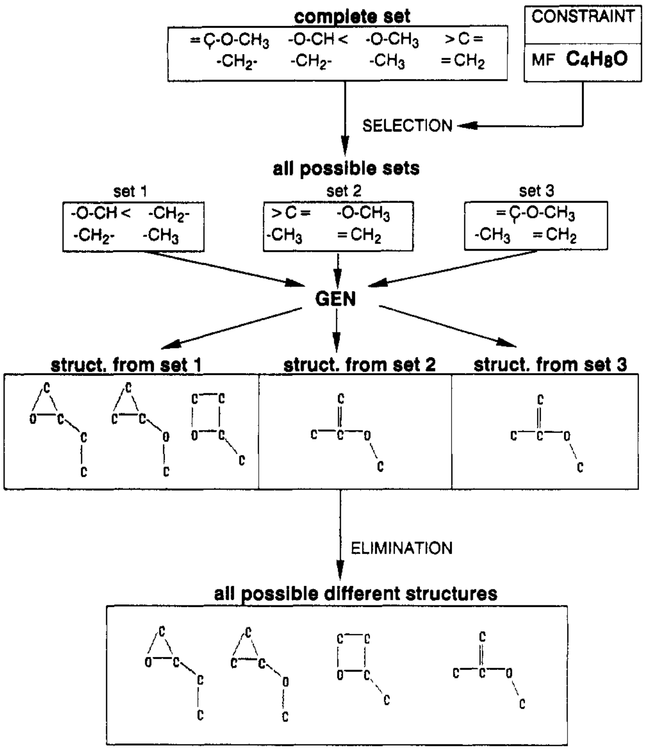


Figure 11. Example of structure generation from fragments determined by the CARBON system and using the molecular formula C_4H_8O as a constraint.

fragments built with the structure editor[19,20]

fragments obtained from different spectral data using the interpretational programs like CARBON[16]

The fragments in the complete set, if given by the user, have usually well-defined free bonds (marked with 'X'). For example, the connection between two **defined** single free bonds are normally not supposed to merge into a connection forming a double bond.

As the constraints for the selection of the fragments from the complete set of form an actual (i.e., the possible set) the following data can be used:

molecular weight of the target structures

approximate or exact MF (the number of atoms like C, N, O, S, H, and halogens) of the target structures

the number of fragments in the set for generation

the number and the type of free bonds if the target structures are radicals

The fragments for all possible sets are selected automatically from the complete set on the basis of constraints. If there are many constraints, only a few sets can be produced. Before the generation, the automatically selected sets are displayed to the user for confirmation or rejection.

The structures are generated from all fragments of each selected set. In the resulting structures the input fragments do not overlap.

In the present case of automatic selection of fragments (to the contrary of the case of MF input) it is possible that the same structures can be generated from different sets. To remove duplicates, the eliminating routine is used at the end. This routine is identical with the eliminating routine in the generator GEN for testing the presence and absence of structural constraints in the generated structure.

Figure 11 shows an example of the generation of structures from fragments obtained by the CARBON system using the ^{13}C NMR spectrum (20.4, 53.9, 80.4, and 160.2 ppm) taken from the literature[21] as input. The eight most probable fragments were selected for the complete set. From this set three possible sets using the MF (obtained by the element analysis of the unknown compound) as a constraint were found. From the first, second, and third set three, one, and one structure are generated, respectively. The structures obtained from the second and third set are identical because the fragment $\geqslant$C-O-CH_3 in the third set consists of two fragments, >C= and -O-CH_3, from the second set. As a consequence, identical structures are generated. The eliminating routine detects the duplicates and eliminates one of them.

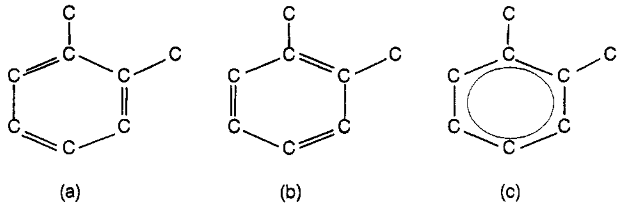


Figure 12. Two canonic forms of 1,2-dimethylbenzene with localized single and double bonds (a, b) and 1,2-dimethylbenzene with resonant bonds (c).

In order to determine the most probable structure, the simulation program SIMULA[22] for simulation of the ^{13}C NMR spectra was applied on all generated structures. The structure yielding the simulated spectrum (20.8, 51.5, 76.7, and 163.5 ppm for fourth structure of all possible structures in Figure 11) that matches the above experimental spectrum most closely was selected as the resulting one.

CONCLUSION

The generator GEN was developed for the generation of structures using data obtained from different types of spectroscopies. To find out the quality and the efficiency of the generator GEN the comparisons with other generators have been made.

The comparison of the results given by generators or systems DENDRAL,[23,24] MOLGRAPH,[24] CHEMICS,[5,24] ASSEMBLE,[24] and GEN shows that most of them generate an equal number of structures using the same molecular formula as input (Table IV, section a). By employing identical structural constraints (Table IV, section b) the same structures were obtained with the generators MOLGRAPH and GEN.

The comparison of efficiency was made between the generator in the system CHEMICS and the generator GEN (Table IV, section c).

The generator GEN does not distinguish resonant or conjugated bonds from the localized single and double free bonds that alternate. For example, 1,2-dimethylbenzene can be written in two canonic forms (Figure 12a,b) or in the form of the resonant bonds (Figure 12c). The generator considers them to be different.

As a consequence, generator GEN generates more structures because all canonic forms are regarded as different.

This problem will be solved in the next version by the introduction of a basic connection matrix E_{0r} and a connection matrix E_r for resonant bonds. The eliminating routine for the elimination of duplicate structures will also be changed to find the places where delocalization can arise.

The generator GEN was designed for personal computers, and so it is easily transportable to where the chemists can use it most profitably. The use of the generator and all surrounding programs is simple and easy to learn. The generator has a content-dependent HELP option. All operations are available over menus. The fragments, structural constraints, and output structures are displayed as two-dimensional graphs on the screen.

The use of the connection matrix E instead of the adjacency matrix M enables the consideration of majority of structural constraints, even before the actual generation starts. This in turn causes a reduction in the number of possible connections among the fragments and the reduction of the generation time (see Table IV, section c). The GEN program considers ad-

Table IV[a]

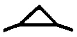
Section a

mol formula	DENDRAL	MOLGRAPH	CHEMICS	ASSEMBLE	GEN
$C_2H_5NO_2$	84	84	87	84	84
C_4H_7NO	764	764	802	764	764
C_3H_4BrCl		10	8		10
C_5H_8BrCl		140	108		140
$C_6H_{10}O$	747	747	745		747

Section b

			no. of structures	
mol formula	MPSF	MASF	MOLGRAPH	GEN
$C_8H_{16}O_2$	O═C—O	O═C—OH	105	105
$C_6H_{11}NO$	-CN		64	64
$C_6H_{11}NO$	-CN, -OH		31	31

Section c

				CPU time (s)				
	$\mathrm{MPSF}^{(P)}$	no. of struct			GEN			
mol formula	or $\mathrm{MASF}^{(A)}$	CHEM	GEN	CHEMICS	XT[b]	AT[c]	286[d]	386[e]
$C_4H_5O_2Cl$	/	907	907	840	9007	5967	3631	907
C_4H_7NO	/	802	764	1436	1524	898	490	162
$C_4H_5O_2Cl$	$-CO-CH_3^{(P)}$ $-CO-Cl^{(P)}$	1	1	6	1.9	1.7	0.4	0.2
C_4H_7NO	$-NH-CO-^{(P)}$ $-CO-NH_2^{(A)}$	5	5	34	11	8	3	2
C_3H_7NO	/	87	84	52	28	17	10	8
C_3H_7NO	$-NH_2^{(P)}$  (A)	25	25	44	19	14	8	7

[a] Number of isomer structures with the same molecular formula as those obtained with system/generator DENDRAL, MOLGRAPH, CHEMICS, ASSEMBLE, and GEN (section a) and implementing the same MF and the same structural constraints on the generators MOLGRAPH and GEN (section b). The comparison of the generation time for the generation of various structures between CHEMICS and GEN are given in section c. The computations were made on an MV/2000DC computer and on personal computers, respectively. All free bonds in MPSFs and MASFs are treated to be labeled as X's. Therefore, both free bonds in X-NH-CO-X are not allowed to be linked to the same atom, that excludes the formation of any three-atom ring containing the X-NH-CO-X MPSF as a part of it. [b] PC XT, CI = 1.7, clock rate = 1.4 MHz. [c] PC AT, CI = 6.3, clock rate = 6 MHz. [d] PC AT 286, CI = 15.9, clock rate = 16 MHz. [e] PC AT 386, CI = 27.4, clock rate = 41.8 MHz.

ditional constraints like molecular weight (if the MF is not given), the number of fragments that form the final structure, and/or the number of free bonds for the case of substructure generation. Substructures generated in some of the previous runs can be used as input fragments for the generation of further (more complex) structures. In the MF composed of standard atom symbols and numbers, the user can include in the same manner any 'substructure' labeled by two characters. If such 'atom symbol' is not known, the system asks for its valency, i.e., for the number of free bonds. The group -CO-OH with a 'valence' number equal to 1 can be labeled as a 'CG', for example, and included into the molecular formula. The structure generation is limited by the disk space because all generated structures are stored on it. For about 1000 structures the generator needs about 1.1 Mbyte of space. For saving more structures, correspondingly more space should be available. The generator with the environment needs additional 0.5 Mbyte of space.

As shown by the example in Figure 11, the generator GEN can play a very important role as a part of a large structure elucidation system based on various spectroscopic data. The generator links fragments predicted from different parts of such a system into the final structures. Among them, the most probable structures according to the spectral data can be selected by using the programs for spectra simulations like SIMULA.[22]

The generator GEN is useful also in the mass spectrometry. It can predict the structures of fragments from peaks in mass spectrum. Each peak determines a set of fragments with defined mass. From certain mass, one or more molecular formulas can be predicted. The generator generates fragments from these molecular formulas and links them together obtaining the possible structures belonging to or producing the query spectra.

The generator GEN is written in Turbo Pascal 5.5 programming language. It works on IBM PC XT/AT/286/386 personal computers and compatibles. For drawing chemical structures and fragments the Turbo Pascal standard graphic routines are used.

REFERENCES AND NOTES

(1) Gray, N. A. B. *Computer-Assisted Structure Elucidation*; John Wiley & Sons: New York, 1986.

(2) Gray, N. A. B. Dendral and Meta-Dendral—The Myth and the Reality. *Chem. Intell. Lab. Sys.* **1988**, *5*, 11–32.

(3) Oshima, T.; Ishida, Y.; Saito, K.; Sasaki, S. Chemics-UBE, A Modified System of Chemics. *Anal. Chim. Acta* **1980**, *122*, 95–102.

(4) Abe, H.; Okuyama, T.; Fujiwara, I.; Sasaki, S. A Computer Program for Generation of Constitutionally Isomeric Structural Formulas. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 220–229.

(5) Funatsu, K.; Miyabayaski, N.; Sasaki, S. Further Development of Structure Generation in the Automated Structure Elucidation System CHEMICS. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 18–28.

(6) Shelley, C. A.; Hays, T. R.; Munk, M. E.; Roman, R. V. An Approach to Automated Partial Structure Expansion. *Anal. Chim. Acta* **1978**, *103*, 121–132.

(7) Shelley, C. A.; Munk, M. E. CASE, a Computer Model of the Structure Elucidation Process. *Anal. Chim. Acta* **1981**, *133*, 507–516.

(8) Munk, M. E.; Farkas, M.; Lipkus, A. H.; Christie, B. D. Computer-Assisted Chemical Structure Analysis. *Mikrochim. Acta* **1986**, 199–215.

(9) Christie, B. D.; Munk, M. E. The Application of Two-Dimensional Nuclear Magnetic Resonance Spectroscopy in Computer-Assisted Structure Elucidation. *Anal. Chim. Acta* **1987**, *200*, 347–361.

(10) Christie, B. D.; Munk, M. E. Structure Generation by Reduction: A New Strategy for Computer-Assisted Structure Elucidation. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 87–93.

(11) Robien, W. Computer-Assisted Structure Elucidation of Organic Compounds III: Automatic Fragment Generation from ^{13}C NMR Spectra. *Mikrochim. Acta* **1986**, 271–279.

(12) Zhu, S.; Zhang, J. Exhaustive Generation of Structural Isomers for a Given Empirical Formula—A New Algorithm. *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 34-38.

(13) Masinter, L. M.; Sridharan, N. S.; Lederberg, J.; Smith, D. H. Applications of Artificial Intelligence for Chemical Inference. 12. Exhaustive Generation of Cyclic and Acyclic Isomers. *J. Am. Chem. Soc.* **1974**, *96*, 7702-7714.

(14) Carhart, R. E.; Smith, D. H.; Brown, H.; Djerassi, C. Application of Artificial Intelligence for Chemical Inference. 17. An Approach to Computer-Assisted Elucidation of Molecular Structure. *J. Am. Chem. Soc.* **1975**, *97*, 5755-5762.

(15) Carhart, R. E.; Smith, D. H.; Gray, N. A.; Nourse, J. G.; Djerassi, C. GENOA: A Computer Program for Structure Elucidation Utilizing Overlapping and Alternative Substructures. *J. Org. Chem.* **1981**, *46*, 1708-1718.

(16) Zupan, J.; Novič, M.; Bohanec, S.; Razinger, M.; Lah, L.; Tušar, M.; Košir, I. Expert System for Solving Problems in Carbon-13 Nuclear Magnetic Resonance Spectroscopy. *Anal. Chim. Acta* **1987**, *200*, 333-345.

(17) Carhart, R. E.; Smith, D. H.; Brown, H.; Sridharan, N. S. Applications of Artificial Intelligence for Chemical Inference. 16. Computer Generation of Vertex-Graphs and Ring Systems. *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 124-131.

(18) Nakayama, T.; Fujiwara, Y. Structure Generation on the Basis of BCT Representation of Chemical Structures. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 218-223.

(19) Zupan, J. *Algorithms for Chemists*; John Wiley & Sons: Chichester, 1989.

(20) Zupan, J.; Bohanec, S. Creation and Use of Chemical Data Bases with Substructure Search Capability. *Vestn. Slov. Kem. Drus.* **1987**, *34* (1), 71-81.

(21) Kalinowski, H. O.; Berger, S.; Braun, S. *Carbon-13 NMR Spectroscopy*; John Wiley & Sons: Chichester, 1988.

(22) Lah, L.; Tušar, M.; Zupan, J. Simulation of ^{13}C NMR spectra. *Tetrahedron Comput. Methodol.* **1989**, *2* (2), 5-15.

(23) Lederberg, J.; Sutherland, G. L.; Buchanan, B. G.; Feigenbaum, E. A.; Robertson, A. V.; Duffield, A. M.; Djerassi, C. Application of Artificial Intelligence for Chemical Inference. 1. The Number of Possible Organic Compounds. Acyclic Structures Containing C, H, O, and N. *J. Am. Chem. Soc.* **1969**, *91*, 2973-2976.

(24) Kerber, A.; Lane, R.; Moser, D. Ein *Strukturgenerator für molekulare* Graphen. *Anal. Chim. Acta* **1990**, *235*, 221-228.

Subductive and Inductive Derivation for Designing Molecules of High Symmetry

SHINSAKU FUJITA

Ashigara Research Laboratories, Fuji Photo Film Company, Ltd., Minami-Ashigara, Kanagawa, 250-01 Japan

Received May 14, 1991

Two types of derivations of molecules, subductive and inductive, are presented to design molecules of high symmetry. A subductive derivation consists of substituting a parent molecule with another set of atoms; on the other hand, an inductive derivation is composed of substituting a parent skeleton with a set of ligands that have three-dimensional structure. These derivations are discussed on the basis of subduction and induction of coset representations.

Since van't Hoff founded organic stereochemistry,[1] three-dimensional molecular models have been employed to comprehend stereochemical phenomena. Although the physicochemical explanation of chemical bonding has been changing, the importance of such models remains invariant. Thus, many organic molecules of high symmetry, achiral and chiral, have been designed and synthesized on the basis of molecular models.[2] Farina and Morandi[3] have proposed a principle for designing such molecules, in which a parent molecule of higher symmetry is desymmetrized with appropriate substituents. However, no theoretical rationalization has been reported for the desymmetrization process. We have presented the concept of promolecule for characterizing stereochemical relationships in nonrigid molecules.[4] This concept is also effective for rigid molecules; we can, however, discuss their symmetrical properties without employing the concept. The latter treatment is capable of rationalizing the Farina–Morandi proposal comprehensively; this issue is the object of the present paper. Thus, we propose here subductive and inductive derivations, which are the general methodologies for designing molecules of high symmetry.

SUBDUCTIVE DERIVATION OF MOLECULES

We first explain a minimum set of notations concerning subduction of coset representations.[5] A molecule is considered to be a derivative of a parent molecule, where the hydrogens of the parent are replaced by another set of atoms. Let $\mathbf{G}$ be the point group that represents the symmetry of the parent molecule. A set of equivalent atoms (or substituents) in the parent molecule

$$\Delta = \{\delta_1, \delta_2, ..., \delta_r\}$$

constitutes an orbit that is assigned to a coset representation (CR) $\mathbf{G}(/\mathbf{G}_i)$, where $\mathbf{G}_i$ is a subgroup of $\mathbf{G}$. The notation $\mathbf{G}(/\mathbf{G}_i)$ comes from a coset decomposition represented by

$$\mathbf{G} = \mathbf{G}_i g_1 + \mathbf{G}_i g_2 + ... + \mathbf{G}_i g_r \qquad (1)$$

where each g_k represents a representative of the coset $\mathbf{G}_i g_k$. Let us consider the set of the cosets

$$\mathbf{G}/\mathbf{G}_i = \{\mathbf{G}_i g_1, \mathbf{G}_i g_2, ..., \mathbf{G}_i g_r\} \qquad (2)$$

When we apply a symmetry operation ($g \in \mathbf{G}$) to each of the cosets, we have a permutation

$$\pi_g = \begin{pmatrix} G_i g_1 & G_i g_2 & \cdots & G_i g_r \\ G_i g_1 g & G_i g_2 g & \cdots & G_i g_r g \end{pmatrix} \qquad (3)$$

Suppose that g runs over all of the symmetry operations of $\mathbf{G}$. The resulting set of permutations constructs the CR of $\mathbf{G}$, i.e.

$$\mathbf{G}(/\mathbf{G}_i) = \{\pi_g | \ \forall \ g \in \mathbf{G}\} \qquad (4)$$

The assignment of Δ to $\mathbf{G}(/\mathbf{G}_i)$ is based on the correspondence between Δ and $\mathbf{G}/\mathbf{G}_i$: $\delta_k \leftrightarrow \mathbf{G}_i g_k$ (k = 1, 2, ..., r).

For example, an allene derivative (**1**) of D_{2d} symmetry (○ = H) is considered to be generated from a parent molecule by substituting hydrogens for the eight terminal positions (Figure 1). The right-hand figure is a sideview of the allene molecule. Atoms in the molecule are divided into orbits (Δ_1 to Δ_4) governed by coset representations [$D_{2d}(/C_1)$, $D_{2d}(/C_2)$, $D_{2d}(/C_{2v})$, and $D_{2d}(/D_{2d})$].[5,6]

Let us next consider the desymmetrization of $\mathbf{G}$ into its subgroup $\mathbf{G}_j$. This process restricts the CR $\mathbf{G}(/\mathbf{G}_i)$ within the

0095-2338/91/1631-0540$02.50/0 © 1991 American Chemical Society