match no. 30 pp. 213-224 1994

COMPUTER-AIDED GENERATION OF MOLECULAR GRAPHS

Sergey G. Molodtsov
Institute of Organic Chemistry
Siberian Division of the Russian Academy of Science
9, Prospekt akad. Lavrentieva, Novosibirsk 630090, Russia

(received: July 1993)

Abstract

An effective algorithm and a program for a generation of all nonisomorphic molecular graphs from a set of labeled vertices with a given valence are described. The algorithm allows apriori specification of information about forbidden and necessary fragments in the graphs to be generated. Examples of applications are given.

1. Introduction.

The problem of the generation of chemical structures is of much interest. It arises, for example, in structure elucidation of chemical compounds with the help of their molecular spectra, or in the design of structural formulas of compounds hypothetically having a given type of biological activity or some other property. In both cases a researcher needs to construct a complete set of structural formulas corresponding to the given molecular formula and satisfying structure limitations. For the computer-aided handling of structural formulas researchers widely employ the structural representation as a molecular graph (labeled multigraph) whose vertices correspond to atoms or microfragments, and the edges to bonds of a chemical compound.

There are some published algorithms and programs for molecular graph generation [1-4]. However, due to importance of this problem, there is constant interest in developing more efficient algorithms providing structure generation using various starting data. This paper describes an efficient algorithm for the generation of all nonisomorphic molecular graphs (their canonical adjacency matrices) from a set of labeled vertices with a given valence, and reports the results of the generation of certain classes of isomers.

The suggested algorithm of molecular graph generation is based on the enumerative variant of the branch-and-bound method [5]. The main idea consists in choosing from each set

of mutually isomorphic graphs one graph (canonical graph) satisfying a certain criterion. Only canonical graphs in the process of generation are constructed. An advantage of this approach is that the researcher need not verify the structure for isomorphism with the previously constructed graphs.

2. Definitions.

Let $G=(V,E)$ be a multigraph (undirected, without loops but with multiple edges), where V is the set of labeled vertices ($|V|=n$), and E the set of edges. For brevity G below is called a graph. The number of incident edges of the vertex is called valence.

Let us divide all vertices of V into classes of vertices having the same labels and valences. Define some order of the classes: $V_1 < V_2 < ... < V_p$. We assume below that the order of the classes remains fixed. Each class is specified by a label and the valence of its constituent vertices. We'll consider only the graphs specified by a given set of vertices V and having the same ordered classes.

Let us number the vertices of the graph. The vertex numbered i will be denoted by v_i and its valence by val_i. Clearly, the graph vertices may be numbered by $n!$ different ways.

DEFINITION 1. We call the enumeration of the graph vertices *admissible* if the vertices of the lower class have a smaller index.

Below we'll consider only admissible enumerations. Thus, the set $N=\{1,2,...,n\}$ is divided into the classes $N_1,N_2,...,N_p$, where N_k is the set of indices of the vertices of V_k.

A graph may be represented by it adjacency matrix $A=(a_{ij})$, which is a symmetric square $(n \times n)$ matrix, where a_{ij} equals the multiplicity (the number) of edges between the v_i-th and the v_j-th vertex, in particular $a_{ij}=0$ if the vertices are nonadjacent. Evidently, the sum of the elements in each row of the matrix equals the valence of the corresponding vertex: $\sum_{j=1}^{n} a_{ij}=val_i$ for each $i \in N$. The representation of a graph as an adjacency matrix is not unique, it depends on the way of enumeration of vertices.

DEFINITION 2. The adjacency matrix corresponding to the admissible enumeration of vertices is called *admissible*.

Now we introduce a linear order on the set of matrices of the same size. The matrix $A=(a_{ij})$ is assigned the n^2-component vector $W(A)=(a_{11},a_{12}...,a_{nn})$, by concatenation of the rows. We say that $A<B$ if and only if $W(A)<W(B)$, where the order of the set of vectors is specified lexicographically.

Let $S(N_k)$ be the symmetric group of permutations whose action on the adjacency matrix leads to the same permutation of rows and columns of the matrix with indices from N_k, i.e. to renumbering of vertices from V_k. Then the action of the elements of the

group $S(N)=\bigoplus_k S(N_k)$ on the adjacency matrix leads to the admissible renumbering of the vertices of the graph.

DEFINITION 3. The graphs G and H are called *isomorphic* if there is a one-to-one correspondence between vertices of the graphs with preserved adjacency and labels of vertices.

It immediately follows from the definition that G and H are isomorphic if and only if there exists a permutation $g\in S(N)$ converting their adjacency matrices into each other: $A=gBg^{-1}$, where A and B are the adjacency matrices of G and H respectively.

DEFINITION 4. The adjacency matrix A of a graph is called canonical if it is not increased by any permutation from $S(N)$, i.e. $A \geq gAg^{-1}$ for every $g\in S(N)$.

Clearly, two different canonical adjacency matrices correspond to nonisomorphic graphs.

Let us consider *partially filled* matrices $B=(b_{ij})$, which are symmetric square (n×n) matrices with a zero diagonal, where b_{ij} is either a number or a special symbol δ [6]. The matrix elements equal to δ are assumed to be undefined. We call the element b_{ij} filled if b_{ij} is a number, and unfilled if $b_{ij}=\delta$. The row (column) is called filled if all the elements are filled, and unfilled otherwise.

Let $I_k=I_k(B)$, $J_k=J_k(B)$ be the sets of indices of the filled and unfilled rows of B in the class N_k, where $N_k=I_k \cup J_k$. Let $I(B)=\bigcup_k I_k$, $J(B)=\bigcup_k J_k$ be the sets of indices of the filled and unfilled rows.

DEFINITION 5. A partially filled matrix A is called a *complement* of the partially filled matrix B if they coincide on the filled elements of B.

DEFINITION 6. A partially filled matrix B is called *admissible* if there exists an admissible adjacency matrix A being its complement.

Obviously, there are the following conditions of admissibility for the partially filled matrices:

CONDITION 1. The sum of all elements of each filled row $i\in I(B)$ is equal to the valence of the corresponding vertex: $\sum_{j=1}^{n} b_{ij}=val_i$.

CONDITION 2. (Minimal admissibility). The sum of filled elements of each unfilled row $i\in J(B)$ does not exceed the valence of the corresponding vertex: $\sum_{b_{ij}\neq\delta} b_{ij}\leq val_i$.

Define $R=(r_{ij})$ to be a matrix of maximal possible multiplicity of edges between graph vertices. We assume the maximal possible multiplicity of edges between all vertices of the

classes N_k and N_l to be constant: $r_{ij} = const$ for any $i \in N_k$ and $j \in N_l$. In particular, using R one can forbid the formation of edges between some vertices ($r_{ij}=0$). As the multiplicity of edges between two vertices does not exceed the valence of each vertex, then $r_{ij} \le \min(val_i, val_j)$.

Denote by $fv_i = val_i - \sum_{b_{ij} \neq \delta} b_{ij}$ the free valence of the vertex v_i. Let b_{ij} be the unfilled element of B. Clearly, r'_{ij} the maximal multiplicity of edges between the v_i-th and v_j-th vertex does not exceed $\min(r_{ij}, fv_i, fv_j)$. Define

$$r'_{ij} = \begin{cases} \min(r_{ij}, fv_i, fv_j) & , \ b_{ij} = \delta \\ b_{ij} & , \ b_{ij} \neq \delta \end{cases}$$

Let us formulate a third condition of admissibility for the partially filled matrix.

CONDITION 3. (Maximal admissibility). The sum of all maximal multiplicitis of edges of the v_i-th vertex, $i \in J(B)$, is not less than its valence: $\sum_{j=1}^{n} r'_{ij} \ge val_i$.

Let $S(I_k)$, $S(J_k)$ be symmetric groups of permutations whose action consists in permutation of filled and unfilled rows and columns of the partially filled matrix B with indices from N_k. Consider the permutations group $S(I,J) = \bigoplus_k (S(I_k)+S(J_k))$ permuting independently the filled and unfilled rows and columns of B within the classes.

DEFINITION 7. The partially filled matrix B is called *strongly canonical* if it is not increased by any permutation from $S(I,J)$, i.e. $B \ge gBg^{-1}$ for every $g \in S(I,J)$.

STATEMENT. Any adjacency matrix being a complement of a not strongly canonical matrix is noncanonical.

The proof immediately follows from the fact that $S(I,J)$ is a subgroup of $S(N)$.

3. Algorithm of Graphs Generation.

Obviously, it is necessary and sufficient that the following conditions are satisfied for the existence of a graph consisting of a given set of vertices:

1) The connection of edges: the sum of valences of all vertices $\sum_{j=1}^{n} val_i$ is even;

2) The absence of loops: the valence of each vertex does not exceed the number of edges of the graph equal to $\frac{1}{2} \sum_{j=1}^{n} val_i$.

Let $B=(b_{ij})$ be a partially filled matrix, $m = \min J(B)$ the first unfilled row. Introduce on the set $J=J(B) \setminus m$ the equivalence relation:

$$j_1 \sim j_2 \Leftrightarrow \exists\, k:\ j_1, j_2 \in J_k \text{ and } \forall\, i:\ i < m\ b_{ij_1} = b_{ij_2},$$

i.e. the unfilled rows are equivalent if they belong to the same class and the first m-1 elements of the corresponding columns in them coincide. Then J is divided into equivalence classes $\{Y_r\}$, called in [7] "*pieces of stability*". Let $S(Y_r)$ be a symmetric group of permutations, whose elements permute the rows and columns of B with indices from Y_r. Consider a group $S(Y) = \bigoplus_r S(Y_r)$, whose elements permute the unfilled rows and columns independently and similarly within all pieces of stability, in particular, the elements of its m-th row within pieces of stability. Denote the m-th row of A by A_m.

DEFINITION 8. The matrix A is called a *weakly canonical complement* of the matrix B if it is an admissible complement with the filled m-th row and $A_m \geq (gAg^{-1})_m$ for every $g \in S(Y)$.

Clearly, the matrix that is not a weakly canonical complement is not a strongly canonical matrix. The maximal value $(gAg^{-1})_m$, $g \in S(Y)$ is achieved when the elements of the m-th row of A are arranged in decreasing order within each piece of stability. Thus, every weakly canonical complement of B corresponds to the decreasing order of the m-th row elements within every piece of stability.

Note that in some cases all admissible complements of B contain similarly filled rows that are not filled in B. This holds for those unfilled rows of B for which the following conditions are satisfied:

CONDITION 4. (Minimal forcing). The sum of filled elements of each unfilled row $i \in J(B)$ is equal to the valence of the corresponding vertex: $\sum_{b_{ij} \neq \delta} b_{ij} = val_i$.

CONDITION 5. (Maximal forcing). The sum of all maximal multiplicity of edges of the v_i-th vertex, $i \in J(B)$, is equal to the valence of the vertex: $\sum_{j=1}^{n} r'_{ij} = val_i$.

In the former case, all unfilled elements of the row must be zero and in the latter case, they must be equal to maximal multiplicity of edges. The rows for which the forcing conditions are satisfied may be filled. We call this process the *forced filling* of B.

The algorithm of generation is a stepwise procedure. Each step involves the construction of all weakly canonical complements of a strongly canonical matrix obtained at a previous step, and a selection of the strongly canonical complements.

Let s be a step of the procedure; A^s a strongly canonical matrix obtained at the step s, $m^s = \min J(A^s)$ the first unfilled row of A^s. It is initially assumed that B is unfilled: $b_{ij} = \delta$ at $i \neq j$. Then the generation algorithm is as follows:

1. Verification of admissibility of B. **If** the matrix is inadmissible, **then** termination: generation of graphs from the given set of vertices is impossible.
2. Forced filling of B.
3. **If** B is completely filled, **then** termination: there is only one graph (B is its adjacency matrix).
4. $s = 0$
5. $s = s + 1$
   $A^s = B$
   $m^s = \min J(B)$
6. Filling of the m^s-th row (column) of B maximal way, go to 10.
7. **If** $s = 1$, **then** termination: all nonisomorphic graphs (their canonical adjacency matrices) are constructed.
8. $s = s - 1$
   $B = A^s$
9. Filling of the m^s-th row (column) of B lower way. **If** the filling is impossible, **then** go to 7.
10. Verification of admissibility of B. **If** the matrix is inadmissible, **then** go to 9.
11. Forced filling of B.
12. Verification of strongly canonical of B. **If** the matrix is not strongly canonical, **then** go to 9.
13. **If** B is completely filled, **then** the next nonisomorphic graph is constructed (B is its canonical adjacency matrix), go to 9; **else** go to 5.

The Figure shows the construction of the first canonical adjacency matrix from the given set of vertices $V = \bigcup_{k=1}^{5} V_k$ and the matrix R. For better perception the unfilled elements of partially filled matrices are shown as dots. The rows and columns filled in a nonforced way are boldfaced. Near the matrices are their corresponding graphs. In the given example the adjacency matrix A was constructed in two steps.

The verification procedure of the partially filled matrix $B = A^s$ for being strongly canonical uses an algorithm [8] modified in the following way. Clearly, the force-filled rows of B are determined by the rows with indices m^k, $k = 1,s$. Consequently, when searching for the permutation from $S(I,J)$ that increases B it is sufficient to verify the permutations of rows and columns only in places of the rows m^k, $k = 1,s$, as the force-filled rows of B and gBg^{-1} coincide, when the rows m^k, $k = 1,s$ do.

$V_1 = \{ >C< \}$ $N_1 = \{ 1 \}$

$V_2 = \{ >CH- \}$ $N_2 = \{ 2 \}$

$V_3 = \{ -CH_2-, -CH_2- \}$ $N_3 = \{ 3, 4 \}$

$V_4 = \{ -CH_3 \}$ $N_4 = \{ 5 \}$

$V_5 = \{ -OH \ , -OH \}$ $N_5 = \{ 6, 7 \}$

$$R = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$A^1 =$

	Y_1	Y_2	Y_3	Y_4
	0	· ·	·	· ·
	·	0	·	· ·
	·	·	0	· ·
	·	·	·	0 · ·
	·	·	·	0 0 0
	·	·	·	0 0 0
	·	·	·	0 0 0

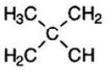
$m^1 = 1$

$Y_1 = \{ 2 \}$

$Y_2 = \{ 3, 4 \}$

$Y_3 = \{ 5 \}$

$Y_4 = \{ 6, 7 \}$



$A^2 =$

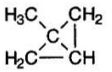
0	**1**	**1**	**1**	1	**0**	**0**
1	0	·	·	0	·	·
1	·	0	·	0	·	·
1	·	·	0	0	·	·
1	0	0	0	0	0	0
0	·	·	·	0	0	0
0	·	·	·	0	0	0

$m^2 = 2$

$Y_1 = \{ 3, 4 \}$

$Y_2 = \{ 6, 7 \}$

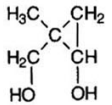
For row 5, the minimal forcing condition is valid.



$\bar{A} =$

0	**1**	**1**	**1**	1	**0**	**0**
1	0	**1**	**1**	0	**0**	**0**
1	**1**	0	·	0	·	·
1	**1**	·	0	0	·	·
1	0	0	0	0	0	0
0	**0**	·	·	0	0	0
0	**0**	·	·	0	0	0

Inadmissible matrix: for row 6 and 7 the maximal admissibility condition is invalid.



$A =$

0	**1**	**1**	**1**	1	**0**	**0**
1	0	**1**	**0**	0	**1**	**0**
1	**1**	0	0	0	0	0
1	**0**	0	0	0	0	1
1	0	0	0	0	0	0
0	**1**	0	0	0	0	0
0	**0**	0	1	0	0	0

For row 3 and 6 the minimal forcing condition is valid.

For row 4 and 7 the maximal forcing condition is valid.

Figure. Construction of the first adjacency matrix.

An advantage of this approach is an essentially reduced search tree of permutations from $S(I,J)$ with many force-filled rows. In particular, all the rows standing at the end of the adjacency matrix and corresponding to univalent vertices are force-filled (see the Figure).

Moreover, this procedure uses information about the orbits of permutations group $S(I,J)$ to cut off branches in the search tree of permutations.

If the maximal possible multiplicity of edges between two vertices equals the minimal number of their valences, then all graphs are generated, including the unconnected ones. However, molecular graphs are generally connected graphs. Therefore, after the force filling module (item 11) in the generation algorithm it is necessary to include the module of connectivity verification of a subgraph represented by the partially filled matrix B:

11' Search for a connected component containing the vertex v_1.
If the component contains all vertices, **then**
the graph is connected;
else if the component contains only the vertices
whose respective rows in B are filled, **then**
the graph is unconnected, go to 9;
else
nothing can be said about connectivity of the graph.

For generation of only connected graphs, the maximal possible multiplicity of edges between vertices of the same valence may be specified as equal to the valence of vertices minus one.

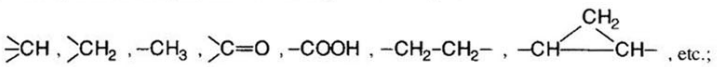
4. Discussion.

The structures generated are primarily determined by the set of vertices: the number of vertex classes, the number of vertices in each class and their valences. In particular, this set of vertices may be represented by a molecular formula. For example, when all benzene isomers with molecular formula C_6H_6 are to be generated, the set of vertices is specified as follows:

class 1: 6 vertices C with valence 4
class 2: 6 vertices H with valence 1.

The graph vertices may be not only atoms but also structural fragments satisfying the following conditions:

1) the fragment has only equivalent free edges, for example,



2) when some free edges of the fragment are connected, the remaining free edges must retain equivalence, for example, >CH-CH< may not be represented by one vertex;

3) the fragment must not be generated by linking other fragments and/or atoms with each other.

Molecular formulas of organic compounds generally contain many hydrogen atoms. To specify the class of hydrogen vertices, the program provides for a procedure of preliminary distribution of hydrogens among other vertices to generate a set of microfragments (new classes of vertices). Such microfragments correspond to the starting vertices with a definite number of adjacent hydrogens. The graphs (their canonical adjacency matrices) are generated from each set of microfragments. The use of this procedure essentially saves computer time due to the reduced number of vertices and the smaller size of the adjacency matrix. For example, the time of generation of 1230 isomers of molecular formula C_7H_6 with hydrogen atoms represented as separate vertices is 9.6 sec, while using this procedure reduces the time to 1.4 sec on IBM PC AT 386 (20 MHz).

It has been suggested above that the matrix R gives the maximal possible multiplicity of edges between vertices. It may also be used to specify the necessary multiplicity of edges. If it is known that the multiplicity of edges between any vertices from V_k and V_l is t, then it is given that $r_{ij} = -t$ for any $i \in N_k$ and $j \in N_l$. Then it is possible to fix the corresponding elements of the partially filled matrix A^1, keeping them constant. When filling a row at any step, the information specified in R is taken into account. The elements of the partially filled matrix A^S corresponding to nonpositive elements of the matrix R are disregarded.

This technique essentially saves computer time. Table 1 lists the data on generation of some isomers, both in an unrestricted variant and with known information on heteroatoms environment.

Moreover, specification of the necessary multiplicity of edges between vertices allows introduction into the adjacency matrix of the information on the large fragments of the graphs to be generated. This question will be discussed in detail in [9].

Table 1.

Molecular formular	GOODLIST	Number of isomers	Time (s)
$C_{10}H_{20}O$	—	13372	38.0
$C_{10}H_{20}O$	–OH	6355	18.0
$C_{10}H_{20}O$	$>C=O$	405	1.5
$C_{10}H_{20}O$	$-C-O-C-$	6612	18.7
$C_5H_{10}N_2O$	—	33689	32.4
$C_5H_{10}N_2O$	$-C\equiv N$	31	0.1
$C_5H_{10}N_2O$	$(-NH_2)_2$, $>C=O$	213	0.2
$C_5H_{10}N_2O$	–N=N– , –O–	690	1.0

The graph generation time depends much on the order of vertex classes. The least time is achieved when vertex classes are arranged in increasing order of the number of vertices belonging to these classes. The generation time is saved due to the reduced search tree on verification of strong canonicity of the partially filled matrices: in each row to be filled, the first piece of stability is reduced. Secondly, in the case of the same number of vertices, the classes are to be ordered according to decreasing valence of their vertices. The number of steps required for the construction of canonical adjacency matrices decreases, because at each step the number of edges to be allocated increases. Table 2 shows the results of graph generation depending on mutual arrangement of the classes:

Table 2.

Molecular formular	Number of isomers	Time
$N_2O_3C_5$	83751	1:59.7
$O_3N_2C_5$	83751	2:16.5
$C_5N_2O_3$	83751	2:31.2
$F_1C_6Cl_5$	685	1.6
$C_6F_1Cl_5$	685	4.4
$F_1Cl_5C_6$	685	5.7

Therefore we have added to the algorithm a procedure of ordering the vertex classes according to the above-mentioned criteria. In the case of prior allocation of hydrogens among other vertices, this procedure works for each set of microfragments constructed.

5. Conclusion.

The GENM program implementing the above-described algorithm is written in FORTRAN and C. The computer code of the program translated by Microsoft C 5.1 takes 26 Kb. The internal arrays require approximately $5n^2$ bytes more, where n is the number of vertices of molecular graphs being generated.

The program was tested on published series of isomers [10] and has shown high efficiency. Table 3 summarizes the data on generation of isomers with molecular formulas $C_nH_{2n+4}N_2O$, $C_nH_{2n-2}O$ and C_nH_n:

Table 3.

n	Number of isomers $C_nH_{2n+4}N_2O$	Time	Number of isomers $C_nH_{2n-2}O$	Time	Number of isomers C_nH_n	Time
2	31	0.0	3	0.0	1	0.0
3	102	0.1	13	0.0		
4	333	0.4	55	0.1	11	0.0
5	1041	1.6	205	0.2		
6	3218	6.0	747	0.8	217	0.2
7	9780	22.9	2589	3.1		
8	29487	1:26.0	8796	12.7	7437	8.9
9	88122	5:20.0	29172	50.7		
10	261876	19:51.0	95312	3:14.0	369067	8:21.0
11	774060	1:13:32.0	306958	12:30.0		
12	2278754	4:28:09.0	977939	47:10.0	23862255	9:50:43.0

These data make it possible to evaluate the efficiency of the program depending on the number and valence of vertices. Proceeding from the first molecular formula, one can construct only acyclic isomers, the second molecular formula corresponds to isomers having only two multiple edges or cycles, and the third one to those having $n/2+1$ multiple edges or cycles.

It is seen from the examples that the algorithm suggested shows a high computation speed and may be used in systems of structure elucidation with the help of molecular spectra, expert systems for design of compounds with a given type of property, etc.

Acknowledgment.

I would like to thank A.A.Dobrynin (Institute of Mathematics Siberian Division of the Russian Academy of Science), Prof. A.Kerber (University of Bayreuth) for their valuable remarks and suggestions, and also the German VW-Stiftung for financial support.

References.

1. R.E.Carhart, D.H.Smith, H.Brown and C.Djerassi, Applications of Artificial Intelligence for Chemical Inference. XVII. An Approach to Computer-Assisted Elucidation of Molecular Structure, J.Am.Chem.Soc., 97 (1975), 5755-5762.
2. R.E.Carhart, D.H.Smith, N.A.B.Gray, J.G.Nourse and C.Djerassi, GENOA: A Computer Program for Structure Elucidation Utilizing Overlapping and Alternative Substructures, J.Org.Chem., 46 (1981), 1708-1718.
3. K.Funatsu, N.Miyabayashi and S.Sasaki, Futher Development of Structure Generation in the Automated Structure Elucidation System CHEMICS, J.Chem.Inf.Comput.Sci., 28 (1988), 18-28.
4. B.D.Christie and M.E.Munk, Structure Generation by Reduction: A New Strategy for Computer-Assisted Structure Elucidation, J.Chem.Inf.Comput.Sci., 28 (1988), 87-93.
5. I.A.Faradzev, Constructive enumeration of combinatorial objects, Problems Combinatoires et Theorie des Graphes. Colloque Internationale CNRS 260, CNRS, Paris (1978), 131-135.
6. I.A.Faradzev, Generation of Nonisomorphic Graphs with a Given Degree Sequence, in: Algorithmic Studies in Combinatorics, Ed.Nauka, Moscow (1978), 11-19 (in Russian).
7. M.Z.Rozelfeld, Construction and Properties of Some Classes of Strongly Regular Graphs, Uspehi Mat. Nauk, 28 (1973), 197-198 (in Russian).
8. V.A.Zajchenko, A.B.Ivanov, M.Z.Rozelfeld and I.A.Faradzev, Algorithm of Canonicity Verification of the Partially Filled Adjacency Matrix of a Graph, in: Algorithmic Studies in Combinatorics, Ed.Nauka, Moscow (1978), 19-25 (in Russian).
9. S.G.Molodtsov, Generation of Molecular Graphs with a Given Set of Nonoverlapping Fragments, (in preparation).
10. A.Kerber, R.Laue and D.Moser, Ein Strukturgenerator fur molekulare Graphen, Anal.Chimica Acta, 235 (1990), 221-228 (in German).