# Computer Generation of Molecular Structures by the SMOG Program

M. S. Molchanova, V. V. Shcherbukhin, and N. S. Zefirov*

N. D. Zelinsky Institute of Organic Chemistry, Leninsky Prosp., 47, Moscow, 117913, Russia

Received December 7, 1995[⊗]

The SMOG program for exhaustive and irredundant generation of chemical structures by the given molecular formula is described. The program makes use of the graph-theoretical Faradjev algorithm, which was essentially modified to be efficiently applied for chemical purposes. The major improvements of the algorithm include adequate consideration of the sets of required and forbidden structural fragments ("Goodlist" and "Badlist", respectively), effective use of invariant substructures ("core" fragments), possible consideration of various atomic valence states (atoms with formal charges, *etc*.), and rigorous treatment of aromatic structures. The output structures are visualized as 2D graphs according to an original algorithm. The advantages and possible applications of this software are discussed.

## 1. INTRODUCTION

Computer-assisted design of organic structures is a field of significant importance in mathematical chemistry. An increasing number of generators (computer programs for constructing structural formulas of organic compounds) are being developed for solving problems in chemistry, spectroscopy, QSAR studies, *etc*.[1−14]

The goal of any structural generator is the efficient and exhaustive generation of nonisomorphic structures that comply with the given restrictions. The most common constraints are (1) the conservation of the molecular formula and/or (2) the necessary presence of some structural fragments (the set of such fragments is usually called *Goodlist*) and/or the required absence of certain fragments (*Badlist*).[1,15] With respect to these considerations, structural generators may be formally divided into two overlapping classes.

Firstly, there are numerous programs that generate chemical structures with one and the same molecular (empirical) formula.[1−10] Atoms are regarded as having constant valences, such as C(IV), H(I), N(III), *etc*. From the graph-theoretical standpoint, a program of this kind generates the complete set of nonisomorphic graphs using a given set of vertices (atoms) with the given degrees (valences). This combinatorial problem has many rigorous algorithmic solutions. However, these algorithms are not always well adapted to the solution of actual chemical problems, where the presence and/or absence of certain fragments in target structures is also important. Obviously, straightforward generation of all structures, followed by elimination of those which do not satisfy the given constraints, inevitably results in a dramatic loss of efficiency.

The second class includes generators which assemble graphs from a set of definite structural fragments, whereas the molecular formulas of the resultant structures either should be constant or may vary.[11−14] However, some programs of this class are based on semiheuristic algorithms, which cannot guarantee rigorous solution of the problem. We have to emphasize that duplicate structures are produced by many known programs of both classes,[6,10c,11b,13] and, what is more, nonexhaustive generation is sometimes encountered.[11b]

The objective of our work was to develop an algorithm and a computer program **SMOG** (**S**tructural **MO**lecular **G**eneration)[16] that would combine the advantages of both approaches: conduct mathematically rigorous, exhaustive, irredundant, and efficient generation of chemical structures with the given molecular formula, at the same time considering additional restrictions (such as the presence or absence of fragments) at the earliest stages of the process.

The SMOG software is based on an effective and rigorous graph-generating algorithm, which was designed by Faradjev,[9,17] developed further by Molodtsov,[10] and significantly modified in our work to combine its high efficiency with chemical versatility.

## 2. ELEMENTS OF METHODOLOGY

**2.1. Basic Notions. 2.1.1. Molecular Graphs.** Within SMOG, target molecules are considered as *molecular graphs*—multigraphs without loops and with labeled vertices. Such a graph represents only the types and connectivities of atoms in a molecule; problems of spatial isomerism lie outside the scope of our problem. A molecular graph must be connected: catenanes or rotaxanes are regarded as consisting of separate molecules.

Vertices of a molecular graph may be distributed by *vertex types*: vertices belong to one and the same type if and only if the corresponding atoms have identical characteristics (the symbols of chemical elements, valences, charges, and possibly some additional features, such as isotopic labels). The multiplicities of the graph edges are equal to formal multiplicities of the corresponding chemical bonds (from 1 to 3). Special types of bonds (aromatic or semipolar) also may be considered by SMOG, but they are denoted by formal single, double, or triple bonds during generation—for example, a benzene ring is generated as a sequence of alternating single and double bonds. Evidently, the degree of each vertex (the summarized multiplicity of all adjacent edges) is equal to the valence of the corresponding atom.

If valences of all elements in the molecular formula are definite and constant, this formula unambiguously provides the distribution of vertices by types with definite degrees, which represents the complete set of required data for generation by the Faradjev algorithm.[9] For example, if the input formula is $C_6H_3Cl_5$ and the valences are C(IV), H(I), and Cl(I), generation will be based on six vertices of type 1 (degree 4), three of type 2 (degree 1), and five of type 3 (also degree 1).

[⊗] Abstract published in *Advance ACS Abstracts*, May 15, 1996.

S0095-2338(95)00393-3 CCC: $12.00

**2.1.2. Various Valence States. Superatoms.** If the valence of some element in the molecular formula may vary, then the generation process should be automatically repeated several times, so that all possible valence states of the element are considered successively. Let us consider the empirical formula $C_2H_2N_2O_2$ and state that C is tetravalent, H is univalent, but N may be trivalent in the neutral state and tetravalent is the formal positively charged state, O is divalent is the neutral state and univalent in the negatively charged state. Then, the following possible *sets of vertices* (or *sets of valence states*) are automatically found by SMOG and successively used for generation:[18]

$$2 \times \mathrm{C(IV)},\ 2 \times \mathrm{H(I)},\ 2 \times \mathrm{N(III)},\ 2 \times \mathrm{O(II)} \quad \text{(four types of vertices)} \quad (1)$$

$$2 \times \mathrm{C(IV)},\ 2 \times \mathrm{H(I)},\ 1 \times \mathrm{N(III)},\ 1 \times \mathrm{N^{+}(IV)},\ 1 \times \mathrm{O(II)},\ 1 \times \mathrm{O^{-}(I)} \quad \text{(six types)} \quad (2)$$

$$2 \times \mathrm{C(IV)},\ 2 \times \mathrm{H(I)},\ 2 \times \mathrm{N^{+}(IV)},\ 2 \times \mathrm{O^{-}(I)} \quad \text{(four types)} \quad (3)$$

Examples of graphs for these three sets are given in Figure 1. Apparently, molecular structures that are produced for one and the same set of valence states may be referred to as *valence isomers*, according to the accepted definition.[2b]

We may similarly consider any elements with variable valences: P(III) and P(V); S(II) and S(IV); N(III), $N^+$(IV), and $N^-$(II), *etc*. However, it often happens that atoms in some "unusual" valence states are encountered in real molecules only in a definite environment—for example, Cl(VII) is usually included in the $-O-ClO_3$ group, $N^+$ often must have a negatively charged atom ($O^-$, $N^-$ or $C^-$) as one of its neighbors, and so on. If this constraint is not taken into account, an overwhelming majority of chemically undesirable solutions usually appear. For example, the number of nitropropane ($C_3H_7NO_2$) isomers for the set of valence states with one $N^+$(IV) and one $O^-$(I) atom is 2, if we specify that $N^+$(IV) should be included in the nitro group $-NO_2$, and 367 without this constraint.

Therefore, SMOG provides a possibility to consider atoms in unusual valence states **only if** they are accompanied by some definite user-specified environment(s) in the target molecule. Such permissible environments are called *superatoms* within SMOG. For example, the superatoms for $N^+$ may be $N^+{-}O^-$ (see Figure 1), $N^+{\equiv}C^-$ (isonitrile), $N^-{=}N^+{=}N-$ (azide), and so on. Of course, a chemist who uses SMOG is provided with the possibility to extend and edit the set of superatoms or to remove this constraint altogether.

**2.1.3. Fragment Graphs.** Exactly as the target molecules, fragments from Goodlist and Badlist are represented as graphs by SMOG (we will refer to them as *fragment graphs*). Fragment graphs should be "compatible" with the given molecular formula and have free valences. They may be connected or disconnected. The presence of a fragment in a target molecule is determined as isomorphic inclusion of the fragment graph in it, *i.e.*, as the presence of the corresponding subgraph in the molecular graph.

**2.1.4. Core Fragments.** In the solution of many actual problems, some fragments of the desired molecular graphs are predefined before the start of generation and may be used as the basis for the construction process, alongside with separate atoms. In SMOG, such invariant substructures are called *core fragments* and form a specific part of Goodlist. Their efficient use is an extremely helpful feature for the solution of numerous chemical problems. For example, if a chemist is trying to find the structure of a product of some chemical reaction, he usually knows its empirical formula and some structural fragments (from the molecules of the starting reagents) that were incorporated in the product. Another general problem of a similar kind is the generation of isomers that belong to a definite chemical class.

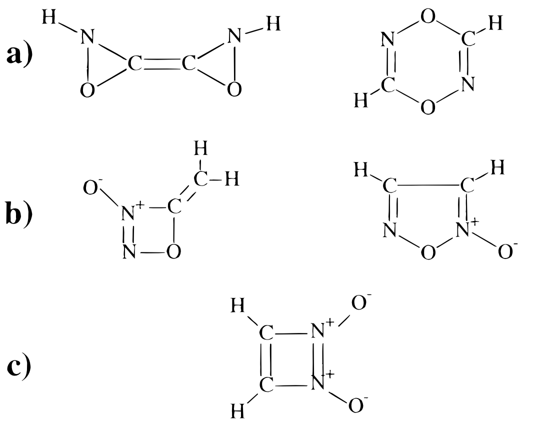
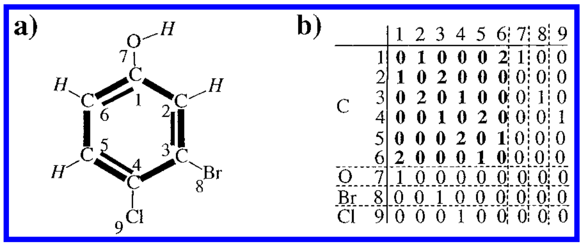


**Figure 1.** Examples of structures with the formula $C_2H_2N_2O_2$, corresponding to different sets of valence states. For each of the first two sets, two valence isomeric structures are presented.



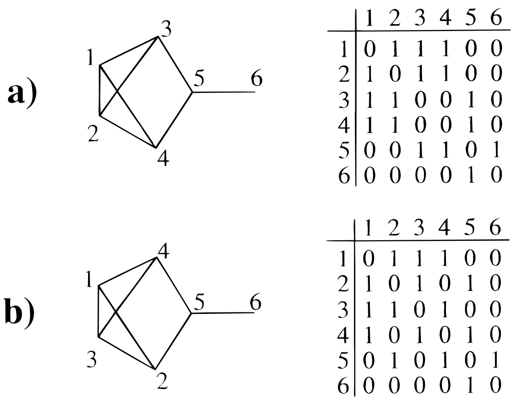
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| C | 1 | **0** | **1** | **0** | **0** | **0** | **2** | 1 | 0 | 0 |
| | 2 | **1** | **0** | **2** | **0** | **0** | **0** | 0 | 0 | 0 |
| | 3 | **0** | **2** | **0** | **1** | **0** | **0** | 0 | 1 | 0 |
| | 4 | **0** | **0** | **1** | **0** | **2** | **0** | 0 | 0 | 1 |
| | 5 | **0** | **0** | **0** | **2** | **0** | **1** | 0 | 0 | 0 |
| | 6 | **2** | **0** | **0** | **0** | **1** | **0** | 0 | 0 | 0 |
| O | 7 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Br | 8 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Cl | 9 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

**Figure 2.** An example of generation with the use of a core fragment. The invariant part (benzene ring) is marked (a) by thick lines and (b) by boldface.

Let us consider the enumeration of all bromochlorophenols ($C_6H_4BrClO$). The straightforward but totally inefficient solution would be to generate all $C_6H_4BrClO$ isomers and then exclude those which do not contain the benzene fragment. A much more suitable approach, however, is to fix the benzene ring as an invariant (core) substructure and then consider only the connectivity of the remaining atoms, as is described in the following sections. Figure 2a shows one of the structures that are built on the basis of the benzene ring.[19]

**2.2. Consideration of the Irredundancy Problem. 2.2.1. The Canonicity Criterion.** The most important problem in structural generation, as in any combinatorial enumeration process, is to prevent the appearance of duplicate resultant structures. That is, a generator should produce strictly one graph out of each family of isomorphic graphs.

There are two possible ways to exclude duplicates in the course of stepwise generation.[17a] The first method (breadth-first search) is to generate all possible intermediate structures at each step, on the basis of all intermediate structures generated at the previous step. Then duplicate intermediates are found and excluded, and afterwards one may pass to the next step and repeat the procedure. However, this approach[20]



$$\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 1 & 0 & 0 \\ 3 & 1 & 1 & 0 & 0 & 1 & 0 \\ 4 & 1 & 1 & 0 & 0 & 1 & 0 \\ 5 & 0 & 0 & 1 & 1 & 0 & 1 \\ 6 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}$$

$$\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 & 1 & 0 \\ 3 & 1 & 1 & 0 & 1 & 0 & 0 \\ 4 & 1 & 0 & 1 & 0 & 1 & 0 \\ 5 & 0 & 1 & 0 & 1 & 0 & 1 \\ 6 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}$$

**Figure 3.** Examples of (a) canonical and (b) noncanonical adjacency matrices.

requires the storage and handling of many intermediate structures and becomes inefficient as their number increases.

The second approach, implemented in algorithms of constructive search (including the Faradjev algorithm), uses the principles of the branch-and-bound method.[21] Namely, one unique distinguishing rule, called the *canonicity criterion*, is formulated for the target graphs: exactly one graph out of each family of isomorphic graphs satisfies this criterion and is defined as *canonical*. Then, this criterion is *extended* to partially generated structures at intermediate stages, thus making it possible to find and eliminate (at the earliest possible stages) those intermediates which *a priori* cannot produce canonical graphs.

Within SMOG, molecular graphs are represented in the form of adjacency matrices. As is known, the family $\{\mathbf{G}_i\}$ of isomorphic $N$-vertex graphs is described by $N!$ adjacency matrices $\mathbf{A}_i$, which are interconvertible by a permutation of lines and the corresponding permutation of columns: graphs $\mathbf{G}_i$ and $\mathbf{G}_j$ are isomorphic if and only if the symmetrical group $S_N$ contains such a permutation $g$ that $\mathbf{A}_j = g\mathbf{A}_i g^{-1}$. A canonical graph, unambiguously selected from each isomorphic family, corresponds to a *canonical adjacency matrix*.

In his algorithm,[9] Faradjev regards the "maximum matrix" as canonical: the $N^2$-dimensional numerical vector $(a_{11}a_{12}...a_{1N}a_{21}...a_{2N}...a_{NN})$ of the canonical matrix $\mathbf{A}(N,N) = (a_{ij})$ should **be lexicographically maximum throughout the family of interconvertible matrices: A** $\geq g\mathbf{A}g^{-1}$ for all $g \in S_N$. (A similar definition was later independently introduced for molecular graphs by Hendrickson[22] as the development of Randic's idea about the coding of graphs by the "minimum matrix".[23])

Apparently, the matrix in Figure 3b is not canonical by this definition, since it is smaller than the one in Figure 3a. Moreover, consideration of all permutations from $S_N$ directly proves that the matrix in Figure 3a is canonical.[24]

The chemical version of the Faradjev algorithm, designed by Molodtsov,[10] is based on a modified definition of the canonicity criterion, because vertices are distinguishable by their chemical labels. All $N$ vertices of the graph are classified by atom types, as described above; let us assume that the number of types is $T$. Then, let us order lines/columns of the adjacency matrix so that the first $n_1$ lines correspond to vertices of type $k_1$; ...; and the last $n_T$ lines correspond to type $k_T$. Let us consider only those permutations which preserve this distribution of lines/columns by types (matrices that are produced by such permutations are called *admissible*).[10b] If we denote the symmetrical group of permutations among vertices of type $k_t$ as $S(k_t)$, the permutations producing admissible matrices (these permutations are also called admissible) belong to the group $S(T) = \oplus_t S(k_t)$; $S(T) \subset S_N$. An admissible matrix is regarded as canonical if **no permutation** $g \in S(T)$ **produces a lexicographically greater admissible matrix**.

In the absence of core fragments, the same concept is used in SMOG. In the opposite case, it is modified further.[25] If a core fragment is specified, some elements of the adjacency matrix, which correspond to bonds within this core fragment, form an *invariant part* of the target matrix. That is, they are regarded as *fixed* from the start of generation. For example, see the matrix in Figure 2b: its elements that are displayed in boldface correspond to the adjacency matrix of the core fragment and therefore must remain constant during generation. Hence, we should consider only those permutations $g \in S(T)$ which do not change any elements in the invariant parts of the matrix: $a_{g(i)g(j)} = a_{ij}$. Then, **a canonical matrix is the one which cannot be increased by any permutation** $g \in S(T)$ **preserving invariant parts**. These permutations also form a group: $S(T,C) \subset S(T)$.

Note that the matrix in Figure 2b is not canonical by the two earlier definitions but canonical by the last definition.

**2.2.2. Extension of Canonicity Criterion to Partially Filled Matrices.** At the start of generation, elements of the adjacency matrix (except for invariant parts that correspond to core fragments) are regarded as unknown (*unfilled*). Each generation step consists in the *filling* of a new line, that is, in the assignment of definite values to its yet-undefined elements. Lines are filled in the order of their identity numbers, starting from the top. The corresponding columns are then filled automatically: $a_{ji} = a_{ij} (j > i)$.

The generation is conducted on the basis of the backtracking principle. When the process moves one step back, the line that was filled at the last stage is regarded is unfilled again. A step back is performed in one of the following cases:

(1) all lines have been filled, and the next target structure appeared at the output;

(2) for some intermediate line, all ways of filling the lower-lying lines have been considered;

(3) the yet-unfilled elements of the current partially filled matrix *a priori* cannot be filled so that the resultant adjacency matrix would correspond to any graph satisfying the given structural constraints (the connectivity requirement, Goodlist/Badlist, *etc*.);

(4) the yet-unfilled elements of the partially filled matrix *a priori* cannot be filled so that a canonical matrix would result.

Let us focus our attention on the fourth case. According to the above considerations, the determination of partially filled matrices that cannot produce a canonical matrix requires *the extension of the canonicity criterion to partially filled matrices*.

Let us consider the sets $I$ and $J$ of the lines that are filled and unfilled at the current generation stage, respectively. If all lines of the partially filled matrix $\mathbf{A} = \mathbf{A}(N,N)$ from 1 to $M - 1$ belong to the $I$ set and line $M$ belongs to the $J$ set, the rectangular matrix $\mathbf{A}(M - 1,N)$ is called the *upper part* of $\mathbf{A}$.

Now let us consider the sets of filled and unfilled lines for each vertex type $k_t$: $I_t$ and $J_t$. The group $S(I,J) = \oplus_t(S(I_t) + S(J_t))$ includes all admissible independent permutations among filled and unfilled lines/columns. If some parts of the matrix are fixed because of the presence of core fragments, we must consider $S(I,J,C)$ as the subgroup of $S(I,J)$ that is formed by those permutations which preserve invariant parts.

Now we can introduce a definition: a partially filled matrix is called *strongly canonical* if no permutation $g \in S(I,J)$—or, in the presence of core fragments, $g \in S(I,J,C)$—increases this matrix.

It may be easily proved[17b] that the requirement of strong canonicity is indeed an extension of the canonicity criterion: if a partially filled matrix (or, in particular, its upper part) is not strongly canonical, it cannot produce a canonical adjacency matrix.

**2.2.3. The Problem of Redundancy for Aromatic Structures.** Although SMOG does not explicitly consider aromatic bonds during generation, it can recognize aromatic systems in the generated molecules: five-membered aromatic rings with heteroatoms (furan, pyrrole, thiophene); six-membered rings with alternating single and double bonds like benzene and pyridine; the cyclopentadienyl anion and the tropylium cation; and various polycyclic systems that obey the “4$n$+2” rule.[26]

The majority of known generators cannot correctly treat the structures that correspond to different resonance forms of one and the same aromatic compound. That is, some isomers in the output of such generators may seem different from the formal standpoint but are actually equivalent for a chemist. The classical examples of such redundancy are *o*- and *m*-disubstituted benzenes:[12b] *e.g.*, if we regard the benzene ring as a system of single and double bonds, we get two “isomers” for *o*-dichlorobenzene (with a single or double bond between the C atoms with Cl substituents) instead of one. The same difficulty arises for numerous polysubstituted or polycyclic aromatic compounds.

Within SMOG, this problem is solved by finding a unique *canonical distribution of double bonds in the aromatic system*. The method of finding this distribution is described in section 4.4.

**2.3. Treatment of Univalent Vertices.** Let us consider a set of vertices: $n_1$ vertices of type $k_1$, ... $n_T$ of type $k_T$. Then, each of the generated graphs must contain $N_0 = \sum_{t=1}^{T} n_t$ vertices. However, the worktime of the generation algorithm depends exponentially on the size of target graphs. Therefore, if the set of vertex types contains a large number $n_H$ of univalent vertices of one of the same type $k_H$, it is appropriate not to generate complete molecules but to construct “quasimolecules”—pseudographs without vertices of the type $k_H$ but with $n_H$ free valences, which may be automatically saturated by the remaining $n_H$ univalent atoms. Then, the matrix size is reduced by $n_H$ lines/columns: $N = N_0 - n_H$.

For example, let us return to Figure 2: although the total number of atoms in the molecular formula $C_6H_4BrClO$ is 13, the size of the constructed pseudograph and of its adjacency matrix is 9. The hydrogens (their symbols are italicized in Figure 2) saturate the valences that remain free at each atom after all other bonds of this atom have been constructed during generation.

Hereafter, univalent atoms that are treated in this way are referred to as quasihydrogens. Indeed, in the majority of actual problems they are H atoms, as in the above example.

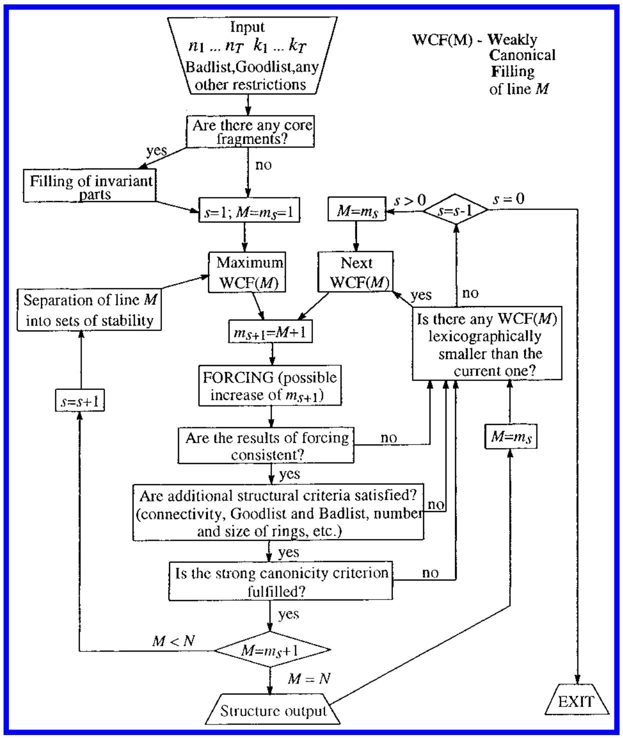

Input
$n_1 \ldots n_T$ $k_1 \ldots k_T$
Badlist,Goodlist,any other restrictions
WCF(M) - Weakly Canonical Filling of line $M$
Are there any core fragments?
yes
no
Filling of invariant parts
$s$=1; $M$=$m_s$=1
$M$=$m_s$
$s > 0$
$s$=$s$-1
$s = 0$
Maximum WCF($M$)
Next WCF($M$)
yes
no
Separation of line $M$ into sets of stability
$m_{s+1}$=$M$+1
Is there any WCF($M$) lexicographically smaller than the current one?
FORCING (possible increase of $m_{s+1}$)
$s$=$s$+1
Are the results of forcing consistent?
no
$M$=$m_s$
yes
Are additional structural criteria satisfied? (connectivity, Goodlist and Badlist, number and size of rings, etc.)
no
yes
Is the strong canonicity criterion fulfilled?
no
yes
$M < N$
$M$=$m_s$+1
$M = N$
Structure output
EXIT

**Figure 4.** Simplified flow chart of SMOG. The notation is explained in the text.

However, if the molecular formula is $C_6HBrCl_4O$ rather than $C_6H_4BrClO$, SMOG will treat Cl atoms as quasihydrogens, because their number is greater than the number of hydrogens. Then, the H atom is explicitly considered during generation, and Cl saturates the free valences.

According to our experience, the use of this reduction technique makes the CPU time of generation smaller by a factor of 5−30 for the majority of actual chemical problems.[27]

SMOG is based on a concept of *variable* vertex degrees. Indeed, the degree of a vertex in a quasistructure (the sum of the formal multiplicities of all its adjacent bonds) may vary between the maximum value $D_i$ (equal to the atom valence) and minimum value $D_i^0$

$$D_i^0 = D_i - \min(D_i - 1, H_i) \qquad (4)$$

where $H_i$ is the number of free valences that have not yet been attached to any vertex at the stage when the $i$th line of the matrix is being filled.[28]

At the start of generation, $H_1 = n_H$. After the filling of each line $j$, the $H_i$ value for any still-unfilled line $i$ is reduced by $D_j - D_j^*$, where $D_j^* = \sum_{k=1}^{N} a_{jk}$ is the actual degree of the vertex $j$. The condition $\sum_{j=1}^{N} (D_j - D_j^*) = n_H$ should be satisfied at the final stage.

## 3. GENERATION ALGORITHM

The main procedures and stages of the generation process are outlined below and schematically shown in Figure 4, where $s$ is the number of the current generation step, $m_s$ is the number of the line that is filled at step $s$, $M$ is the line directly below the upper part $\mathbf{A}(M-1,N)$ of the matrix (the line to be filled at the current step). The remaining notation is explained above. The definition of a “weakly canonical filling” (WCF) is given in section 3.3.

For simplicity, in the next subsections we consider the case when core fragments are absent (unless specially stated otherwise).

**3.1. Treatment of Initial Data.** Again, let us assume that the first $n_1$ lines of the target adjacency matrix correspond to the vertex type $k_1$, ..., the last $n_T$ lines, to the type $k_T$. In addition, there are also $n_H$ quasihydrogens of the type $k_H$. The size of the target adjacency matrices is

$$N = \sum_{\substack{t=1 \\ t \neq H}}^{T} n_t$$

At the start of generation, elements of the adjacency matrix $\mathbf{A}(N,N)$ are unknown, except for diagonal ones: $a_{ii} = 0$. For any pair of vertices $i,j \leq N(i \neq j)$, we may determine the maximum possible multiplicity of an edge as $E_{ij} = \min(\min(D_i,D_j),3)$, because the bond multiplicity in organic molecules does not exceed three. However, if $D_i = D_j$ and $N > 2$, then $E_{ij} = \min(D_i - 1,3)$, as directly follows from the requirement of connectivity.

**3.2. Filling of the First Line.** To fill the first line of the target matrix, we should find such a set of $a_{1i}$ values ($0 \leq a_{1i} \leq E_{1i}$) that the upper part $\mathbf{A}(1,N)$ would be strongly canonical and $D_1^0 \leq D_1^* = \sum_{i=2}^{N} a_{1i} \leq D_1$, where $D_1$ is the atomic valence and $D_1^0$ is calculated by eq 4.

One can easily prove[10,17] that the upper part $\mathbf{A}(1,N)$ may be strongly canonical only if the condition

$$(j > i) \rightarrow (a_{1i} \geq a_{1j}) \qquad (5a)$$

is true for any pair of vertices $i,j > 1$ that belong to one and the same vertex type $n_t$.

If the adjacency matrix contains invariant parts, this rule has to be modified: if lines $i$ and $j$ belong to one and the same type **and** their permutations do not change the values of the fixed elements, then condition (5) necessarily must be satisfied.

That is, we may say that all vertices are divided into subsets in such a way that condition (5) should be true within each subset—otherwise, the upper part $\mathbf{A}(1,N)$ is not strongly canonical. Hereafter, such subsets are called *initial sets of stability*.

The difference $h_1^* = D_1 - D_1^*$ corresponds to free valences that are "attached" to the first atom; therefore, the remaining number of free valences at the second stage will be $H_2 = H_1 - h_1^*$. A similar rule refers to all subsequent lines: after the filling of each line, we get $H_{i+1} = H_i - h_i^*$.

**3.3. Separation of Next Lines into Sets of Stability.** Let us assume that the first line was filled according to the above-described rules. Since $a_{21} = a_{12}$ and $a_{22} = 0$, the interval $[a_{23},a_{2N}]$ should be filled so that $D_2^0 \leq D_2^* = (a_{12} + \sum_{i=3}^{N} a_{2i}) \leq D_2$, and no permutation of lines/columns within the initial sets of stability would increase $\mathbf{A}(2,N)$. As is easily shown,[9] the condition of strong canonicity $(j > i) \rightarrow (a_{2i} \geq a_{2j})$ should necessarily refer only to those $(i,j)$ pairs within these sets for which $a_{1i} = a_{1j}$.

Let us introduce some new definitions. If all lines up to $M - 1$ are filled at the current stage, then two vertices $i,j \leq M + 1$ belong to one and the same set of stability in line $M$ if and only if $i$ and $j$ lie within one and the same initial set of stability **and** $a_{mi} = a_{mj}$ for all $m < M$. If line $M$ is filled so that the general rule

$$(j > i) \rightarrow (a_{Mi} \geq a_{Mj}) \qquad (5b)$$

is true within each set of stability in line $M$ ($i,j \leq M + 1$), such filling is referred to as *weakly canonical*.

Now we can generalize the above statements. **A partially filled matrix may be strongly canonical only if each successive line was filled in the weakly canonical manner**. The proof of this statement is evident:[9,10] if the rule of weak canonicity is violated for some pair of vertices ($i,j$), the upper part of the matrix may be increased by the permutation of these vertices.[29]

**3.4. Test for Strong Canonicity.** Weakly canonical filling of all lines is a necessary but insufficient condition for complete exclusion of noncanonical matrices. For example, let us consider the matrix in Figure 3b again, assuming that vertices 1−5 belong to one and the same vertex type and no core fragments are present. Apparently, all lines are filled in the weakly canonical way, but the permutation (1 3 2 4 5 6) increases this matrix.

In this connection, a special procedure for checking the strong canonicity of partially filled matrices is required after each step of generation. For example, the matrix in Figure 3b should not be generated at all, because the partially filled matrix with the upper part $\mathbf{A}(3,6)$ should be discarded (as not strongly canonical) after the filling of the third line: the aforementioned permutation is contained in the $S(I,J)$ group at this stage.

Apparently, the test for strong canonicity should be based on the search for such a permutation from $S(I,J)$—or $S(I,J,C)$ in the presence of invariant fragments—that increases the partially filled matrix. If such a permutation exists, the process moves one step back (Figure 4). Since exhaustive search within the $S(I,J)$ or $S(I,J,C)$ group is the most complex and time-consuming procedure in the algorithm, several ways to optimize it have been developed.[30] We will not consider them in this paper.

**3.5. Forcing.** *Forcing* is the procedure that "complements" the weakly canonical filling of the current line by subsequent automatic filling of some other yet-unfilled lines **at the same step**. Thus, it is a powerful tool for increasing the efficiency of generation.

For example, let us assume that the $M$th line of the matrix was filled at the current stage, and all lines with greater numbers are unfilled as yet. Let us consider some unfilled line $\mu > M$ and denote the sum of its already filled $M$ elements as $\Sigma_\mu = \sum_{i=1}^{M} a_{i\mu}$. The number of free valences that have not yet been "attached" to the $M$ filled lines is $H_{M+1} = n_H - \sum_{j=1}^{M} (D_j - D_j^*)$, then, $D_\mu^0 = D_\mu - \min(D_\mu - \Sigma_\mu, H_{M+1})$.

Also, let us denote the sum of the maximum possible values for all unfilled elements of line $\mu$ as

$$\Gamma_\mu = \sum_{\substack{j=M+1 \\ j \neq \mu}}^{N} E_{j\mu}$$

Automatic filling of line $\mu$ is possible in each of these cases:

(a) $D^0_\mu - \Sigma_\mu = \Gamma_\mu + H_{M+1}$. Then, all elements $a_{j\mu}$ ($j > M, j \neq \mu$) may be automatically assigned the $E_{j\mu}$ values, and $h^*_\mu = H_{M+1}$.

(b) $D_\mu - \Sigma_\mu = 0$. All $a_{j\mu}$ elements should be filled with zeros.

(c) $H_{M+1} = 0$ and only one $a_{j\mu}$ element in line $\mu$ remains unfilled as yet. Then, $a_{j\mu} = D_\mu - \Sigma_\mu$.

Also, a special type of forcing (we call it *quasiforcing*) is possible when $\sum_{j=M+1}^{N} (D_j - 1) = H_{M+1}$ or when $H_{M+1} = 0$. As a result of quasiforcing, all $h^*_j$variables for lines $M + 1 \leq j \leq N$ are assigned definite values: $h^*_j = D_j - 1$ or $h^*_j = 0$, respectively. After quasiforcing, the actual sums of elements in the lines below $M$ must be constant at subsequent generation steps: $D^*_j = D_j - h^*_j = const$.

If an attempt to conduct forcing produces inconsistent results (for example, $\Sigma_\mu > D_\mu$), the program returns to the previous step.

## 4. SELECTION RULES

**4.1. Connectivity Test.** As was emphasized above, structural generation is explicitly aimed at the search for connected molecular structures. Therefore, SMOG includes a simple test for connectivity, which is applied after each step of the generation process to the partially filled adjacency matrix. If this matrix represents a disconnected partially constructed graph that cannot be made connected by any subsequent filling of the matrix, a step back is made (Figure 4).

**4.2. Goodlist, Badlist and Other Constraints. 4.2.1. General Consideration of Goodlist and Badlist.** SMOG includes a dialog module for the input of various user-specified structural constraints. The most important among them are Goodlist and Badlist, which are used for consideration of some known (from spectroscopic and/or chemical studies) or assumed data about the target molecules: the presence or absence of definite functional groups or cyclic systems; the data about hybridization states or environments of atoms, *etc*. Badlist is also often used for the elimination of chemically infeasible and/or strained structures: small- and medium-size rings with triple or cumulated double bonds, unwanted tautomers derived from OH or NH groups adjacent to $\mathrm{C{=}C}$ or $\mathrm{C{\equiv}C}$ bonds, and so on.

SMOG is capable of considering diverse forms of Goodlist and Badlist. For example, Goodlist may be constructed on the basis of different logical operations: **AND** (each of the target molecules must contain **all** fragments in the list), **OR** (molecules need to contain **at least one** fragment from the list),[31] **XOR** (molecules must contain **strictly one** fragment out of the list), or any combinations of these operations. Similar constructions may be used with respect to Badlist as well.

For each fragment from Goodlist/Badlist, the user also may specify the minimum and maximum permissible/forbidden number of its inclusions in the target structure. In this case, all different isomorphic inclusions of the fragment graph in the molecular graph will be found (they may overlap but not coincide), and then the number of inclusions is compared with the user-input threshold(s).

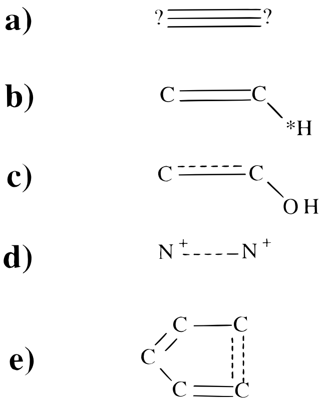


**Figure 5.** Examples of generalized fragments that may be used for the formation of Goodlist/Badlist: (a) "?" stands for any non-hydrogen atom; (b) "*" stands for any heteroatom (N, O, S, etc.); (c) a double solid-and-dashed line denotes an aromatic bond; (d) a single dashed line denotes a bond of arbitrary multiplicity; and (e) a double dashed line denotes a connected path of any length.

For convenience, SMOG also allows the user to represent structural fragments (that may be included in Goodlist/ Badlist) in the "generalized" form, as is explained in the legend to Figure 5. For example, if Badlist includes the fragment in Figure 5a, it means that the target molecules should contain no triple bonds regardless of the atoms that form them; inclusion of the fragment in Figure 5d into Badlist means that all bonds between $N^+$ atoms are forbidden (regardless of the multiplicity), *etc*.

An especially interesting feature is the possibility to consider the inclusion of "generalized" substructures that contain chains of arbitrary length, as is shown in Figure 5e. The double dashed line between atoms may mean either a bond (or arbitrary multiplicity) or a connected chain that is formed by any number of atoms (apart from those that are explicitly denoted in the fragment). As will be shown in example 5 of this paper, the possibility to include such fragments in Goodlist and/or Badlist may be extremely useful in some cases. For example, it is helpful in the generation of molecules that contain or do not contain a definite system of rings, in the case when the lengths of some rings are unknown but the structure of their conjugation is important.

**4.2.2. Allowed Substructures in Badlist.** A useful modification of Badlist in SMOG is the concept of *allowed fragments*. Let us explain it by an example.

The enol fragment $\mathrm{C{=}C{-}OH}$ generally denotes an unstable group and therefore is often added to Badlist. However, in some cases such a group may be chemically feasible. For example, the $\mathrm{C{=}C{-}OH}$ fragment is stable if the formal double $\mathrm{C{=}C}$ bond in it actually represents a bond from an aromatic ring, or if there is another double bond in the $\beta$-position ($\mathrm{O{=}C{-}C{=}C{-}OH}$, as in the enol form of acetylacetone).

So, to avoid the exclusion of these stable structures, we may add the enol fragment to Badlist, at the same time adding the $\mathrm{O{=}C{-}C{=}C{-}OH}$ fragment and the generalized fragment that is shown in Figure 5c (with an aromatic $\mathrm{C{-}C}$ bond) to the list of allowed fragments.[32] In this case, molecular graphs containing the enol fragment as a subgraph will be discarded **only if** the enol in them is not a part of another subgraph corresponding to one of the allowed fragments.

Superatoms, which are described in section 2.1 of this paper, are also examples of implicitly used allowed fragments. Indeed, the program considers atoms in unusual valence states only if they are surrounded by user-specified allowed environments.

**4.2.3. Microfragments.** *Microfragments* are structural fragments that contain only one non-hydrogen atom (possibly with a definite number of hydrogens attached to it) with explicitly defined distribution of multiple bonds or hybridization state. For example, $CH_3-$, $=C=$, and C⟨$sp^2$⟩ are some of the possible microfragments for carbon atoms. Microfragments form a special part of Goodlist and Badlist: for each type of the microfragment, the user may specify its minimum and/or maximum content ($W_{min}/W_{max}$) in target structures.

**4.2.4. Constraints Concerning the System of Rings.** The use may set restrictions on the structure of the cyclic system in target molecules. This set of constraints may include

(a) the minimum/maximum number of independent rings;

(b) the minimum/maximum length of a simple ring ($\geq 3$);

(c) the possible presence/absence of rings with triple bonds or cumulated double bonds (as a rule, small- and medium-size rings containing such bonds are extremely strained and should be avoided).

**4.2.5. The Use of Structural Restrictions at Intermediate Stages.** The straightforward way to use Goodlist and Badlist is to generate complete molecules and then to check them for the presence of fragments. However, a much more efficient approach is to perform this test as early during generation as possible, so that some branches of the generation tree could be eliminated.

Before generation, SMOG determines the minimum and maximum "test steps" $L_{min}$ and $L_{max}$ for each fragment: the presence of a fragment in the partially constructed graph that corresponds to a partially filled matrix is searched only when the lowest line $M$ of the upper part $\mathbf{A}(M,N)$ lies within these boundaries: $L_{min} \leq M \leq L_{max}$.

For example, let us consider generation by the formula $C_6H_4N_2O$ ($n_1 = 6$, $n_2 = 2$, $n_3 = 1$; $n_H = 4$), assuming that Goodlist contains the two-atom fragment C=C. For this fragment, $L_{min} = 2$ and $L_{max} = 6$. Indeed, the submatrix

$$\mathbf{A}^{\#}(2,2) = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

of the adjacency matrix $\mathbf{A}$ is the smallest subset of $\mathbf{A}$ that may correspond to a subgraph containing the C=C fragment. At the same time, the submatrix $\mathbf{A}^{\#}$ (6,6) is the last subset that should be checked for the presence of C=C, because the filling of lines that correspond to N and O atoms will not affect the presence of the C=C substructure.

Isomorphic inclusion of fragment graphs in intermediate structures is determined by a back-tracking algorithm,[33] modified in SMOG for treatment of partially filled adjacency matrices.

**4.3. Symmetry Test.** In some particular cases, generation is specially aimed at the search for symmetrical structures. SMOG determines the topological symmetry of the generated structures as the order of its automorphism group, using the algorithm described in ref 33. Structures with insufficient symmetry (the user-specified limit exceeds the order of their automorphism group) are discarded.

The symmetry test is conducted only at the last stage of generation, for completely filled adjacency matrices. This is natural, because the addition of each new bond to a partially filled matrix may change the symmetry of a complete structure.

**4.4. Handling of Aromatic Systems.** If SMOG finds that the generated molecule contains an aromatic ring system (see section 2.2.3), a special additional test for the canonical form of the aromatic system is conducted (by the user's desire) in the following way:

(a) All aromatic bonds are found.

(b) A "temporary" adjacency matrix of the generated structure is constructed. This temporary matrix is similar to the actual adjacency matrix, but the multiplicity of aromatic bonds is assumed to be 1.5 instead of 1 or 2 in the initial matrix.

(c) Vertices are assigned unique numbers by the procedure of canonical indexing[17b,34] applied to the temporary matrix.

(d) All permissible "embeddings" of a conjugated system of double bonds in the aromatic system are found. These embeddings are lexicographically ordered according to the numbers that are unambiguously assigned to vertices at step (c).

(e) If the actual placement of double bonds in the generated adjacency matrix coincides with the lexicographically maximum permissible embedding of double bonds found at step (d), *the distribution of double bonds in the aromatic system is regarded as canonical*. Otherwise, the current aromatic molecule is discarded.

This method works both for polysubstituted aromatic molecules and for polycyclic hydrocarbons (anthracene, phenanthrene, *etc*.). However, the search for aromaticity within SMOG is based on semiheuristic rules. Some aromatic structures are not recognized (for example, the cyclopropylium cation or annulenes with large rings). Moreover, the "$4n+2$" Hückel rule used in SMOG is one of the accepted concepts but surely not the only one possible: for example, CHEMICS[11] defines a system as aromatic if the overall number of $\pi$-electrons in it is $4n+2$ or $4n$ (but not 4 or 8), and some additional rules of ring conjugation are satisfied. Naturally, the formalism used in SMOG may appear to be inapplicable in some particular cases.

**4.5. Detailed Consideration of Core Fragments.** As was mentioned above, core fragments are included into the target structures as invariant substructures: that is, adjacency matrices of core fragments are directly "embedded" into the unfilled adjacency matrix of target molecules before the start of generation, and the corresponding parts of the adjacency matrix remain constant during generation. The resultant molecules will automatically contain the necessary fragments, and the generation itself is essentially accelerated, since the fixed elements in the target matrices need not be found by additional search.

If a core fragment has only one atom with free valences, like $-CO-NH_2$ or $\rangle C-CCl_2-CH_2-CH=O$, and the exact number of such invariant substructures in the target molecules is known, the generation procedure may be accelerated even more efficiently: the whole core fragment is considered during subsequent generation as one vertex of some special type called *group* within SMOG. That is, all "peripheral" (without free valences) atoms of the core fragment do not participate in generation. Evidently, such an approach reduces the size of the adjacency matrix. Before visualization or tests for the presence of other fragments, groups are automatically *expanded*—*i.e.*, the corresponding core fragments are substituted at their place.

By default, core fragments do not overlap. However, the user may explicitly specify the possibility of overlapping

**Table 1.** Comparison of the Numbers of Isomers Produced by Different Generators[a]

| formula | SMOG | MOLGRAPH | CHEMICS | CAMGEC | GI |
|---|---|---|---|---|---|
| $C_6H_{10}O$ | 747 | 747 | 745 | 748 | 747 |
| $C_5H_8BrCl$ | 140 | 140 | 108 | | |
| $C_3H_7NO$ | 84 | 84 | 87 | | |
| $C_4H_7NO$ | 764 | 764 | 802 | 767 | 764 |
| $C_6H_9NO$ | 35759 | 35759 | 37491 | | |

[a] Atomic valences are regarded as constant.

within the set of core fragments. In this case, their nonoverlapping parts are automatically determined (by an original procedure aimed at the search for isomorphic intersections[33]) before the start of generation, and only these parts are directly included as fixed in the adjacency matrix, whereas the isomorphic inclusion of the complete fragments has to be additionally checked for.

## 5. OUTPUT AND TREATMENT OF RESULTS

Structures generated by SMOG may be saved to a file and/or shown at the computer screen in the form of 2D graphs. Template-free satisfactory visualization of molecular graphs, especially complex ones, is a problem that has no perfect solution for the general case. However, there are some well-known approaches to the generation of acceptable 2D Cartesian coordinates on the basis of known molecular connectivity; we can mention the methods elaborated by Carhart[35] and Shelley.[36] Our visualization program **VICOM** (**VI**sualization of **CO**mplex **M**olecules), which forms a part of SMOG, combines some features of these algorithms with our ideas and corrections. According to our experience, VICOM yields a satisfactory 2D representation of target molecules in the overwhelming majority of cases. The algorithm will be described in a forthcoming paper.

If the generated structures are stored to a file, the user may perform their subsequent additional screening with the help of some new Goodlist and/or Badlist. This procedure is helpful in the case when some new structural data become available only after the generation is completed (see example 1 of this paper).

## 6. RESULTS AND DISCUSSION

Comparing the results produced by SMOG with those of some known structure-generating systems, such as DENDRAL,[1] MOLGRAPH,[2] IGOR,[3] AEGIS,[7] or GEN,[12] we see that these programs and SMOG yield the same numbers of isomers for any definite formula (without additional constraints). The comparison of SMOG with the CHEMICS program[11] showed that their results are different (Table 1); however, CHEMICS is known to produce some duplicates, on the one hand, and to disregard certain structures, on the other hand.[12] As is also apparent from Table 1, some of the results produced by SMOG differ from those obtained by the recently developed CAMGEC program;[37] the reason is that CAMGEC produces some duplicates during the generation of polycyclic structures (as was shown by the specially designed GI program).[38] As to Faulon's software,[39] the results produced by SMOG match those published for the deterministic version of that program,[39a] but the data available are too scarce for making any definite conclusions.

When Goodlist and/or Badlist are introduced, discrepancies between the numbers of isomers produced by SMOG and other programs (MOLGRAPH, AEGIS, *etc*.) are sometimes encountered. According to our experience, the main reason for this lies in the different ways of treating structural constraints within different programs.

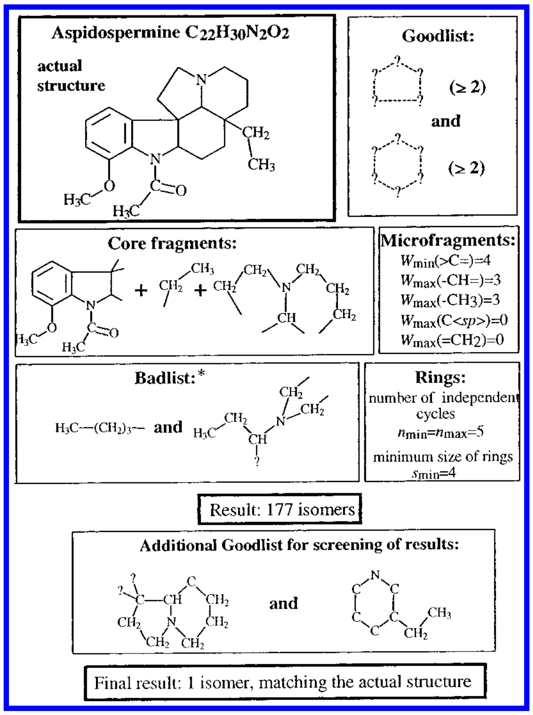


**Figure 6.** The use of SMOG for restoring the structure of a natural alkaloid by its empirical formula and structural information. **Note: in this problem, structures where any free valences of core fragments are saturated by hydrogens are also explicitly forbidden.*

For example, MOLGRAPH[2a] produces 320 isomers for the formula $C_8H_{16}O_2$ with the Goodlist containing the C═O and C−O−C fragments. At the same time, SMOG generates 425 isomers, because the possibility of overlapping between the above two fragments in the resultant structures is taken into account. If we explicitly forbid such overlapping, SMOG produces 320 isomers, exactly as MOLGRAPH.

Another reason for the difference between the results produced by SMOG and by other programs is due to the treatment of atoms with formal charges and other unusual valence states. For example, the well-known IGOR and RAIN programs[3] both generate 1806 isomers for $C_2H_2N_2O_2$ (no rings with triple bonds or cumulated double bonds are allowed; $N^+$, $N^-$, and $O^-$ may be present; total charge is zero). At the same time, SMOG generates 1900 isomers under the same conditions, because it makes allowance for the possible simultaneous presence of two $N^+$ and two $O^-$ atoms. Again, if we explicitly forbid this set of valence states, the results produced by SMOG will match those of IGOR and RAIN. At the same time, the majority of known generators totally disregard charged atoms or variable valences.

The examples below illustrate the applicability of SMOG for the solution of theoretical and practical chemical problems.[40]

**Example 1.** The CONGEN program, which had been developed within the DENDRAL project,[1d] was successfully used for elucidating the complex structure of aspidospermine (see the top of Figure 6).[1c]

After converting the structural information into the form that is acceptable for SMOG, we also conducted generation

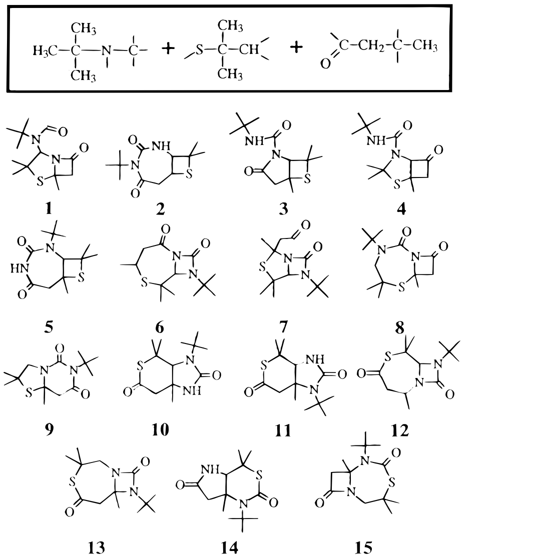


**Figure 7.** The family of $C_{13}H_{22}N_2O_2S$ isomers generated by SMOG. The set of core fragments is shown at the top.

using the formula of aspidospermine $C_{22}H_{30}N_2O_2$, as is shown in Figure 6. Note that it was a two-stage process: after a set of 177 isomers was generated, it was subjected to screening with the use of some supplementary structural information (this additional set of structural data had been experimentally obtained later than the main bulk of information).[1c] Only one isomer remains after screening; it matches the result obtained by CONGEN and the actual experimentally confirmed structure.

Similarly, SMOG successfully generated the structure of the tricyclic terpenoid uvidine A with the formula $C_{15}H_{24}O$. We used the set of selection rules that had been earlier used by the GENOA program for the same purpose.[1e] The final result is identical with that of GENOA and also matches the actual structure.

**Example 2.** The reaction between ethyl-$\beta$-aminocrotonate, α-mercaptoisobutyraldehyde, and *tert*-butyl isocyanide produces a compound with the formula $C_{13}H_{22}N_2O_2S$ (Figure 7, compound **1**).[41] An attempt to restore this structure by the RAIN2 computer program[42] had been undertaken by its authors,[3] and the resultant set of nine most probable $C_{13}H_{22}N_2O_2S$ isomers (Figure 7, **1**−**9**) included the actual structure.

To solve this problem using SMOG, we used the same set of nonoverlapping core fragments (incorporated from the starting reagents) as had been used by RAIN2. These fragments are shown at the top of Figure 7. We also included unstable fragments (such as gem-diols, hemiacetals, and the like) into Badlist.

However, the number of isomers produced by SMOG still exceeded several hundred. Since the complete list of selection rules used by the authors of RAIN2 was not available,[3] we used some constraints that are apparently common for the nine structures produced by that program:

(a) The only allowed type of multiple bonds is C═O; any bonds between heteroatoms are forbidden.

(b) The target structures contain two fused rings, each having the size from four to seven atoms; the sulfur atom is cyclic.

Under all these restrictions, SMOG produced 15 isomers, which included all nine molecular graphs provided by RAIN2 and also six chemically feasible interesting structures (Figure 7, **10**−**15**), distinguished by the presence of the O═C−S fragment. So, apparently, SMOG may be a suitable tool for elucidating the structures of reaction products.

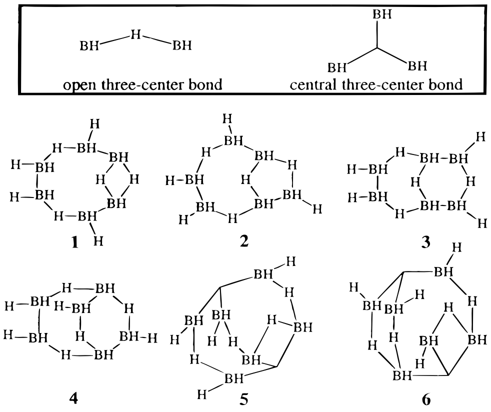


**Figure 8.** Some conceivable isomers generated by SMOG for the formula $B_6H_{14}$. The possible types of three-center bonds are shown at the top of the figure.

**Example 3.** Another interesting application of the RAIN software had been an attempt to predict the possible structure of the boron hydride $B_6H_{14}$, which is unknown as yet.[43] This problem is complicated by the peculiarities that characterize structures of boron hydrides: in addition to “normal” chemical bonds, these molecules also may include two types of three-center bonds, which are shown at the top of Figure 8.

The hypothetical structures had been assembled by RAIN2 from BH units and separate H atoms (possibly participating in three-center bonds).[3] The maximum permissible number of central three-center bonds had been set to 2. The important condition for the selection of appropriate structures was that, according to NMR evidence,[43a] the $B_6H_{14}$ molecule must contain strictly two different classes of B atoms and two classes of H atoms (the symmetry perception being confined to one sphere around each atom). As a result, 20 structures were hypothesized (structures **2**−**21** in ref 3).

We attempted to solve the same problem with the help of SMOG, stating that the “formal valence” of H may be either 1 or 2 and assuming that all bonds formally have single multiplicity. As to the problem of considering central three-center bonds, we solved it by introducing an imaginary trivalent chemical element X, which denotes the center of such a bond and should be surrounded by three BH units. We also formed Badlist, which contained substructures that are hardly probable from the chemical standpoint: acyclic bonds with bridging H atoms, H−H bonds, chains of two acyclic bonds, *etc*. During direct visual analysis of the resultant isomers, structures with the number of B or H classes different from two were rejected.

Finally, a set of 26 candidate structures remained. It includes all 20 structures produced by RAIN2 as well as six new molecular graphs which are shown in Figure 8. All of them indeed contain two types of B atoms and two types of H atoms, if classification of atoms is based only on their first environment.

A recent paper on computer generation of boron hydrides[43c] envisages a different concept of molecular symmetry, and the number of hypothetical structures is 48. All of them are also found among the molecular graphs produced by SMOG with the use of appropriate constraints.

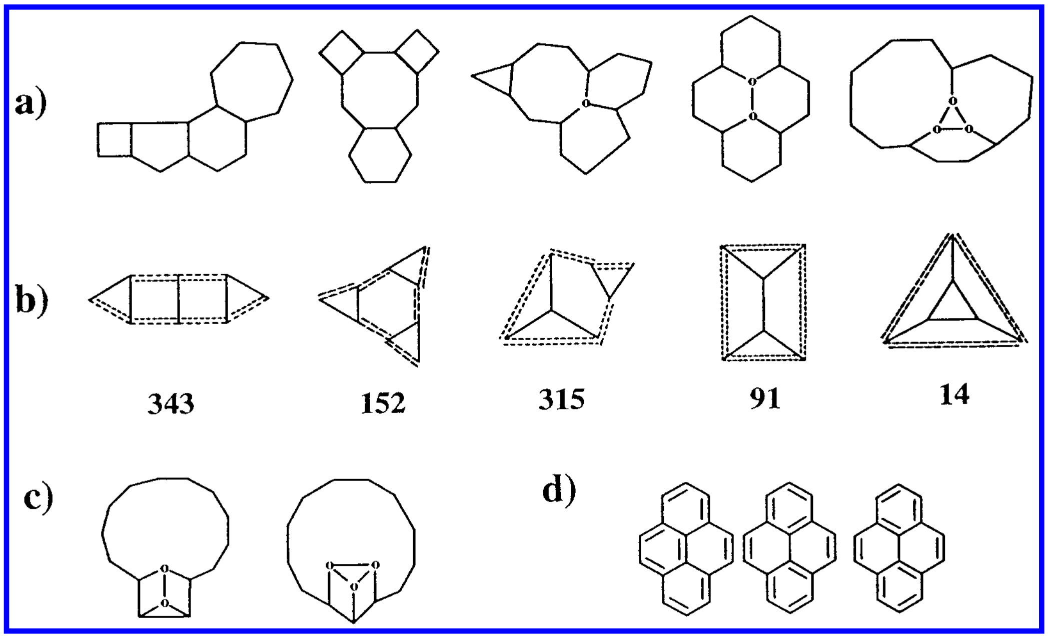


**Figure 9.** Generation of pyrene analogs: (a) the frameworks of pyrene isomers representing five different classes (internal vertices are marked by small circles); (b) the generalized subgraphs for these classes, contained in Goodlist during generation; (c) the structure that falls into two classes simultaneously; and (d) the three different ways to distribute double bonds in the pyrene framework.

**Example 4.** The algorithm of SMOG inherited many features from that of the GENM generator.[10] Numerous tests showed that the total numbers of isomers produced for one molecular formula without any structural constraints are equal for both programs.

The algorithmic advantage of SMOG over GENM lies in correct treatment of core fragments. Generation by GENM may become redundant in the presence of invariant substructures, whereas SMOG avoids redundancy by applying a rigorous test for strong canonicity.

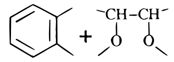
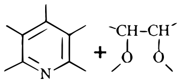
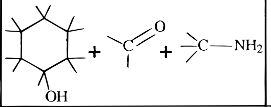
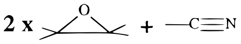
This statement is illustrated by the results in Table 2. Note that SMOG does not "skip" any actual molecules: when the results produced by GENM were directly checked for the presence of duplicates,[10c] the remaining numbers of isomers were the same as are produced by SMOG.

**Example 5.** The problem of enumerating analogs of pyrene ($C_{16}H_{10}$) is considered in ref 44a,b. By pyrene analogs, the authors of those papers mean 16-atom tetracyclic frameworks where each two rings out of four are either fused by exactly two adjacent C atoms or have no atoms in common. The positions of double bonds in the frameworks are not considered during enumeration.

The theoretical solution of this problem[44b] was based on the classification of all pyrene analogs by five classes: (1) no "internal" vertices, no branching; (2) no internal vertices, branching is present; (3) one internal vertex; (4) two internal vertices; and (5) three internal vertices forming a cycle of size 3. Examples of hypothetical frameworks for these five classes are shown in Figure 9a. Then a special combinatorial procedure was designed to enumerate isomers of all the five classes. The resultant numbers of isomers were 343, 152, 315, 91, and 14, respectively, and it was decided[44b] that the total number of the desired pyrene analogs was 343 + 152 + 315 + 91 + 14 = 965.

However, we decided to verify this result by SMOG, because our program provides an easy and straightforward procedure for enumeration of such frameworks. Firstly, the five aforementioned cyclic systems were specified by the generalized fragments shown in Figure 9b (as was mentioned above, a double dashed line may denote a connected chain of any length). Then, we conducted generation of tetracyclic structures with the use of SMOG, successively including these five fragments into Goodlist.

**Table 2.** Comparison of Some Results Obtained by GENM and SMOG for the Formula $C_9H_{11}NO_2$

| No. | Core fragments | Number of isomers | |
|---|---|---|---|
| | | GENM | SMOG |
| 1 |  | 220 | 215 |
| 2 |  | 2990 | 2945 |
| 3 |  | 36575 | 17468 |
| 4 |  | 8825 | 4457 |

For each of the fragments in Figure 9b considered separately, our results matched those obtained in ref 44b. However, when we simultaneously included all the five fragments into Goodlist (and united them by the OR operation, as is explained in section 4.2.1), the total number of isomers was 964 rather than 965. Direct analysis shows that the reason for this inconsistency is one structure that may be regarded as having either two or three internal vertices, depending on the representation (Figure 9c). That is, 964 is the true number of target structures. Hence, SMOG is a suitable tool for verifying the validity of various theoretical models. It is also capable of providing additional details concerning the problem in question, such as finding (if desired) all nonequivalent "Kekulé structures" for each of the generated frameworks. For example, there are three topologically nonequivalent distributions of double bonds corresponding to pyrene itself (Figure 9d).[45] However, note that the pyrene system is not regarded as aromatic by SMOG, because the number of $\pi$-electrons is $16 \neq 4n + 2$.

## CONCLUSION

SMOG is written in C/C++ and runs on IBM PC compatible computers with a EGA or VGA card and a mouse. It has a user-friendly interface and is supplied with on-line help. The program occupies about 1 Mb of disk space, and 2−3 free Mb for storing the resultant structures may be needed. Together with the manual, SMOG is accessible as is described in ref 46.

## REFERENCES AND NOTES

(1) (a) Masinter, M.; Sridharan, N. S.; Lederberg, J.; and Smith, D. H., Applications of Artificial Intelligence for Chemical Inference. XII. Exhaustive Generation of Cyclic and Acyclic Isomers. *J. Am. Chem. Soc.* **1974**, *96*, 7702−7714. (b) Masinter, M.; Sridharan, N. S.; Lederberg, J.; and Smith, D. H., Applications of Artificial Intelligence for Chemical Inference. XIII. Labeling of Objects Having Symmetry. *J. Am. Chem. Soc.* **1974**, *96*, 7714−7723. (c) Carhart, R. E.; Smith, D. H.; Brown, H.; Djerassi, C. Applications of Artificial Intelligence for Chemical Inference. XVII. An Approach to Computer-Assisted Elucidation of Molecular Structure. *J. Am. Chem. Soc.* **1975**, *97*, 5755−5762. (d) Lindsay, R. K.; Buchanan, B. G.; Feigenbaum, E. A.; Lederberg, J. *Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project*; McGraw-Hill: New York, 1980. (e) Carhart, R. E.; Smith, D. H.; Gray, N. A. B.; Nourse, J. G.; Djerassi, C. GENOA: A Computer Program for Structure Elucidation Utilizing Overlapping and Alternative Substructures. *J. Org. Chem.* **1981**, *46*, 1708−1718.

(2) (a) Kerber, A.; Laue, R.; Moser, D. Ein Redundanzfrier Strukturgenerator für Molekulare Graphen. *Anal. Chim. Acta* **1990**, *235*, 221−228. (b) Grund, R.; Kerber, A.; Laue, R. MOLGEN, ein Computer-algebra-system für die Konstruktion Molekularer Graphen. *Commun. Math. Chem.* (**MATCH**) **1992**, *27*, 87−131.

(3) Bauer, J.; Fontain, E.; Ugi, I. IGOR and RAIN—the First Mathematically Based Multipurpose Problem-Solving Computer Programs for Chemistry and Their Use as Generators of Constitutional Formulas. *Commun. Math. Chem.* (**MATCH**) **1992**, *27*, 31−47.

(4) Elyashberg, M. E.; Gribov, L. A.; Serov, V. V. *Molecular Spectral Analysis and Computer*; Mir: Moscow, 1980; p 152 (in Russian).

(5) (a) 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. (b) Lipkus, A. H.; Munk, M. E. Automated Classification of Candidate Structures for Computer-Assisted Structure Elucidation. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 9−18.

(6) (a) Bangov, I. P. Computer-Assisted Structure Generation from a Gross Formula. 3. Alleviation of the Combinatorial Problem. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 277−289. (b) Bangov, I. P. Computer-Assisted Structure Generation from a Gross Formula. 4. Fighting against Graph-Isomorphism Disease. *Commun. Math. Chem.* (**MATCH**) **1992**, *27*, 3−30.

(7) Luinge, H. J. AEGIS, a Structure Generation Program in Prolog. *Commun. Math. Chem.* (**MATCH**) **1992**, *27*, 175−189.

(8) (a) Kvasnicka, V.; Pospichal, J. Canonical Indexing and Constructive Enumeration of Molecular Graphs. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 99−105. (b) Kvasnicka, V.; Pospichal, J. An Improved Version of the Constructive Enumeration of Molecular Graphs with Prescribed Sequence of Valence States, *Chemom. Intell. Lab. Syst.* **1993**, *18*, 171−181.

(9) Faradjev, I. A. Generation of Nonisomorphic Graphs with a Given Distribution of Vertex Degrees. In *Algorithmic Investigations in Combinatorics*; Faradjev, I. A., Ed.; Nauka: Moscow, 1978; pp 11−19 (in Russian).

(10) (a) Molodtsov, S. G.; Piottukh-Peletsky, V. N. Generation of All Nonisomorphic Chemical Graphs from a Given Set of Structural Fragments. In *Algorithms for the Analysis of Structural Information*; Vychislitelnye Sistemy: Novosibirsk, 1984; Vol. 103, p 51 (in Russian). (b) Molodtsov, S. G. Computer-Aided Generation of Molecular Graphs. *Commun. Math. Chem.* (**MATCH**) **1994**, *30*, 213−224. (c) Molodtsov, S. G. Generation of Molecular Graphs with a Given Set of Nonoverlapping Fragments. *Commun. Math. Chem.* (**MATCH**) **1994**, *30*, 203−212.

(11) (a) Kudo, Y.; Sasaki, S., The Connectivity Stack: A New Format for Representation of Organic Chemical Structures. *J. Chem. Doc.* **1974**, *14*, 200−202. (b) Funatsu, K.; Miyabayashi, N.; Sasaki, S. Further Development of Structure Generation in the Automated Structure Elucidation System CHEMICS. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 18−28.

(12) (a) Bohanec, S.; Zupan, J. Structure Generation of Constitutional Isomers from Structural Fragments. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 531−540. (b) Bohanec, S.; Zupan, J. Structure Generator GEN. *Commun. Math. Chem.* (**MATCH**) **1992**, *27*, 49−85. (c) Bohanec, S. Structure Generation by the Combination of Structure Reduction and Structure Assembly. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 494−503.

(13) Lomova, O. A.; Sukhachev, D. V.; Kumskov, M. I.; Palyulin, V. A.; Tratch, S. S.; Zefirov, N. S. The Generation of Molecular Graphs for QSAR Studies by the Acyclic Fragment Combining. *Commun. Math. Chem.* (**MATCH**) **1992**, *27*, 153−174.

(14) Nilakantan, R.; Bauman, N.; Venkataraghavan, R. A Method for Automatic Generation of Novel Chemical Structures and Its Potential Applicability to Drug Discovery. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 527−530.

(15) Another restriction that is common for all generation algorithms is the consideration of connectivity: structures comprising two or more disconnected parts should be disregarded.

(16) The previous version of the program, named **GEM** (**GE**neration of **M**olecules), was mentioned in our earlier publications: Pivina, T. S.; Molchanova, M. S.; Shcherbukhin, V. V.; Zefirov, N. S. Computer Generation of Caged Frameworks Which Can Be Used as Synthons for Creating High-Energetic Materials. *Propellants, Explosives, Pyrotechnics* **1994**, *19*, 286−289; **1995**, *20*, 144−146, etc.

(17) (a) Faradjev, I. A. Constructive Enumeration of Combinatorial Objects. In *Algorithmic Investigations in Combinatorics*; Faradjev, I. A., Ed.; Nauka: Moscow, 1978; pp 3−11 (in Russian). (b) Zaichenko, V. A.; Ivanov, A. B.; Rozenfeld, M. Z.; Faradjev, I. A. Algorithm for Verifying the Canonicity of a Partially Filled Adjacency Matrix of a Graph. In *Algorithmic Investigations in Combinatorics*; Faradjev, I. A., Ed.; Nauka: Moscow, 1978; pp 19−25 (in Russian).

(18) The sets of valence states are selected in such a way that the total formal charge remains constant. For all examples in this paper, this charge is zero (electrical neutrality).

(19) As is apparent from Figure 2a, atoms in SMOG-generated structures are numbered in accordance with their valences: atoms with higher valences have smaller ordinal numbers.

(20) Among programs that use modifications of such an approach, we can mention the AEGIS software (ref 7).

(21) Lowler, E. L.; Wood, D. E. Branch-and-Bound Methods: A Survey. *J. Oper. Res. Soc. Am.* **1966**, *14*, 217−282.

(22) (a) Hendrickson, J. B.; Toczko, A. G. Unique Numbering and Cataloguing of Molecular Structures. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 171−177. (b) Hendrickson, J. B.; Parks, C. A. Generation and Enumeration of Carbon Skeletons. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 101−107.

(23) (a) Randić, M. Recognition of Identical Graphs Representing Molecular Topology. *J. Chem. Phys.* **1974**, *60*, 3920−3928. (b) Randić, M. On Canonical Numbering of Atoms in a Molecule and Graph Isomorphism. *J. Chem. Inf. Comput. Sci.* **1977**, *17*, 171−180.

(24) Other algorithms may use different criteria of canonicity; we can mention the criteria of maximum stack,[11] maximum upper triangle,[4,8b,22b] and maximum lower triangle.[8a] Similar criteria may be formulated with respect to the minimum matrix[23] or its parts. As is easy to show, the criteria of a maximum (minimum) matrix or maximum (minimum) upper triangle are equivalent.

(25) A similar modification of the canonicity criterion was mentioned in ref 10c, but it was not quite correctly implemented in that study, as is shown by example 4 of this paper.

(26) A ring is not regarded as aromatic by SMOG if additional short “cross-linking” bridges connect its atoms and thus form an extremely strained structure (for example, an extra chain of one or two atoms between the 1- and 4-positions of benzene). The consequence of this rule is that two aromatic rings cannot have more than two atoms in common, and these have to be adjacent. The SMOG user may specify the minimum lengths of cross-linking bridges at which the structure is already regarded as aromatic.

(27) An alternative way of treating hydrogens (suggested in ref 10) is their distribution among the vertices of the skeleton before the start of generation, so that new types of vertices appear: $CH_3$, $CH_2$, CH, OH, *etc*. Generation proceeds on the basis of a set of these new vertices. This procedure makes generation significantly faster but is inapplicable in many important cases, such as the presence of core fragments.

(28) SMOG explicitly assumes an additional limitation concerning the numbers of free valences at vertices of quasistructures: the number of hydrogens bound to any non-hydrogen atom may not exceed 3.

(29) The concept of weak canonicity, which was introduced by Faradjev,[9] modified by Molodtsov[10b] and used after further modification in our work, is similar to the notion of semicanonicity introduced by Kvasnicka.[8b] However, there are at least two points of difference: in Kvasnicka’s approach, classification of vertices by initial sets of stability (*i.e.*, by different vertex types) is disregarded, but at the same

time each atom must obey some explicit rules limiting the number and type of its multiple bonds.

(30) The method for combinatorial enumeration of permutations in the *S*(*I*,*J*) group and some of the ways to optimize this search are thoroughly described in ref 17b. We modified this procedure so that the *S*(*I*,*J*,*C*) group might be considered.

(31) The importance of considering alternative substructures was shown in refs 1e, 5a and numerous other works dealing with problems of structural generation.

(32) Hereafter, free valences and bonds with an unspecified terminus in the structural formulas of fragments may be saturated by any atom(s), including hydrogen, unless explicitly stated otherwise.

(33) Nechepurenko, M. I.; Popkov, V. K.; Mainagashev, S. M.; Kaul', S. B.; Proskuryakov, V. A.; Kokhov, V. A.; Gryzunov, A. B. *Algorithms and Programs for the Solution of Problems on Graphs and Networks*; Nauka: Novosibirsk, 1990 (in Russian).

(34) Arlazarov, V. L.; Zuev, I. I.; Uskov, A. V.; Faradjev, I. A. The Algorithm for Bringing Finite Nonoriented Graphs to the Canonical Form. *Zh. Vychisl. Mat. Mat. Fiz.* **1974**, *14*, 3, 737−743 (in Russian).

(35) Carhart, R. E. A Model-based Approach to the Teletype Printing of Chemical Structures. *J. Chem. Inf. Comput. Sci.* **1976**, *16*, 82−88.

(36) Shelley, C. A. Heuristic Approach for Displaying Chemical Structures. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 61−65.

(37) Contreras, M.; Valdivia, R.; Rozas, R. Exhaustive Generation of Organic Isomers. (a) 1. Acyclic Structures. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 323−330. (b) 2. Cyclic Structures: New Compact Molecular Code. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 481−491. (c) 3. Acyclic, Cyclic, and Mixed Compounds. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 610−616.

(38) Barone, R.; Barberis, F.; Chanou, M. Exhaustive Generation of Organic Isomers from Base 2 and Base 4 Numbers. *Commun. Math. Chem.* (**MATCH**) **1995**, *32*, 19−25.

(39) (a) Faulon, J.-L. On Using Graph Equivalence Classes for the Structure Elucidation of Large Molecules. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 338−348. (b) Faulon, J.-L. Stochastic Generator of Chemical Structure. 1. Application to the Structure Elucidation of Large Molecules. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 1204−1218.

(40) It would be appropriate to compare the performance of other generators with that of SMOG. Unfortunately, a rigorous comparison is impossible, because different authors obtained their results with the use of different computers; in addition, there is too little published data concerning the time of generation in the presence of structural constraints. So, we only made some qualitative estimates for the case when constraints are absent: SMOG generates molecules (with 10−20 non-hydrogen atoms) much faster than such generators as AEGIS,[7] CHEMICS,[11] GEN,[12] GI,[38] or the deterministic version of Faulon's generator[39a] but usually slower than the fast-working generators MOLGEN[2b] or GENM.[10b,c] However, this comparative slowness is largely due to the fact that SMOG is initially oriented at the presence of constraints; therefore, SMOG represents and calculates some characteristics of the partially generated structure at each stage of generation in such a way that various constraints may be quickly and easily considered. Such "data processing" is an important source of retardation: if we exclude it from SMOG, the performance of such a program in the absence of constraints becomes comparable to that of MOLGEN and GENM.

(41) Ugi, I.; Wischhöfer, E. Isonitrile, XI. Synthase einfacher Penicillansäure-Derivate. *Chem. Ber.* **1962**, *95*, 136−140.

(42) The generation of chemical isomers is one of the possible applications of the well-known IGOR and RAIN software.

(43) (a) Buehl, M.; Schleyer, P.; McKee, M., The Structures of the Hypho-Compound $B_5H_{12}^-$ and $B_6H_{14}$: Application of the combined *ab initio*/IGLO/NMR Method. *Heteroat. Chem.* **1991**, *2*, 4, 499−506. (b) Binder, H.; Brellochs, B.; Frei, B.; Simon, A.; Hettich, B. Über die ersten monosubstituierten Derivate von Triboran (7) -Kristall- und Molekülstruktur von Benzoyloxytriboran (7). *Chem. Ber.* **1989**, *122*, 6, 1049−1056. (c) Fontain, E. The $B_6H_{14}$ Problem: Generation of a Catalogue of Conceivable Isomers. *Heteroat. Chem.* **1994**, *5*, 61−64.

(44) (a) Cyvin, S. J.; Cyvin, B. N.; Brunvoll, J. The Number of Pyrene Isomers is Still Unknown. *Commun. Math. Chem.* (**MATCH**) **1994**, *30*, 73−77. (b) Cyvin, S. J.; Brunvoll, J.; Cyvin, B. N. Number and Forms of Tetracyclic Polygonal Pyrene Isomers. *Commun. Math. Chem.* (**MATCH**) **1995**, *32*, 59−70.

(45) Actually, there are six Kekulé structures for the pyrene system, but they represent three pairs of topologically equivalent mirror images. Since SMOG always avoids topologically equivalent structures, it considers only one formal distribution of double bonds out of each such pair.

(46) This program is available from the Archives of Computational Chemistry list (top page is http://ccl.osc.edu/chemistry.html). To go directly to the program, use the following URLs: gopher://infomeister.osc.edu:73/11/software/MS-DOS/SMOG or ftp://ccl.osc.edu/pub/chemistry/software/MS-DOS/SMOG. It is also available via e-mail if the message [select chemistry; limit 2Mbytes, cd software/MS-DOS/SMOG, get*; quit] is sent to MAILSERV@ccl.osc.edu.

CI950393Z