Exhaustive Generation of Organic Isomers. 1. Acyclic Structures

M. L. Contreras,* R. Valdivia, and R. Rozas

Chemistry Department, University of Santiago, Casilla 5659, Santiago 2, Chile

Received December 16, 1991

The system reported here describes selective, exhaustive, and nonredundant generation and counting algorithms for acyclic connectivity isomers associated with any molecular formula. Isomer structures can have multiple bonds and heteroatoms with mixed valences, as in the case of thiosulfonic acids. Structural isomer characteristics (IC) for each molecular formula are determined according to an expression derived from basic graph principles. The generation process uses a tuple notation and the concept of lexicographic order (see ref 13) and is done in three steps: (i) generation of the skeleton of the acyclic structure; (ii) incorporation of heteroatoms to the structure; (iii) incorporation of multiple bonds. Isomer redundant filtering processes and algorithms for doing a single- or a multiple-pattern restriction over the structures to be generated were developed. The code describing the generated isomers is compact and allows for both efficient molecular database storage and interaction with a graphic interface and with different calculation modules of CAMD such as those of topological indexes, molecular volume, and other molecular properties derived from semiempirical and ab initio methods. The system is therefore of great utility in structure elucidation, in organic synthesis, and especially in molecular design.

INTRODUCTION

Enumeration and generation of structural isomers of organic compounds has been an interesting research subject to many chemists and mathematicians for a long time. Polya's method[1] has been widely used for enumeration of different types of chemical structures like alkanes, polyhexes, and other families. The method has also been used for heteroannulenes.[2,3]

Generation of isomers has been envisaged as an important tool in organic syntheses by computer[4,5] and in structure elucidation problems.[6,7] Graph theory has shown to be very useful in this context[4,8,9] and in other structural problems.[10,11] We are extending its use here to the exhaustive generation of acyclic isomers as the first step toward generation of molecular structures of any complexity[12] to be used as a molecular design tool.

Generation of isomers has to be exhaustive and nonredundant. The first job is easier to achieve than the second one. From the point of view of memory use and disk access time, it is clear that it is inconvenient to generate all of the structures first and then to search for the redundant ones to be eliminated. The search, during the generation process, within the previously created structures is also inappropriate. Many approaches have been used for solving this problem. Most of them use the following in the process of eliminating redundant structures: a canonical representation of the structure itself, i.e., a tuple with a maximum reverse lexicographic order;[4,13] a maximal adjacency matrix;[6] a maximal upper-right triangle of the adjacency matrix;[14] and a greatest connectivity stack.[7]

In this work, the exhaustive irredundant generation of acyclic isomers having one or more heteroatoms in the skeleton, multiple bonds, and atoms with multiple valences is presented. The generation process is complemented with filtering algorithms and selective structural constraints which are applied over the isomer that is being designed, making in this way a more efficient process. In addition, a compact unique code for the topological structures is described. The code allows for efficient database storage and for interacting with a graphic interface and with other modules of CAMD[15] such as those of calculations of topological indexes, molecular volume, and other molecular properties derived from semiempirical and ab initio calculations.

SYSTEM DESCRIPTION

The system developed here allows for the generation of all the open-chain isomeric structures associated with a particular molecular formula that include heteroatoms, with multiple valences, and multiple bonds. The program is called CAMGEC for computer-assisted molecular generation and counting.

Generation of structures is done in a selective way according to user requirements. For instance, the user might require that molecules generated by the program have a defined number of double or triple bonds, one or more heteroatoms, or some defined substructures. In other words, generation is done upon a set of isomer characteristics selected by the user from a list automatically deduced from the molecular formula at the beginning of the generation process and also upon one or more structural patterns or selection criteria. Structures generated by this process are represented or codified in a tuple notation. Finally, a decoding process is in charge of writing each structure in a format that is ready to be interpreted and displayed by a graphic interface (see Figure 1).

The system has been designed in a friendly way, and it interacts with the user through menus.

1. Isomer Characteristics. Generation of structures starts after receiving the molecular formula. This information is processed, and the basic characteristics of the molecules are determined. The procedure for that is based on general graph theory,[16] which establishes the following relationship between the sum of the node degree (deg p) and the number of the graph edges (q):

$$\sum \deg p = 2\sum q \qquad (1)$$

On the other hand, for a tree, which is an important class of graph,[16] there exists a relationship between the sum of the

0095-2338/92/1632-0323$03.00/0

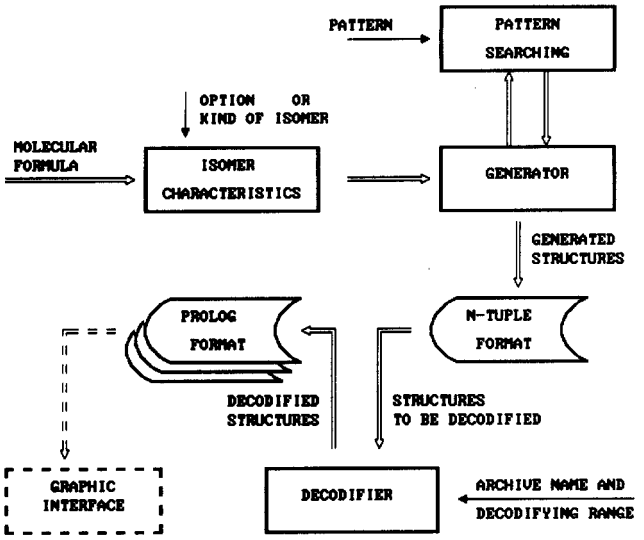


Figure 1. General block diagram for acyclic isomer generation.

tree lines (q) and the number of the tree nodes (p):[17]

$$\sum q = \sum p - 1 \tag{2}$$

For each point of a saturated molecular structure, the valence of the element (v) equals the degree of that point:

$$v = \deg p \tag{3}$$

If each pair of atoms is connected by a unique chain, then, for n atoms in a structure

$$\sum \deg p = \sum (n_i v_i) \tag{4}$$

where n_i is number of atoms of the element i, and v_i is valence of the element i. From eqs 1 and 2

$$\sum \deg p = 2(\sum p - 1) \tag{5}$$

and from eq 4

$$\sum (n_i v_i) = 2(\sum p - 1) \tag{6}$$

For unsaturated molecules, however, graph theory considers a double bond like a cycle between two points and a triple bond like two cycles between two points. Therefore, for unsaturated molecules, expression 6 is written as:

$$\sum (n_i v_i) > 2(\sum p - 1) \tag{7}$$

and new edges designated here as isomer characteristics (IC) have to be added to the graph for keeping the equality as in eq 6. In this case the graph degenerates to a cyclic one.[17] Each additional edge adds two to the total number of graph degrees. Then

$$\sum (n_i v_i) = 2(\sum p - 1) + 2\,\mathrm{IC} \tag{8}$$

as

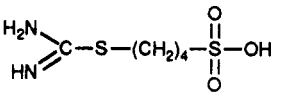
$$\sum n_i = \sum p$$

$$\sum (n_i v_i) - 2\sum n_i + 2 = 2\mathrm{IC} \tag{9}$$

and

$$\mathrm{IC} = [\sum n_i (v_i - 2) + 2]/2 \tag{10}$$

This IC equation (10) is similar to the one normally used in the organic chemistry field, for the Index of Hydrogen Deficiency (IHD), which is applied only when the elements are in their minimum valence state.[18-20] Our system, however, is able to work with elements having multiple valences into a single structure. For instance, sulfur in compound A displays valences 2 and 6.



compound A

For distinguishing between these two valence states, the symbols S and Sx have been used for the sulfurs in their different oxidation states. At the end of the generation process, the system assigns an S to the atom Sx. In this way the system copes with any atom of multiple valences. In these conditions the known IHD becomes a subset of our general system of isomer generation and representation.

When working with atom valences bigger than four, the install option of the principal menu allows for changing the maximum graph degree to the appropriate value. Otherwise, by default, this value is considered as four.

As a result of applying expression 10, a set of options for each structure appears. For instance for the molecular formula $C_5H_{12}N_2O_3S_2$, which includes the structure of compound A

$$\mathrm{IC} = \frac{5(2) + 12(-1) + 2(1) + 3(0) + 1(0) + 1(4) + 2}{2} = 3$$

An algorithm constructs a list of isomer characteristic options based on the following expression:

$$\mathrm{IC} = f1 + f2 + 2f3$$

where $f1$ is the number of double bonds, $f2$ is the number of cycles, and $f3$ is the number of triple bonds. So, in this example, the possibilities for the IC are (a) three double bonds; (b) two double bonds and one cycle; (c) one double bond and two cycles; (d) one double bond and one triple bond; (e) one triple bond and one cycle; or (f) three cycles. This is the set of basic characteristics resulting from the input molecular formula. The user can select one of the options or all of them. Also at this moment other characteristics are settled on, like the presence of one or more particular patterns in the molecular structure as explained later.

2. Tuple Notation. Nonredundant exhaustive generation and counting of acyclic molecules, including multiple bonds and heteroatoms as a part of the graph, need a particular notation to be created, and that will be explained now.

The method considers the notion of lexicographic order of the tuples that represents rooted trees.[13] A K-tuple (a_1, a_2, ..., a_K) of integers is defined lexicographically smaller than an L-tuple (b_1, b_2, ..., b_L), if there exists an index j with $1 \le j \le L$ so that $a_i = b_i$ for $1 \le i \le j$ and either $j = K + 1$ or $a_j < b_j$.

A given rooted tree with $N > 1$ vertices and M edges incident to the root vertex gives rise to M rooted subtrees by removing the root vertex and all of its edges. These rooted subtrees (taking as the root in the subtree the neighbor of the removed vertex) with L_1, L_2, ..., L_M vertices are represented by subtuples. These subtuples are sorted into the reverse lexicographic order, taking in consideration that a subtree with 1 vertex is represented by the 1 tuple (0). Concatenation of the sorted subtuples allows for getting the N-tuple representative of the rooted tree whose first component is the degree of the root vertex. Many tuples can be defined for the selected root vertex. So, after inspecting all the rooted trees (by selecting the vertices one after the other as the probable root) the lexicographically largest obtained N-tuple is assigned to the tree as its canonical representation.

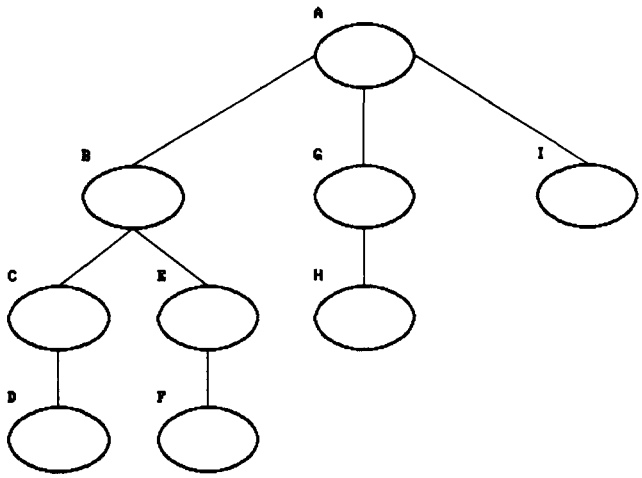


Figure 2. Rooted tree diagram and its tuple representation.

Thus, a tuple consists of an array of integer numbers that represent the number of sons a particular component (a node or a point of a tree) has (see Figure 2). Component A is the root, and it has a degree 3 or three sons. Component B has a degree 2. Components C, E, and G have a degree 1. Finally, components D, F, H, and I have a degree of 0.

Incorporation of heteroatoms and multiple bonds gives rise to trees of a different type, not described in the literature, whose nodes contain more information than before. Now, the atom type must be defined and also the type of bonds associated with it. To accomplish that requirement the following procedure was developed.

(a) For describing the atom type, each number of the tuple was accompanied with a corresponding atom symbol. The symbol was written preceding each tuple component (see Figure 3a).

(b) For defining the bonds associated to an atom, consideration of the fact that in a tree every node has only one father from which it comes allows for defining this bond between the component and its father. For that purpose, a letter s, for single, d, for double, or t, for triple, was written in the tuple immediately following the number of the tuple component (see Figure 3b). For the root component a letter r was used since a root vertex is considered to have no father.

3. Isomer Generation. When the requirements are established, the generator module begins its work.

The generation process uses the described tuple notation, and it is done basically in three steps: (i) generation of the skeleton of the acyclic structure, (ii) incorporation of heteroatoms to the structure, and (iii) incorporation of multiple bonds.

(*i*) *Generation of the Skeleton.* The first of these steps considers the generation of trees having N points with a maximum degree determined by the maximum valence of an atom, where N is the total number of skeleton atoms given by the molecular formula. Each atom has assigned a valence. There is an install subroutine that allows one to define these parameters. An algorithm generates numeric tuples for representing the trees. Unique combinations are created starting from the tuples with maximum lexicographic values. That results in an exhaustive generation of isomers and avoids unnecessary repetitions. Care had to be taken here for

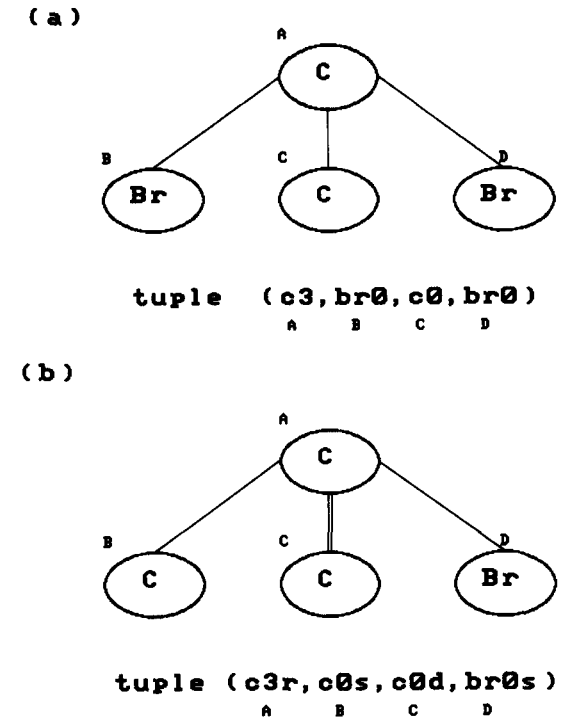


Figure 3. (a) Rooted tree containing heteroatoms and one of the corresponding tuple representations. (b) Rooted tree having both an heteroatom and a double bond, with its tuple representation.

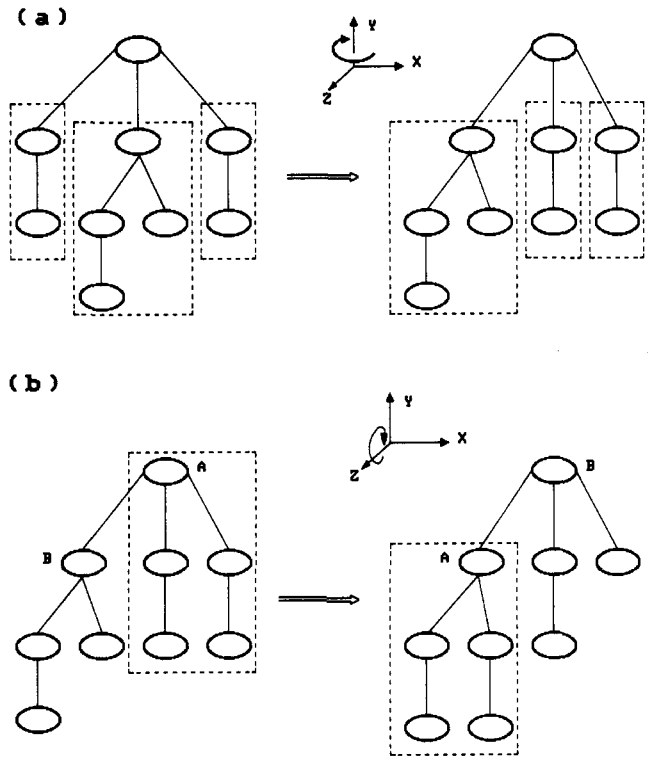


Figure 4. Representation of filtering processes carried out during the isomer generation step: (a) rotation around an imaginary Y-axis; (b) rotation around an imaginary Z-axis.

analyzing each of the combinations while it is being created, choosing the correct canonical tuples. Two tuple filtering procedures have been used.[13] Finally, the tuple that had the biggest lexicographic order value for each tree is chosen.

One of the filtering processes used is represented by the following example (see Figure 4a). The first component of the tuple (3,1,0,2,1,0,0,1,0) has three joined subgraphs or branches: the branch (1,0), the branch (2,1,0,0), and the branch (1,0). Rearrangement of these branches gives rise to

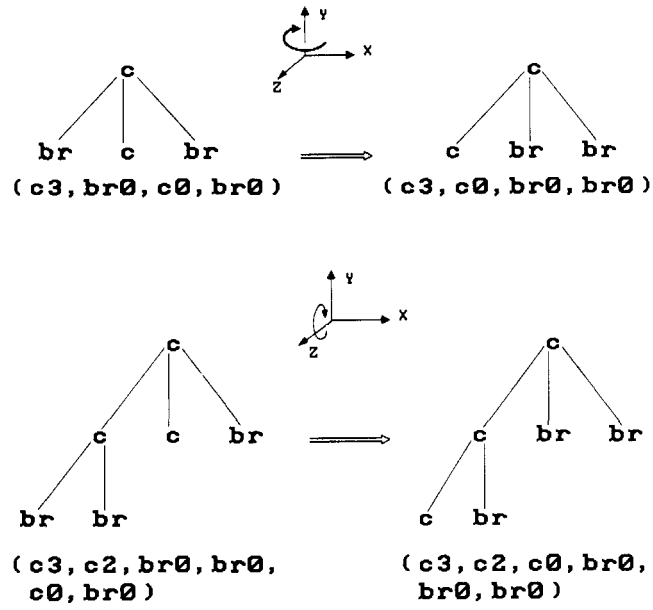


Figure 5. Canonical tuple selection by applying filtering processes according to precedence rules for atoms.

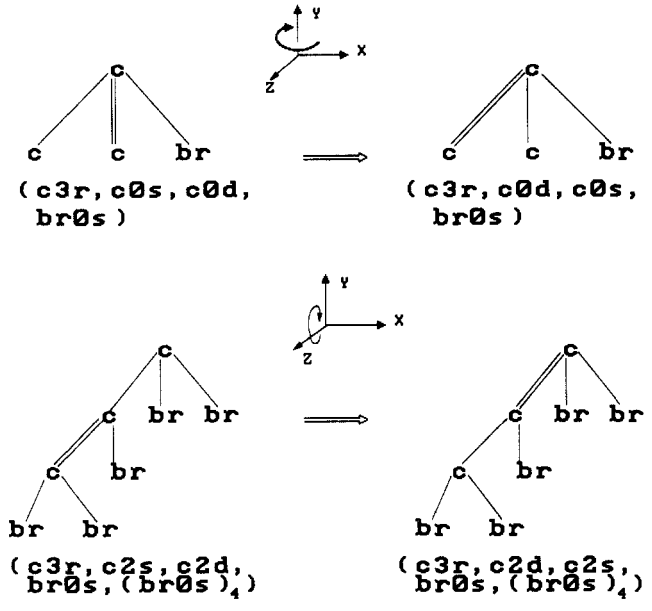


Figure 6. Canonical tuple selection by applying filtering processes according to atom and bond precedence rules.

the tuple (3,2,1,0,0,1,0,1,0). This tuple being lexicographically bigger than the first one is a better candidate to be the canonical tuple or the tree representative tuple. In this rearrangement there is no change of the root vertex. The process can be imagined like an *Y*-axis rotation where the *Y*-axis should be passing vertically through the root vertex. Initial and final situations are represented in Figure 4a. The more ramified branch remains at the left side of the tree, and it is written first in the tuple producing in this way a bigger tuple.

The second tuple filtering procedure can be conceived like an imaginary *Z*-axis rotation with the aim of getting a real change of the root vertex (see Figure 4b). In such a way, the tuple (3,2,1,0,0,1,0,1,0) can give rise to the tuple (3,2,1,0,1,0,1,0,0) which is bigger than the first one and corresponds in this case to the final canonical tuple. Here it can be seen that the initial root component A is changed to the branch (2,1,0,1,0) of the final tree and that the initial second component B represented by (2,1,0,0), having a branch (1,0) and a branch (0), has changed to the root vertex in the final tree having now three branches: (2,1,0,1,0), (1,0), and (0). In this example then, the root vertex A in the first tree has been changed by the root vertex B in the final one.

Tuples that pass satisfactorily both filtering procedures are stored as the representative ones and are counted as unique for that particular structure. In this way, each structure remains codified as a tuple. The described process allows for the generation of all the irredundant connectivity isomer structures.

(*ii*) *Heteroatom Incorporation.* An algorithm was created for making all the possible permutations considering all the heteroatoms to be included, taking care of the particular valence of the atoms. Then tuple filtering processes are done until no modification of the son number of each component is obtained. As a result of this process, different tuples that have the same numerical order can be obtained.

Then a conventional precedence rule is defined for ordering the atoms and for finally getting the bigger reverse lexicographic ordered tuple. In this case, the precedence rule is

$$c > n > o > f > s > cl > br > i > ...$$

An example of a canonical tuple selection is presented in Figure 5. The tuple (c3,br0,c0,br0) when rotated in relation with an imaginary *Y*-axis generates the tuple (c3,c0,br0,br0). On the other hand the tuple (c3,c2,br0,br0,c0,br0) when rotated in relation with an imaginary *Z*-axis generates the tuple (c3,c2,c0,br0,br0,br0). As can be seen, rotation does not change the numerical order (3,0,0,0 in the first case and 3,2,0,0,0,0 in the second one) but changes the citation order of the tuple components (for instance 'c,br,c,br' to 'c,c,br,br'). In both cases canonical tuples are obviously the resulting ones after rotation according to the defined precedence rule.

(*iii*) *Multiple Bond Incorporation.* The same general procedure is followed: first of all, an algorithm generates all the possible permutations allowed for the tuple incorporation of the double or triple bonds, taking care of the atomic valencies. Then, filtering processes are done for choosing the canonical tuple. In this case the following conventional rule of precedence is used:

$$r > t > d > s$$

For instance (see Figure 6), the tuple (c3r,c0s,c0d,br0s) when rotated over an imaginary *Y*-axis gives the tuple (c3r,c0d,c0s,br0s). This last tuple agrees with both precedence rules and also it corresponds with the reverse lexicographic order, so it is denoted as the tuple representative of the tree. In addition, in that figure is represented the tuple (c3r,c2s,c2d,br0s,br0s,br0s,br0s,br0s) whose corresponding tree when rotated over an imaginary *Z*-axis produces the tuple (c3r,c2d,c2s,br0s,br0s,br0s,br0s,br0s), which is the canonical one.

In Figure 7 a general block diagram for the generation procedure of acyclic structures that can contain heteroatoms and multiple bonds is presented, and in Appendix A a detailed example of the whole process is given.

4. Pattern Use. Isomer generation can be further restricted according to one or more patterns. In this case the procedure is similar to the one explained before: (i) skeleton generation, (ii) heteroatoms incorporation, and (iii) multiple bonds incorporation.

Filtering of structures or tuples is done over the generated carbon skeletons while the representative tuple is still in memory. For that, once the canonical tuple is found, a sequential matching with the patterns allows for the correct tuple selection. Simple patterns were found in a great number of skeletons. By contrast, a more complicated pattern allowed for the selection of a much smaller number of skeletons, and

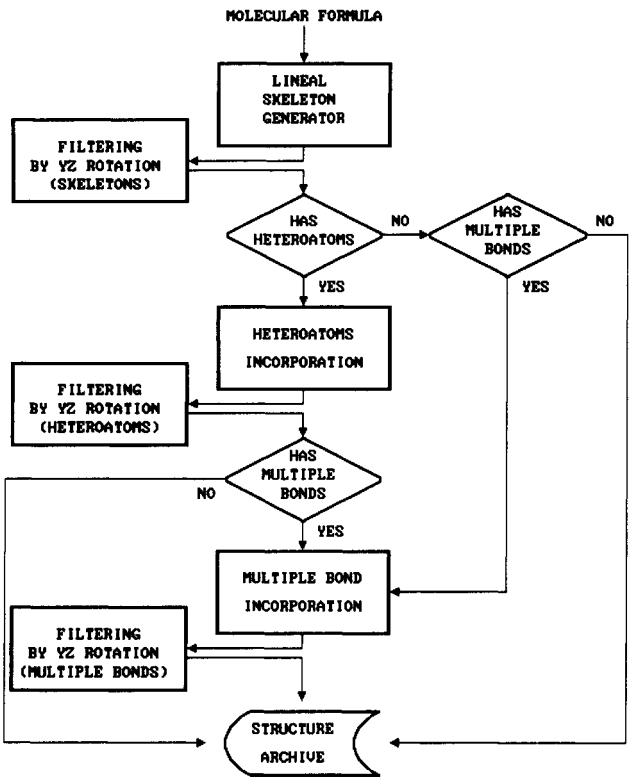


Figure 7. Simplified diagram representing the principal steps of isomer generation. Selection pattern, not represented in this diagram, is done for each step after filtering processes as many times as number of patterns are used.

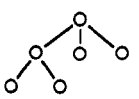
Table I. Generation of $C_4H_6N_2S$ Isomers Having One Double Bond and One Triple Bond with and without Pattern Restrictions

	no of generated structures			CPU
	step 1	step 2	step 3	time,[a] (seg)
case 1	9	333	651	4.93
case 2	9	333	651	5.27
case 3	9	314	520	4.85
case 4	2	4	2	0.83

[a] CPU time determined in a Workstation AVIION 2000 of Data General. Case 1 is without pattern. From case 2 to case 4 the complexity (selectivity) of the pattern is increasing (see text).

so this fact facilitates the next two additional processes, (ii) and (iii). As an example, in Table I the number of generated structures at each step for patterns of different complexity and also the CPU time requirements for the molecular formula $C_4H_6N_2S$ having as molecular characteristics one double bond and one triple bond can be observed. The used independent patterns were

case 1: without any pattern
case 2: with a methyl group (c0r)
case 3: with an ethyl group (c1r,c0s)
case 4: with a group (c3r,n2s,c0s,s0s,c0d,c0s) having a skeleton like



As it can be seen from Table I, only two skeletons match finally with the above pattern (case 4), so incorporation of heteroatoms is done over these two structures which results in four structures. Incorporation of multiple bonds then is done over these four structures. Consequently, the CPU time in this case is lower than in the other cases. It is important to realize here that the whole generation process is done having a single tuple in memory. If the tuple is found to be nonrepresentative or if it does not match with every chosen pattern, it is discarded and a new tuple is analyzed. Tuples that match the user's requirements are stored under N-tuple code.

RESULTS AND DISCUSSION

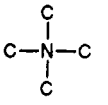
Previous work by Knop et al.[13] provided a method that considered a maximal vertex degree of four, and it was developed for saturated compounds only. On the basis of graph theory, each vertex always represented a C atom and each edge a single bond. Later the method was applied to the generation of compounds having only one double bond or one triple bond in the structure,[4] but there was no detailed explanation about that. Other authors have considered multiple bonds as specially labeled edges[21] or as one or two cycles between two atoms, respectively.[22]

The system developed here allows for the generation of acyclic isomers with heteroatoms within the skeleton, with one or more multiple bonds, and with atoms having multiple valences in an exhaustive and irredundant way. It is based on the fact that the use of tuples provides a very efficient isomer generation method.[4]

The class of isomers generated by our program includes all the connectivity isomers. Stereoisomerism is not considered.

Heteroatoms in a molecule can have different valences. However, their incorporation is not a trivial task. For instance the same molecule can contain a sulfide function and a sulfonic acid function such as exemplified before with compound A, and the degree of the corresponding nodes could be bigger than four. Normally, in the specific literature concerning this subject, heteroatoms are considered to have only one valence or they are treated as substituents, not as part of the skeleton atoms.[4,13,22,23] In addition, these programs do not have the capacity of working with degrees bigger than four.

CAMGEC can generate salt isomers also. For instance, there are 229 acyclic isomers having the formula $C_8H_{20}N^+Cl^-$ corresponding to tetraethylammonium chloride. For that, the program uses a N atom with a valence of 4. This atom is defined as Nx by running an install program specifically created with the aim of defining different kinds of atoms. There are 18 of these isomers that have the following pattern:



On the other hand, organic chemistry literature[18-20] has used until now an equation for establishing the number of multiple bonds or cycles a molecule can have (IHD), which considers a determined number of heteroatoms but only in their minimum valence and without offering any demonstration of such an expression. Recently it has been recognized that such a type of equation is only heuristically used.[23] That equation would be of no service for the previous described molecule (compound A) having sulfur atoms with valences of 2 and 6. In this work, a general equation that calculates isomer characteristics from the general molecular formula was deduced on the base of graph theory,[16,17] and its use was successfully checked. In this way, use of the known IHD expression becomes a subset of the applications of our general IC equation.

Generation of isomers is done selectively. For that, one or more patterns are input at the beginning of the generation process in such a way that the program can eliminate a

Table II. Generation of Acyclic Saturated Isomers

formula	n = 4	5	6	7	8	9	10
C_nH_{2n+2}							
a	2	3	5	9	18	35	75
b	x	x	5	9	18	35	75
this work	2	3	5	9	18	35	75
$C_nH_{2n+1}X$							
a	4	8	17	39	89	211	507
b	4	8	17	39	89	211	507
this work	4	8	17	39	89	211	507
$C_nH_{2n+2}O$							
c	7	14	nc	nc	nc	nc	nc
b	7	14	32	72	171	405	989
this work	7	14	32	72	171	405	989
$C_nH_{2n+3}N$							
c	9	17	nc	nc	nc	nc	nc
b	8	17	39	89	211	507	1238
this work	8	17	39	89	211	507	1238
$C_nH_{2n}X_2$							
a	9	21	52	129	332	859	2261
b	9	21	52	129	332	859	2261
this work	9	21	52	129	332	859	2261
$C_nH_{2n}XY$							
a	12	31	80	210	555	1479	3959
b	12	31	80	nc	nc	nc	nc
this work	12	31	80	210	555	1479	3959
$C_nH_{2n+2}O_2$							
c	28	nc	nc	nc	nc	nc	nc
b	28	69	179	nc	nc	nc	nc
this work	28	69	179	463	1225	3246	8697
$C_nH_{2n+4}N_2$							
c	36	nc	nc	nc	nc	nc	nc
b	38	97	260	nc	nc	nc	nc
this work	38	97	260	686	1857	4994	13550

[a] Ref 24. [b] Ref 25. [c] Ref 22; X, Y: F, Cl, Br, I. x, the method used in ref 25 cannot allow their calculation; nc, not calculated.

Table III. Influence of Double Bonds on Number of Acyclic Isomers[a]

formula	n = 5	7	10	db Nr
C_nH_{2n+2}	3	9	75	0
C_nH_{2n}	5	27	377	1
C_nH_{2n-3}	6	44	901	2
$C_nH_{2n+1}X$	8	39	507	0
$C_nH_{2n-1}X$	21	149	2876	1
$C_nH_{2n-3}X$	20	228	6932	2
$C_nH_{2n+2}O$	14	72	989	0
$C_nH_{2n}O$	41	294	5779	1
$C_nH_{2n-2}O$	44	485	14594	2
$C_nH_{2n+3}N$	17	89	1238	0
$C_nH_{2n+1}N$	56	398	7769	1
$C_nH_{2n-1}N$	69	725	21023	2
$C_nH_{2n}X_2$	21	129	2261	0
$C_nH_{2n-2}X_2$	54	489	12778	1
$C_nH_{2n-4}X_2$	48	721	30101	2
$C_nH_{2n}XY$	31	210	3959	0
$C_nH_{2n-2}XY$	87	840	22877	1
$C_nH_{2n-4}XY$	76	1242	54382	2
$C_nH_{2n+2}O_2$	69	463	8697	0
$C_nH_{2n}O_2$	207	1953	52593	1
$C_nH_{2n-2}O_2$	237	3345	135860	2
$C_nH_{2n+4}N_2$	97	686	13550	0
$C_nH_{2n+2}N_2$	378	3520	93912	1
$C_nH_{2n}N_2$	564	7340	279288	2
$C_nH_{2n-2}N_2$	369	7748	461366	3

[a] X, Y: F, Cl, Br, I.

Table IV. Selective Generation of Acyclic Isomers Having Three Double Bonds and Heteroatoms with Multiple Valences: An S Atom with Valence of 2 and One with Valence of 6[a]

formula	no. of acyclic isomers: without pattern	pattern a	pattern b	patterns a and b
$C_2H_6N_2O_3S_2$	109 008	576	294	6
$C_3H_8N_2O_3S_2$	1 544 734	7 628	1 978	29
$C_4H_{10}N_2O_3S_2$	16 406 550	70 232	10 604	102
$C_5H_{12}N_2O_3S_2$		508 814	49 764	321

[a] Pattern a: NH_2—C—(=NH)S–. Pattern b: —SO_3H.

significant number of isomers according to user requirements. As can be seen from Table I, the process is more effective when the pattern(s) to be found is(are) more selective. This facility is particularly useful when different functional groups are required in the molecules. It is evident that the use of a nondiscriminating pattern increases the time of the generation process. In case 4 of Table I only two skeleton structures were selected for incorporation of heteroatoms, and as a result only four structures were considered for multiple-bond incorporation. These numbers in comparison with those of case 3 show a significant reduction of processing time.

The obtained results of this work reproduce known values from the literature as can be seen from Table II, where some new values determined here are also included. In effect, there is good agreement between the reported number of isomers and the corresponding values produced in this work. That could be an indication of the reliability of our results. From these data, it is observed that the number of isomers increases with the number of skeleton atoms, as expected. Also, the number of isomers and the amount of their variation increase with the number of heteroatoms, their valences, and the diversity of them. In Table II, data have been ordered according to the sequence X, O, N, X_2, XY, O_2, and N_2 for showing that tendency better. In addition, from data in Table III it can be observed that the number of isomers increases with the number of double bonds. This increase is more significant for molecules with a bigger number of carbon atoms. For molecules with five C atoms the same tendency is observed with the exception of isomers C_5H_7X and $C_5H_6X_2$ with 2 double bonds and isomers $C_5H_8N_2$ with 3 double bonds.

In Table IV, the number of acyclic isomers having three double bonds in their structures and seven heteroatoms from which one S atom with valence of 2 and one S atom with valence of 6 are tabulated. Generation of isomers was done in four ways: without any pattern, with the pattern a [NH_2—C—(=NH)S—], with the pattern b (–SO_3H), and with both patterns a and b. Results show again that the number of isomers increase with the number of atoms and that the use of patterns produces a more selective generation of isomers.

On the other hand, interaction with the system is very friendly: it uses menus, and it does not require a graphic terminal. However when a difficult pattern is required, it is more convenient to use a graphic interface for input purposes. A codifier module transforms the drawn pattern into the *N*-tuple notation used by the program.

The output is chosen by the user: it could be just the final total number of structures or an archive containing the generated isomer structures in a codified way. Each of the isomers is stored in just a single line of the archive.

After automatically decoding the output information, the structures of the isomers can be displayed with the help of the graphic interface. Other useful modules can be accessed from there, and, for example, the topological indexes of each isomer can also be automatically calculated.

The graphic interface module that interacts with the generator system is in charge of generating atom coordinates, considering for that standard bond lengths and bond angles. These facilities allow for the system to be in good coordination with other CAMD modules, such as the ones related to semiempirical calculations, molecular volume calculation, statistical correlations, QSAR studies, and ab initio methods.

Memory use is optimized since during the process only one structure is kept in dynamic memory. In addition, the algorithm produces only structures not generated before and makes an internal valence test for generating exclusively isomers that have chemical sense. This is an important point and a problem for other systems, as was recently recognized.[6] There is no limit for the number of present atoms in the isomer structure, except for storage capacity and time of generation.

Another useful contribution is the compact tuple code proposed here for molecules containing heteroatoms and multiple bonds. The code is compact and unique for the molecules that make it an important tool for molecular databases[26,27] and for other CAMD modules.[28,29]

Many other advantages are worth mentioning, such as the system portability. The system is written in C under UNIX, and it runs on accessible workstations or minicomputers without a graphic interface. The source program, CAMGEC, occupies 169.6 KB, and its compiled version occupies 204.8 KB. It is an autonomous system and does not need to be connected to other subroutines to accomplish its work. However for facilitating the use of the results obtained with CAMGEC, in our laboratory we have implemented a connection to a graphic interface, developed in our group. This interface is written in Regis, and as it was mentioned before, it considers standard values for the bonds and angles and provides connections to other CAMD modules. In this way the whole system behaves as a powerful tool for research.

Indeed, the whole system has proved to be very useful not only in research[28] but also in education,[30] particularly in the field of computational chemistry.

In conclusion the system reported here offers selective, exhaustive, and irredundant generation and counting algorithms for acyclic isomers that can have heteroatoms in the skeleton and/or one or more multiple bonds. Also, it offers a compact and topologically unique code for these molecules. The resulting structures can be displayed and submitted to other calculation modules, making the whole integrated system of great utility in structure elucidation, in organic synthesis, and especially in molecular design. All of the mentioned features make this system more general and comprehensive than the ones already in the literature.

ACKNOWLEDGMENT

We gratefully acknowledge the University of Santiago for financial support. Also, we gratefully appreciate Prof. J. B. Hendrickson and Prof. A. T. Balaban for the encouraging comments and useful suggestions to this work.

APPENDIX A

An example showing the principal sequential steps followed in the isomer generation program, CAMGEC, is given.

First, the molecular formula, C_3H_4NBr, is input to the system interactively. Then the system calculates the IC which is equal to 2 in this case and proposes the following options: (a) two double bonds, (b) one double bond and one cycle, (c) one triple bond, or (d) two cycles. The chosen option is (c); one triple bond. No pattern is required.

This finishes the interaction with the system which starts then to work alone.

1. Skeleton Generation.

(*a*) CAMGEC determines from the formula the occurrence of five heavy atoms (different from H atoms) and the maximum valence atom.

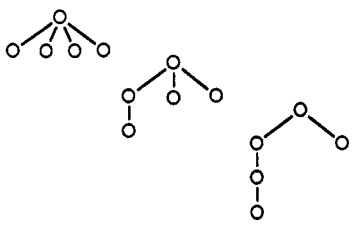
(*b*) An algorithm generates all the possible irredundant combinations. In this way and according to graph theory,[16,17] tuples of five components are generated in such a way that the sum of all the component degree of the tuple is 5 − 1 = 4.

(*c*) Filtering processes allow for canonical tuple selection. As a result of this step the following skeletons or their representative tuples (designed here like g1–g3) are created:

g1: c4r c0s c0s c0s c0s

g2: c3r c1s c0s c0s c0s

g3: c2r c1s c1s c0s c0s



As it is observed, all the atoms are written as C atoms (the atom of maximum valence in the molecular formula), and all the bonds are considered as single bonds in this step.

2. Heteroatom Incorporation.

(*a*) An algorithm makes all the possible combinations of the different atoms according to the valence of them over each skeleton.

(*b*) Filtration processes select the following 12 tuples (h1–h12):

h1: c4r c0s c0s n0s br0s

h2: n3r c1s br0s c0s c0s

h3: n3r c1s c0s c0s br0s

h4: c3r n1s br0s c0s c0s

h5: c3r n1s c0s c0s br0s

h6: c3r c1s n0s c0s br0s

h7: c3r c1s c0s n0s br0s

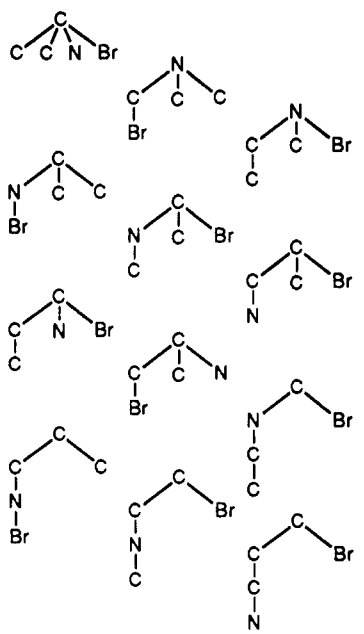
h8: c3r c1s br0s c0s n0s

h9: c2r n1s c1s c0s br0s

h10: c2r c1s n1s br0s c0s

h11: c2r c1s n1s c0s br0s

h12: c2r c1s c1s n0s br0s



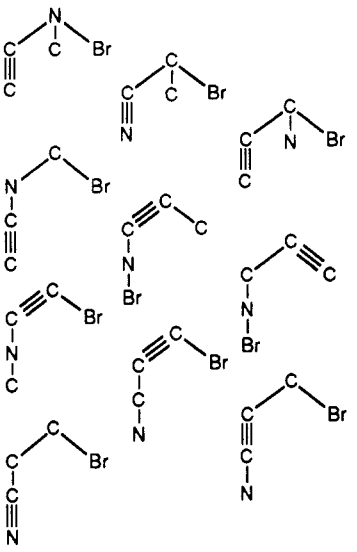
It is easy to observe that skeleton g1 generates tuple h1; g2 generates tuples h2–h8; and g3 generates tuples h9–h12.

3. Multiple Bond Incorporation.

(*a*) An algorithm makes all the possible combinations trying to incorporate the triple bond chosen in this case, in each of the tuple defined as h1–h12. The free or residual valence of each atom is considered.

(*b*) Filtration processes select the final canonical tuples and the following is the resulting list:

1: n3r c1s c0t c0s br0s
2: c3r c1s n0t c0s br0s
3: c3r c1s c0t n0s br0s
4: c2r n1s c1s c0t br0s
5: c2r c1t n1s br0s c0s
6: c2r c1s n1s br0s c0t
7: c2r c1t n1s c0s br0s
8: c2r c1t c1s n0s br0s
9: c2r c1s c1t n0s br0s
10: c2r c1s c1s n0t br0s



Isomers 1, 2, 3, 4, and 7 were generated from h3, h6, h7, h9, and h11, respectively. In the same way, isomers 5 and 6 are generated from h10, and finally isomers 8, 9, and 10 are generated from h12. As it can be seen h1, h2, h4, h5, and h8 were not able to incorporate a triple bond, and therefore they were automatically discarded by the system.

From this example it remains clear the relative precedence for heteroatoms and multiple bonds in the same molecule is

heteroatoms > multiple bonds

REFERENCES AND NOTES

(1) Polya, G.; Read, R. C. *Combinatorial Enumeration of Graphs, Groups, and Chemical Compounds*; Springer-Verlag: New York, 1987; pp 58–74.

(2) Balaban, A. T. Chemical Graphs. XXII. Valence Isomers of Heteroannulenes or of Substituted Annulenes. Coisomeric Cubic Multigraphs. *Rev. Roum. Chim.* **1974**, *19*, 1323–1342.

(3) Balaban, A. T.; Banciu, M.; Ciorba, V. *Annulenes: Benzo-, Hetero-, Homo-Derivatives, and Their Valence Isomers*; CRC: Boca Raton, FL, 1987; Vol. 3.

(4) Trinajstić, N.; Jericevic, Z.; Knop, J. V.; Muller, W. R.; Szymanski, K. Computer Generation of Isomeric Structures. *Pure Appl. Chem.* **1983**, *55*, 379–390.

(5) Lindsay, R. K.; Buchanan, B. G.; Feigenbaum, E. A.; Lederberg, J. *Application of Artificial Intelligence for Organic Chemistry*; McGraw-Hill: New York, 1980.

(6) 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.

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

(8) Kvasnicka, V.; Pospichal, J. Canonical Indexing and Constructive Enumeration of Molecular Graphs. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 99–105.

(9) Read, R. C.; Cameron, R. D.; Colbourn, C. J.; Wormald, N. C. Cataloguing the Graphs on 10 Vertices. *J. Graph Theory* **1985**, *9*, 551–562.

(10) Akutsu, T. A New Method of Computer Representation of Stereochemistry. Transforming a Stereochemical Structure into a Graph. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, (3), 414.

(11) Rucker, G.; Rucker, C. Isocodal and Isospectral Points, Edges, and Pairs in Graphs and How to Cope with Them in Computerized Symmetry Recognition. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, (3), 422.

(12) Contreras, M. L.; Valdivia, R.; Rozas, R. Exhaustive Generation of Organic Isomers. Part 2. Cyclic Structures. New Compact Molecular Code. Manuscript in preparation.

(13) Knop, J. V.; Muller, W. R.; Jericevic, Z.; Trinajstić, N. Computer Enumeration and Generation of Trees and Rooted Trees. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 91–99.

(14) Hendrickson, J. B.; Parks, C. A. Generation and Enumeration of Carbon Skeletons. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 101–107.

(15) Fruhbeis, H.; Klein, R.; Wallmeier, H. Computer-Assisted Molecular Design (CAMD)–An Overview. *Angew. Chem. Int., Ed. Engl.* **1987**, *26*, 403–418.

(16) Harary, F. *Graph Theory*; Addison-Wesley: London, 1972; pp 3–40.

(17) Toranzos, F. A. *Introduccion a la Teoria de Grafos*; Monograph No. *15*, Series Mathematics; OEA: Washington, 1976.

(18) McLafferty, F. W. *Interpretation of mass spectra*; W. A. Benjamin Inc.: New York, 1966; p 25.

(19) Gutsche, C. D.; Pasto, D. J. *Fundamentals of Organic Chemistry*; Prentice-Hall Inc.: Englewood Cliffs, NJ, 1975; 18.3, p 471.

(20) Pine, S. H.; Hendrickson, J. B.; Cram, D. J.; Hammond, G. S. *Organic Chemistry*; McGraw-Hill: New York, 1980; p 13.

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

(22) Masinter, L. M.; Sridharan, N. S.; Lederberg, J.; 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.

(23) Luinge, H. J.; Van der Maas, J. H. AEGIS, an Algorithm for the Exhaustive Generation of Irredundant Structures. *Chemom. Intell. Lab. Syst.* **1990**, *8*, 157–165.

(24) Read, R. C. *Chemical Applications of Graph Theory*; Balaban, A. T., Ed.; Academic Press: New York, 1976; pp 11–60.

(25) Sasaki, S.; et al. A Computer Program for Generation of Constitutionally Isomeric Structural Formulas. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 220–229.

(26) Contreras, M. L.; Deliz, M.; Rozas, R. Personal microcomputer based system of chemical information with topological structure data elaboration. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 163–167.

(27) Contreras, M. L.; Allendes, C.; Alvarez, L. T.; Rozas, R. Perception and Recognition of Digitized Molecular Structures. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 302–307.

(28) (a) Contreras, M. L.; Rozas, R. Predictive Chemistry in CAMD. *Proc. Struct. Mol. Spectr. Symp.*, Santiago, Chile, Sep 1991. (b) Contreras, M. L.; Avila, M.; Cabezas, A.; Rojas, M. T.; Rozas, R. Modelling of the algae pheromone receptor. *Proc. XIX Chem. Latinoam. Congr.*, Buenos Aires, Argentina, Nov 1990.

(29) Rozas, R.; Morales, J.; Vega, D. Artificial Smell Detection for Robotic Navigation. *IEEE Proc. ICAR* **1991**, 1730–1733.

(30) A computational chemistry course has been lectured at the Department of Chemistry of the University of Santiago where isomer generation has been used.