Determination of Topological Equivalence Classes of Atoms and Bonds in C_{20}−C_{60} Fullerenes Using a New Prolog Coding Program

Thierry Laidboeur,[†] Daniel Cabrol-Bass,*,[†] and Ovidiu Ivanciuc[‡]

LARTIC University of Nice-Sophia Antipolis, Parc Valrose 06108 Nice Cedex, France, and Department of Organic Chemistry, Faculty of Chemical Technology, University "Politehnica" of Bucharest, Splaiul Independentei 313, 77206 Bucharest, Romania

A new general molecular coding Prolog program is used in theoretical studies of the properties of fullerene isomers and derivatives to identify the topological equivalence classes of atoms and bonds in C_{20}−C_{60} fullerenes. The symmetry perception algorithm is based on an exhaustive search of a minimal code. During this search all canonical labeled mapping trees are generated, allowing construction of equivalence classes of atoms and bonds. Although the present algorithm makes a complete search of canonical labelings in the molecular graph, computational time is maintained within reasonable limits even for large regular graphs such as fullerenes. In the case of fullerenes, topological equivalence classes of atoms allow an easy generation of heterofullerene isomers, while the bond classes allow for the generation of addition or cycloaddition isomers. Also, the number of atom classes and the number of atoms in each class gives important information on the number of ¹³C NMR signals and intensity patterns in fullerenes.

INTRODUCTION

Fullerenes have been the subject of intensive studies since the discovery of the C_{60} fullerene.[1] Synthesis of other fullerenes and fullerene derivatives has led to an explosion of both theoretical and experimental research on the properties of this three-dimensional aromatic system, but practical applications of fullerenes have yet to be developed.[2] Fullerenes are cage structures with five- and six-membered rings formed by trigonal sp² carbon atoms. In fullerene chemistry, addition and cycloaddition reactions represent a research area that generates important information concerning the reactivity and regioselectivity of fullerenes and provides ways to obtain new materials. Extensive experimental research has been devoted to the study of hydrogenation and halogenation of fullerenes,[3−5] but structural characterization of the addition products is a difficult task due to the presence of a large number of isomers which theoretically can be generated. For example, C_{60} has two types of bonds (i.e., between two six-membered rings and between a five- and a six-membered ring), which can give two addition isomers C_{60}H_2 with different thermodynamical stabilities. On increasing the number of hydrogen molecules, the number of possible isomers becomes very great, and there are no simple reactivity rules to predict the most stable one.

As pointed out above, the usual analysis methods are of little use in determining the structure of polyaddition compounds, and one has to rely on theoretical calculations. Knowing the molecular formula of the addition product from the mass spectrum, all possible isomers are generated, and their thermodynamic stability is estimated by molecular or quantum mechanical methods;[6,7] the most stable structure is assigned to the major addition product. The main difficulty is generating all isomers, and in certain instances only a subset of symmetric isomers is taken into consideration,[8,9]

because an exhaustive generation of all addition isomers is a difficult task without the aid of an efficient computer program.

Despite numerous efforts, the generation of heterofullerenes has not yet been successful, but their structures have been investigated by various theoretical methods.[10,11] Their synthesis will generate new data concerning three-dimensional heteroaromaticity and reactivity as well as new possibilities of obtaining fullerene derivatives. The theoretical study of heterofullerenes requires an exhaustive generation of all isomers with a given molecular formula, because partial studies on selected symmetric isomers can lead to misleading conclusions regarding their stability and reactivity.

In the present paper we present a general, efficient coding algorithm applied to the identification of the topological symmetry of fullerenes. The algorithm was implemented in Prolog and uses a two stage procedure: (a) the atoms of the molecular graph are partitioned in classes according to a topological invariant, and (b) an exhaustive generation of all canonical labeled mapping trees (CLMTs) and ring closure lists is completed, which results in the canonical code. The determination of atom and bond equivalence classes is made by matching all the CLMTs. The efficiency of the algorithm in identifying the topological symmetry of molecular graphs will be tested for a set of 20 C_{20}−C_{60} fullerenes. Determination of the topological symmetry of fullerenes has both a practical and theoretical importance, because identification of the topological equivalence classes of atoms allow easy generation of heterofullerenes isomers, while the bond classes are necessary for the generation of addition or cycloaddition isomers. Also, the number of atom classes and the number of atoms in each class gives important information on the number of ¹³C NMR signals and intensity patterns in fullerenes. The same information can be used for the generation of ESR hyperfine patterns in fullerene radicals. Although the present algorithm makes a complete search of the labelings in the molecular graph, computational

---

[†] LARTIC University of Nice-Sophia Antipolis.
[‡] University "Politehnica" of Bucharest.

time is maintained within reasonable limits even for large regular graphs such as fullerenes.

VERTEX AND EDGE PARTITIONING OF GRAPHS

Isomorphic mapping of the vertices of a graph G onto themselves, preserving the adjacency relationship, is called automorphism of a graph G. The set of all automorphisms of a graph G forms a group which is called the automorphism group of G and offers information on the topological symmetry of the graph.[12,13] As pointed out above, identification of the topological symmetry of fullerenes and fullerene derivatives has both a theoretical and practical importance, but many published algorithms for vertex partitioning fail to distinguish between different topological classes of atoms.[14,15] The graph-vertex-automorphism partitioning of molecular graphs has received considerable attention in recent years.[16−28] Vertex partitioning is of fundamental interest and has practical applications in (a) computer perception of the topological symmetry of a molecule, (b) determination of maximum common substructure, (c) exhaustive and irredundant generation of candidate structures in structure elucidation studies, and (d) computer generation of ^{13}C NMR signals and their intensity patterns. For these reasons, a vertex-automorphism partitioning module is an important part of artificial intelligence systems for synthesis design and computer-assisted structure elucidation and chemical database management and searching. The related problem of the determination of bond equivalence classes has not been well investigated,[29] and it is sometimes erroneously assumed that bonds linking atoms belonging to the same equivalent classes are equivalent.

A large number of methods have been proposed for the partitioning of atoms into topological equivalence classes, but a significant fraction of them offer an incomplete solution to the symmetry perception problem, and they fail if the investigated molecular structure is highly symmetric or contains isospectral vertices. A general characteristic of these methods is the use of graph invariants computed by an iterative process and describing the structural perception of the molecular graph at a particular atomic level. The basic assumption of these methods is that graph vertices with identical values of the used invariant belong to the same equivalence class, a fact which is not always true. Even if such algorithms provide good results for particular graphs, their use in a chemical database, in structure generation, structure elucidation, or synthesis design systems is unreliable because in some cases it fails to recognize the right atom partitioning. The only known reliable procedure for molecular graph symmetry perception can be efficiently developed in a two step algorithm: an initial partitioning using a highly discriminant topological invariant, followed by an exhaustive generation of all labelings, taking into account the partitioning made in the first step.[18,20] Obviously, atoms with different values do belong to different equivalent classes.

An overlooked fact concerning the Morgan algorithm must be noted.[16] After a preliminary assignment of topological classes based on extended connectivity, the second phase comprises an exhaustive generation of all labelings, ensuring generation of a minimal code. Thus, the Morgan algorithm always generates good minimal code in the sense defined in the original paper, despite the known shortcomings of the initial phase of generation of extended connectivity. Also, the algorithm uses several methods to reduce the number of

labelings generated, which ensures that in a reasonable amount of time one can obtain the canonical code of a large molecule like C_{60}, which, with even larger fullerenes, represents, from a computational point of view, an upper limit, since all others molecule, with fewer carbon atoms, containing heteroatoms and/or more than one single type of chemical bond, will require significantly less computational time.

PRINCIPLE

The method used in this paper is a byproduct of the canonical numbering procedure applied to a chemical database management program developed in Prolog for our laboratory. The method consists of three steps: (a) the atoms are partitioned into temporary classes using topology and chromatism local information, (b) a canonical numbering of the atoms is achieved by constructing and storing all equivalent canonical labeled mapping trees (CLMT) of the molecular graph recorded, and (c) the resulting CLMTs are matched to identify classes of equivalent atoms on the one hand and classes of equivalent bonds on the other. The first step of the algorithm consists of a preliminary topological partitioning of atoms based on their extended connectivity (EC) calculated iteratively. Unlike the original Morgan method, atoms which are distinguishable at one level are maintained in different classes in the next levels of the iterative procedure. For this purpose, atoms are classified on the basis of the list of successive extended connectivity values (LSEC), which is initialized with the atom's degree. At each iteration the LSEC of each atom is expanded by appending the extended connectivity value calculated by summing the EC of its neighbor atoms. Atoms with identical LSECs are assigned to the same class. The use of this list instead of sole extended connectivity confers the hierarchical character of the procedure and avoids oscillations in the number of classes. The procedure is repeated until the number of classes of different LSECs remains unchanged for two successive iterations. Next, atom and bond characteristics are taken into account to split the obtained classes whenever possible. The following local characteristics are hierarchically taken into account: (a) atom type, (b) number of attached hydrogen atoms, (c) formal atom charge, and (d) the ordered list of bond types originating at this atom. The resulting classes are ranked, giving priority to the LSEC lexicographic order over the local characteristics.

The procedure will be exemplified on simple classical graphs (see Figure 1). Graph 1 illustrates the oscillatory behavior of the Morgan methods. Application of this method in its original form and with the Wipke modification[17] gives the successive values of extended connectivity (M: Morgan method, W: Wipke modification) reported in Table 1.

As shown in Table 1, atoms 1 and (2,6) which have different initial connectivity (respectively 2 and 3) share the same Morgan extended connectivity value for odd iteration numbers, while atoms (3,5) and 4, which have different Morgan extended connectivity values for odd iteration numbers, share the same value for even iteration numbers. It follows from this oscillatory behavior that complete partitioning of topologically different atoms in distinct classes is not possible. In this particular case the Wipke modification, which consists of keeping connectivity of terminal atoms equal to 1 during the iterative procedure, breaks the oscillations after two iterations, and a correct partitioning is
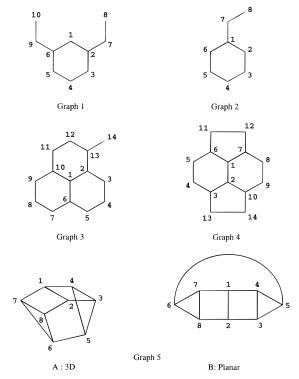
Figure 1. Some graphs of interest.

Table 1. Successive Values of Extended Connectivity for Graph 1[a]

	iteration number									
	0		1		2		3		4	
no.	M	W	M	W	M	W	M	W	M	W
1	2	2	6	6	12	12	30	30	60	58
2	3	3	6	6	15	15	30	29	74	71
3	2	2	5	5	10	10	25	25	50	49
4	2	2	4	4	10	10	20	20	50	50
5	2	2	5	5	10	10	25	25	50	49
6	3	3	6	6	15	15	30	29	74	71
7	2	2	4	4	8	7	19	16	38	30
8	1	1	2	1	4	1	8	1	19	1
9	2	2	4	4	8	7	19	16	38	30
10	1	1	2	1	4	1	8	1	19	1
NCE	3	3	4	4	5	5	5	6	5	6
NCL	3	3	5	5	6	6	6	6	6	6

[a] M: extended connectivity following Morgan, W: Wipke modification, NCE: number of classes of extended connectivity, NCL: number of classes of LSEC.

obtained after three iterations. Oscillatory behavior is prevented from the beginning by using the LSEC instead of the sole EC to constitute the atom classes. Correct partitioning is achieved at iteration number 2. Another difficulty arises from the existence of isospectral vertices in graphs. Such vertices may be topologically different but keep the same extended connectivity values throughout the iterative procedure. For example, atoms 4 and 7 in graph 2 are isospectral. Obviously, the LSECs for isospectral atoms are identical because their ECs remain the same at each iteration. The result is incomplete atom partitioning as shown in Table 2A, in which the atoms are ranked according to their LSEC class priority.

Because atoms 4 and 7 are isospectral, they are gathered into the same class although they are not topologically equivalent. In this particular example, the two isospectral vertices are at different topological distances from one terminal atom; thus the use of the Wipke modification will differentiate these atoms after a number of iterations equal

Table 2. Atom Partitioning Using the List of Successive Extended Connectivity (LSEC) for Graph 2

A: Morgan			B: Wipke		
priority	LSEC	atom no.	priority	LSEC	atom no.
1	[3,6,14,28]	1	1	[3,6,14,27]	1
2	[2,5,10,23]	2,6	2	[2,5,10,23]	2,6
3	[2,4,9,18]	3,5	3	[2,4,9,18]	3,5
4	[2,4,8,18]	4,7	4	[2,4,8,18]	4
5	[1,2,4,8]	8	5	[2,4,7,15]	7
			6	[1,1,1,1]	8

Table 3. Atom Partitioning Using the List of Successive Extended Connectivity (LSEC) for Graph 3

priority	LSEC	atom no.
1	[3,9,22,58,137,353]	1
2	[3,8,20,48,120,283]	2
3	[3,7,19,45,117,275]	10
4	[3,7,19,44,116,270]	6
5	[3,6,14,32,75,183]	13
6	[2,5,12,30,71,179]	3,11
7	[2,5,11,29,67,175]	5,9
8	[2,5,11,29,66,174]	7
9	[2,5,11,26,62,146]	12
10	[2,4,10,23,59,138]	4
11	[2,4,10,22,58,133]	8
12	[1,1,1,1,1,1]	14

Table 4. Atom Partitioning Using the List of Successive Extended Connectivity (LSEC) for Graph 4

priority	LSEC	atom no.
1	[3,9,23]	1,2
2	[3,7,19]	3,6,7,10
3	[3,5,12]	4,5,8,9,11,12,13,14

to the shortest distance between one of the isospectral points and a terminal atom (here 1). Thus, correct partitioning of atoms is obtained after two iterations with the Wipke modification.

However, isospectral points may be at equal topological distances from terminal atoms, as in graph 3, or be in graphs without terminal atoms, as in graph 4. In graph 3, atoms 3 and 11 on the one hand and 5 and 9 on the other hand are isospectral. Application of our procedure results in an incomplete partitioning of atoms, whatever the method used to calculate the EC (Morgan or Wipke). For example, with the Wipke modification the LSEC obtained, and the resulting classes are included in Table 3.

Graph 4 gives an example of the second type of graph containing an isospectral point and no terminal atom, in which all the atoms of connectivity 2 (i.e., 4,5,8,9,11,12,-13,14) are isospectral, although they belong to two different classes [4,5,8,9] and [11,12,13,14]. Thus, incomplete partitioning is obtained, as shown in Table 4.

Finally, atoms belonging to regular graphs (that is, graphs in which all vertices have the same connectivity) cannot be partitioned at all by such a procedure. Cuneane (graph 5) gives a simple example of regular graphs. All atoms in this structure have the same degree, 3. Thus initial partitioning is unable to make any distinction between them, the LSEC being the same for all atoms, that is [3,9,27]. Thus, as pointed out as early as 1977 by Read[18] a complete search remains necessary, because the initial partitioning based on local graphs invariant may be incomplete.

The second step consists of the generation of a canonical linear code following the Morgan method but using the class
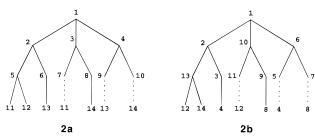
Figure 2. Canonical mapping Tree (CMT) and canonical labeled mapping tree (LCMT) of graph 3 (dotted lines are ring closure edges).
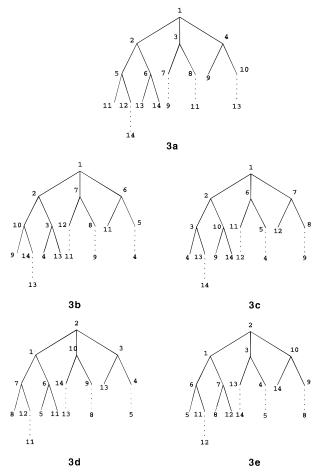

Figure 3. Canonical mapping tree (CMT) and canonical labeled mapping trees (LCMT) of graph 4 (dotted lines are ring closure edges).

Figure 4. Canonical mapping tree (CMT) and the four canonical labeled mapping trees (LCMTs) of cuneane (graph 5) (dotted lines are ring closure edges).

number instead of the extended connectivity as ordering criterion. This code represents the canonical mapping tree (CMT) of the molecular graph and allows the canonical numbering of atoms. If the number of classes equals the number of atoms, all atoms are distinguishable, and the CMT is obtained in only one way; thus, only one CLMT corresponding to a unique numbering is obtained. If a class contains several atoms which have not been differentiated by the previous steps they may be equivalent or not, and thus all possible numberings and corresponding CLMTs are generated by depth-first exploration of all possible selections of atoms within each class. During the process, minimal linear code is retained, and the corresponding CLMTs are stored. To optimize the search, the generation of linear code is aborted as soon as possible, i.e., when it appears to be greater than the current minimal code stored. The set of CLMTs obtained corresponds to the different equivalent canonical numberings of the structure.
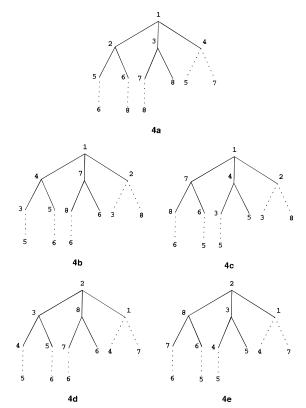
The third and last step consists of matching the different CLMTs obtained to identify atoms occupying the same node and the bonds occupying the same links, such atoms and bonds being topologically equivalent. Steps two and three of the procedure will be illustrated in graphs 3, 4, and 5. In the case of graph 3, initial partitioning leads to 12 classes, although all 14 atoms are topologically different, because two pairs of isospectral points could not be distinguished. The canonical numbering procedure leads to the CMT (Figure 2a); only one CLMT (Figure 2b) is generated, providing evidence that all atoms are different and giving the correspondence between initial and canonical numbering.

The CMT of graph 4 is represented in Figure 3a. The complete search results in the generation of four different CLMTs, represented in Figure 3b−e.

Matching the nodes of the four CLMTs leads to the constitution of four classes of equivalent atoms, namely {1,2}, {3,6,7,10}, {11,12,13,14}, and {4,5,8,9}. Similarly, matching the edges leads to six classes of equivalent bonds: {(1,6),(1,7),(2,3),(2,10)}, {(6,11),(7,12),(3,13),(10,14)}, {(11,12),(113,14)}, {(4,5),(8,9)}, and {(1,2)}. Figure 4a represents the CMT of cuneane (graph 5). The four different CLMTs obtained during the search are represented in Figure 4b−e.

Matching nodes of these trees leads to the constitution of three classes of equivalent atoms, namely {1,2}, {3,4,7,8}, and {5,6}. Similarly, matching links leads to five classes of equivalent bonds: {(1,2)}, {(5,6)}, {(1,4),(1,7),(2,3),-(2,8)}, {(3,5),(4,5),(6,7),(6,8)}, and {(3,4),(7,8)}.

IMPLEMENTATION

ABCLASS, a program applying the above method, is written in PDC Prolog[30] for microcomputer using the DOS operating system. As input the program reads the description

of the structure to be processed from files in standard SMD, Molfile, or Alchemy format and outputs its results in an ASCII file. The name of the input and output files can be passed to the program on the command line or can be introduced at run time. The characteristics of the Prolog language, i.e., nondeterminism, backtracking, pattern matching, simplicity in lists, and trees representation, make this language particularly well suited for this type of application.

The executable and source codes of this program and the structure files of the molecules studied are available as supporting information via the Internet. For this reason, we will not enter into technical details here, limiting ourselves to a short description of the program's logical structure. Therefore, the following program extracts are a slight simplification of the actual code. The description of the molecule read from the input file is stored in a dynamic database as atom and bond tables represented as Prolog clauses:

```
db_atom(at_num,at_type,nH,ch)

      at_num  : atom number,
      at_type : atom type,
      nH  : number of implicit hydrogens,
      ch  : formal charge.

db_bond(at1,at2,bd_type).

      at1  : atom number 1,
      at2  : atom number 2,
      bd_type  : bond type (1 single, 2 double, 3 triple, 4 aromatic).
```

The active part of the program consists of the proceed predicate which links the main steps of the method described above.

```
proceed :-
        build_LSEC,
        make_initial_classes,
        make_canonical_code,
        build_atom_classes,
        build_bond_classes.
```

The names of the predicates constituting the subgoals of the proceed predicate are sufficiently explicit to make further comments about their role almost unnecessary. Nevertheless, brief indications concerning their implementation will be given. "build_LSEC" is a recursive predicate (Prolog having no repetitive constructs, iteration is achieve by recursivity or backtracking) which constructs the LSEC of each atom of the molecule and stores these lists in the dynamic database. The boundary condition is satisfied when the number of different LSECs remains the same for two successive recursivity levels. "make_initial_classes" assigns each atom to a preliminary class ranked using the lexicographic order of their LSECs and local information associated with each atom directly available from the db_atom clauses or logically deductible from the db_bond clauses. An example of such deductible information is the ordered list of bond type originating at this atom. The lists of atoms constituting each of these temporary classes are kept in the dynamic database.

The predicates just described above are deterministic in the sense that there is only one path to their solution. On the other hand, the make_canonical_code predicate is nondeterministic, because if several atoms belong to the same class, each one can be arbitrarily chosen to build the canonical code. This is very fundamental to Prolog, the built-in backtracking mechanism of which allows elegant and efficient exploration of all possible solutions to such a problem. For example, let LA be the list of atoms belonging

to the same temporary class of the highest priority. Each one can be selected as possible root atom to construct the canonical matching tree. The following clauses will generate and store all the possible CLMTs rooted with all atoms which are members of the LA list.

```
member(A,[A|_]).
member(A,[_|R]) :- member(A,R).
make_canonical_code :-
        list_atoms_highest_rank(LA),
        member(R,LA),
        build_rooted_CMT(R,CMT,CLMT),
        store(CLMT),
        fail.
make_canonical_code.
```

The member predicate is a recursive, nondeterministic predicate, which extracts as first argument one element of the list received as second argument and sets a marker for backtracking on the second clause each time it succeeds on the first one. The backtracking itself is initiated when the build_rooted_CMT predicate fails to build a new CMT with the selected root atom or by the fail predicate at the end of the first make_canonical_code clause. The build_rooted-_CMT predicate is also nondeterministic and is constructed in a similar manner in order to choose as successor of a given node all possible atoms with the same priority. However, it is important to observe that only atoms which are bonded to the current node are considered at this stage, reducing considerably the number of possibilities to be examined. Also, as mentioned in the description of the method, but not illustrated in the program extract for sake of simplicity, the canonical linear code is compared during its generation to the current shortest code found so far in order to interrupt the search as soon as possible and provoke the search of another solution by backtracking to the last choice not explored.

As a consequence of this nondeterministic mechanism, all possible LCMTs are stored in the dynamic database and are matched recursively from their root by the build_atom-_classes and build_bond_classes predicates in order to gather in the same classes atoms nodes labels which occupy the same nodes and bonds occupying the same links. These predicates are deterministic. A separate program DSP-CLASS has been developed in Turbo-Pascal to read the output file of ABCLASS and display the equivalent atoms and bonds graphically. DSPCLASS is also available as supporting information. These programs, which take their parameters from the command line and their data from standard text files, can be easily used as external modules by other programs.

APPLICATION

Theoretical research in fullerene chemistry is a rapidly expanding field, with important contributions from both quantum chemistry and molecular graph theory. A large number of graph theoretical invariants and codes were proposed to quantify the structural features of fullerenes.[31−34] In order to demonstrate the computational efficiency of our general coding algorithm, we will present the results obtained for the topological symmetry identification for 20 C_{20}−C_{60} fullerenes. The Schlegel planar diagrams of the fullerenes considered are given in Figure 5, together with the labeling of the atoms. For seven structures we have presented modified Schlegel diagrams, so that the symmetry is depicted easily and the labeling of the atoms is clear.
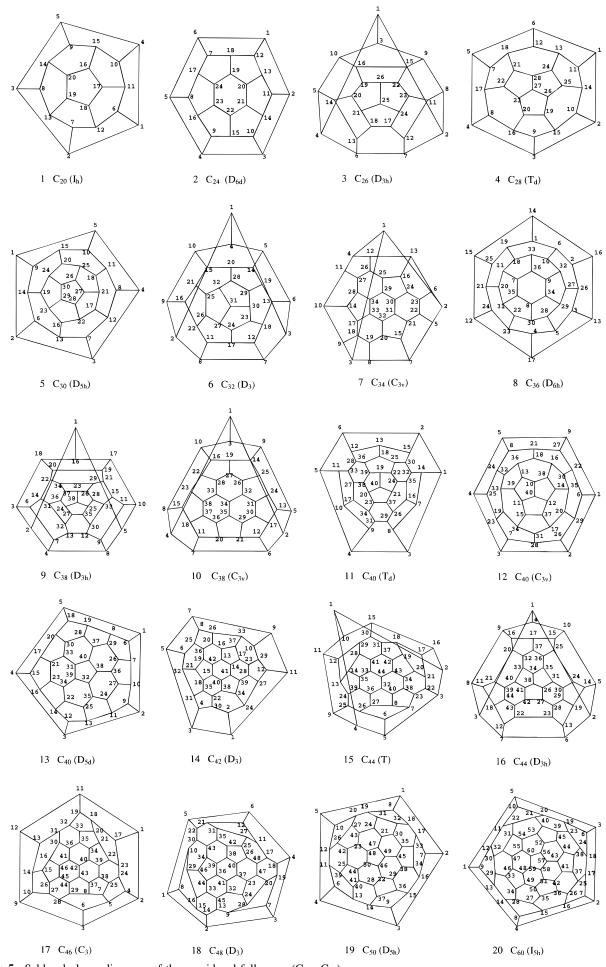
Figure 5. Schlegel planar diagrams of the considered fullerenes (C₂₀−C₆₀).

Table 5. Results for Each Fullerene: Search Time, Number of Atom and Bond Classes, and the Members (Atoms or Bonds, Respectively) of Each Class
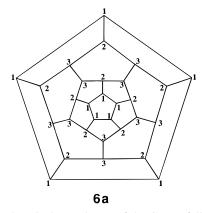
	fullerene	AC	equivalent atoms	BC	ACBA	equivalent bonds
1	C_{20} (I_h)	1	1−20	1	1-1	
	st = 16"					
2	C_{24} (D_{6d})	1	1−6, 19−24	1	1-1	
	st = 12"	2	7−18	2	1-2	
				3	2-2	
3	C_{26} (D_{3h})	1	1, 25	1	1-2	
	st = 18"	2	2−4, 21, 24, 26	2	2-3	
		3	5−10, 17−20, 22, 23	3	3-3	
		4	11−16	4	3-4	
				5	4-4	
4	C_{28} (T_d)	1	1, 3, 5, 13−18, 23, 26, 28	1	1-2	
	st = 19"	2	2, 4, 6, 8, 10, 12, 19−22, 24, 25	2	1-3	
		3	7, 9, 11, 27	3	2-2	
5	C_{30} (D_{5h})	1	1−5, 26−30	1	1-1	
	st = 33"	2	6−10, 21−25	2	1-2	
		3	11−20	3	2-3	
				4	3-3	
6	C_{32} (D_3)	1	1, 31	1	1-2	
	st = 25"	2	2−4, 27, 30, 32	2	2-3	(2,8) (3,6) (4,10) (23,30) (26,27) (28,32)
		3	5, 7, 9, 24, 25, 29	3	2-3	(2,9) (3,7) (4,5) (24,27) (25,32) (29,30)
		4	6, 8, 10, 23, 26, 28	4	3-4	
		5	11, 13, 15, 18, 20, 22	5	3-6	
		6	12, 14, 16, 17, 19, 21	6	4-5	
				7	5-5	
				8	5-6	
				9	6-6	
7	C_{34} (C_{3v})	1	1	1	1-2	
	st = 35"	2	2, 3, 4	2	2-3	
		3	5, 6, 8, 9, 11, 12	3	3-4	
		4	7, 10, 13	4	3-7	
		5	14, 15, 16	5	4-5	
		6	17, 20, 21, 24, 25, 28	6	5-6	
		7	18, 19, 22, 23, 26, 27	7	6-7	
		8	29−34	8	6-8	
				9	7-7	
				10	8-8	(29,30) (31,32) (33,34)
				11	8-8	(29,34) (30,31) (32,33)
8	C_{36} (D_{6h})	1	1−4, 11, 24, 25, 29−33	1	1-2	
	st = 1' 01"	2	5, 6, 10, 18−23, 26−28	2	1-3	
		3	7−9, 12−17, 34−36	3	2-3	
				4	3-3	
9	C_{38} (D_{3h})	1	1, 38	1	1-2	
	st = 1' 11"	2	2, 5, 16, 23−25	2	2-3	
		3	3, 4, 8, 10, 17, 18, 26−28, 35−37	3	3-3	
		4	6, 7, 9, 11, 19, 20, 29−34	4	3-4	
		5	12−15, 21, 22	5	4-4	
				6	4-5	
				7	5-5	
10	C_{38} (C_{3v})	1	1	1	1-2	
	st = 32"	2	2−4	2	2-3	
		3	5−10	3	3-3	
		4	11−16	4	3-4	
		5	17−19	5	4-5	
		6	20−25	6	4-6	
		7	26, 28, 29, 31, 36, 38	7	5-8	
		8	27, 30, 37	8	6-6	
		9	32, 33, 35	9	6-7	
		10	34	10	7-8	
				11	7-9	
				12	9-10	
11	C_{40} (T_d)	1	1−6, 22, 23−39	1	1-1	(1,2) (3,4) (5,6) (22,25) (23,34) (24,35) (26,37) (27,38) (28,33) (29,31) (30,32) (36,39)
	st = 1' 06"	2	7, 10, 13, 40	2	1-1	(1,3) (2,6) (4,5) (22,24) (23,37) (25,30) (26,29) (27,33) (28,36) (31,34) (32,35) (38,39)
		3	8, 9, 11, 12, 14−21	3	1-3	
				4	2-3	
12	C_{40} (C_{3v})	1	1−5, 9	1	1-1	(1,2) (3,4) (5,9)
	st = 1' 06"	2	6−8, 25−27	2	1-1	(1,9) (2,3) (4,5)
		3	10−12	3	1-2	
		4	13−15, 37−39	4	2-7	
		5	16, 17, 33−36	5	2-8	
		6	18−20	6	3-4	
		7	21, 23, 29	7	3-10	
		8	22, 24, 28	8	4-4	
		9	30−32	9	4-5	
		10	40	10	5-6	
				11	5-9	
				12	6-7	
				13	8-9	

Table 5. Continued

fullerene	AC	equivalent atoms	BC	ACBA	equivalent bonds
13 C_{40} (D_{5d})	1	1−5, 31, 32, 38−40	1	1-1	
st = 1' 07"	2	6, 9, 12, 15, 18, 33−37	2	1-2	
	3	7, 8, 10, 11, 13, 14, 16, 17, 19−30	3	2-3	
			4	3-3	(7,10) (8,19) (11,13) (14,16) (17,20) (21,23) (22,25) (24,27) (26,29) (28,30)
			5	3-3	(7,26) (8,29) (10,27) (11,24) (13,25) (14,22) (16,23) (17,21) (19,28) (20,30)
14 C_{42} (D_3)	1	1, 5, 9, 14, 40, 42	1	1-2	
st = 1' 03"	2	2, 6, 10, 17, 19, 38	2	1-3	(1,3) (5,7) (9,11) (13,14) (15,42) (40,41)
	3	3, 7, 11, 13, 15, 41	3	1-3	(1,11) (3,5) (7,9) (13,42) (14,41) (15,40)
	4	4, 8, 12, 16, 18, 39	4	2-6	
	5	20, 26−28, 31, 35	5	2-7	
	6	21, 25, 29, 30, 34, 37	6	3-4	
	7	22−24, 32, 33, 36	7	4-5	
			8	4-6	
			9	5-5	
			10	5-7	
			11	6-7	
15 C_{44} (T)	1	1, 8, 14, 19	1	1-2	
st = 54"	2	2, 4, 7, 10, 13, 18, 20, 27, 28, 35, 40, 42	2	2-3	
	3	3, 9, 15, 21, 23, 24, 26, 29, 31−33, 43	3	2-4	
	4	5, 6, 11, 12, 16, 17, 34, 36−39, 41	4	3-4	
	5	22, 25, 30, 44	5	3-5	
			6	4-4	
16 C_{44} (D_{3h})	1	1, 38	1	1-2	
st = 1' 39"	2	2−4, 26, 34, 41	2	2-3	
	3	5−10, 27, 31, 33, 35, 40, 42	3	3-3	
	4	11−16, 28, 30, 32, 36, 39, 43	4	3-4	
	5	17−19, 29, 37, 44	5	4-5	
	6	20−25	6	4-6	
			7	5-5	
			8	6-6	
17 C_{46} (C_3)	1	1, 3, 12	1	1-2	(1,2) (3,9) (11,12)
st = 1' 14"	2	2, 9, 11	2	1-2	(1,11) (2,3) (9,12)
	3	4, 10, 19	3	1-5	
	4	5, 14, 18	4	2-3	
	5	6, 13, 17	5	3-4	
	6	7, 15, 20	6	3-10	
	7	8, 16, 21	7	4-5	
	8	22, 29, 30	8	4-6	
	9	23, 28, 31	9	5-9	
	10	24, 27, 32	10	6-7	
	11	25, 26, 33	11	6-11	
	12	34, 37, 46	12	7-8	
	13	36, 39, 44	13	7-12	
	14	40, 41, 43	14	8-9	
	15	35, 38, 45	15	8-13	
	16	42	16	9-10	
			17	10-11	
			18	11-15	
			19	12-14	
			20	12-15	
			21	13-14	
			22	13-15	
			23	14-16	
18 C_{48} (D_3)	1	1, 3, 6, 36−38	1	1-2	(1,2) (3,4) (5,6) (36,39) (37,41) (38,40)
st = 1' 04"	2	2, 4, 5, 39−41	2	1-2	(1,5) (2,3) (4,6) (36,41) (37,40) (38,39)
	3	7, 8, 12, 23, 42, 44	3	1-3	
	4	9−11, 33, 35, 47	4	2-6	
	5	13, 17, 30, 43, 45, 48	5	3-4	
	6	14, 18, 22, 26, 32, 34	6	3-7	
	7	15, 19, 21, 24, 25, 46	7	4-5	
	8	16, 20, 27−29, 31	8	4-8	
			9	5-5	
			10	5-6	
			11	6-7	
			12	7-8	
			13	8-8	
19 C_{50} (D_{5h})	1	1−5, 46−50	1	1-1	
st = 3' 09"	2	6−10, 21−23, 44, 45	2	1-2	
	3	11−20, 24, 25, 28−30, 35, 38, 41−43	3	2-3	
	4	26, 27, 31−34, 36, 37, 39, 40	4	3-3	
			5	3-4	
			6	4-4	

Table 5. Continued

fullerene		AC	equivalent atoms	BC	ACBA	equivalent bonds
20	C_{60} (I_{5h}) st = 12' 47"	1	1-60	1	1-1	(1,4) (1,5) (2,3) (2,4) (3,5) (6,18) (6,19) (7,16) (7,17) (8,14) (8,15) (9,12) (9,13) (10,11) (10,20) (11,22) (12,30) (13,29) (14,28) (15,27) (16,26) (17,25) (18,24) (19,23) (20,21) (21,22) (23,24) (25,26) (27,28)(29,30) (31,32) (31,54) (32,47) (33,34) (33,46) (34,49) (35,36) (35,50)(36,42) (37,38) (37,41) (38,44) (39,40) (39,45) (40,53) (41,43) (42,51)(43,44) (45,52) (47,55) (46,48) (48,49) (50,51) (52,53) (54,55) (56,57)(56,60) (57,58) (58,59) (59,60)
				2	1-1	(1,9) (2,7) (3,6)] (4,8) (5,10) (11,12) (13,14) (15,16) (17,18) (19,20) (21,40)(22,31) (23,39) (24,38) (25,37) (26,36) (27,35) (28,34) (29,33) (30,32) (41,42)(43,57) (44,45) (46,47) (48,59) (49,50) (51,58) (52,56)(53,54) (55,60)
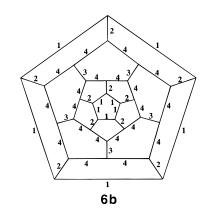
Figure 6. Atom and bond equivalence classes of the $C_{30}D_{5h}$ fullerene.

Table 5 presents for each fullerene the time (st) needed to determine the canonical code and topological symmetry on a PC 486 running at 66 MHz, the number of atom and bond classes, and the elements (atoms or bonds, respectively) which constitute each class. Atoms are identified by the corresponding label in the drawing of the fullerene from Figure 5. In this table, each bond class is identified by the atom class of the two bonded atoms (ACBA). When two different bond classes correspond to the same ACBA, the individual bonds of each class are labeled with the numbers of the two bonded atoms. Although the code has not been optimized for speed, but clarity was privileged, the computational time is kept within reasonable limits, demonstrating that a general coding program can provide a rapid answer to the topological symmetry problem of fullerenes.

As outlined in the introduction, the determination of atom and bond equivalence classes of fullerenes has both a practical and theoretical importance, enabling one to generate all isomers of heterofullerenes or all addition isomers to a particular fullerene. The exhaustive generation of all heterofullerene isomers is necessary in molecular or quantum studies for the identification of the most stable isomer, giving valuable information to the experimental chemist interested in the synthesis of such compounds. From the information on atom equivalence classes provided in Table 5, one can easily generate all mono heterofullerenes by selecting from each atom class one carbon atom and substituting it with a heteroatom X in the molecular graph of the fullerene.

As an example, we present in Figure 6a the three atom classes (labeled 1, 2, and 3) of the $C_{30}D_{5H}$ fullerene. The replacement of one carbon atom from each class leads to three possible heteroisomers $C_{29}X$. A more general algorithm and program, allowing one to generate from a C_n fullerene all topological distinct $C_{n-m}Y_m$ heterofullerenes, was developed, based on the ABCLASS program for canonical

Chart 1. Algorithm in pseudo code for the generation of 1 to m heterofullerenes.

```
let  m         be the maximum number of atoms to be replaced.
let  S_{i,s}    be the i^th Structure of the heterofullerene with s replaced carbons.
let  CC_{i,s}   be the Canonical Code of S_{i,s}.
let  LAC_{i,s}  be the List of Atoms Classes of S_{i,s}.
let  N_s        be the Number of of s heterofullerene isomers.
let  CS         be the Candidate Structure.

initialize  s := 0
initialize  N_0 := 1
initialize  S_{1,0}  with the Structure of the considered fullerene
generate the Canonical Code CC_{1,0} and the List of Atoms Classes LAC_{1,0}
    of S_{1,0} using ABCLASS

while s < m do
    ns := s + 1
    N_ns := 0
    for for each Structure S_{i,s} (i from 1 to N_s) do
        for each Atom Class of sp_2 Carbon do
            select an Atom of this Class
            replace with Carbon Atom by heteroatom X giving
            the Candidate Structure CS
            generate the Canonical Code CC and the List of Atoms Classes
            LAC of the Candidate Structure CS
            if CC is a new code then
                save the Candidate Structure CS as a new generated
                heterofullerene with its Canonical Code and
                List of Atoms Classes
                N_ns := N_ns + 1
                S_{Nns,ns} := CS
                CC_{Nns,ns} := CC
                LAC_{Nns,ns} := LAC
    s := ns
end
```

coding and equivalence class determination. We will briefly describe this algorithm (see Chart 1), which considers that the heteroatom X is a trigonal sp^2 atom; the implementation details are given elsewhere.[35]

Owing to the importance of addition reactions in the chemistry of fullerenes, the generation of all addition isomers has both a theoretical and an experimental importance, giving an indication of the structure of the most probable isomer. By knowing the number and members of bond equivalence classes of a fullerene C_n, one can generate all addition

isomers C_nY_2 by selecting one bond from each class, making it a simple bond, and attaching to each of the two atoms of the bond an atom Y, where Y can represent hydrogen or a halogen. Considering again the $C_{30}D_{5h}$ fullerene, its four bond classes, which are shown in Figure 6b, will generate four addition isomers with the formula $C_{30}Y_2$. The same algorithm can generate all cycloaddition isomers of a parent fullerene. A slight modification of the algorithm, using ABCLASS for generation of heterofullerenes presented above, can be applied for the exhaustive generation of all addition fullerene isomers with the formula C_nY_m, where $m = 2k$. The changes consist of (a) the selection of a bond which is a double bond in at least one Kekulé structure in the considered molecule from each bond class instead of C_{sp^2} from each atom class, (b) the connection of a Y atom to each of the two carbon atoms linked by the selected bond instead of the replacement of the selected atom, and (c) the modification of the selected bond type to single. The ABCLASS program can be used for the further generation of the atom and bond equivalence classes of the fullerene derivatives. For example, the three $C_{29}X$ isomers generated by the replacement of a carbon atom from the three atom classes presented in Figure 6a have 18 atom classes and 25 bond classes. The four $C_{30}Y_2$ addition isomers generated from the four bond classes in Figure 6b present the following numbers of equivalence classes: 18 atom and 25 bond classes for the two $C_{30}Y_2$ isomers from bond classes 1 and 2, 9 atom and 14 bond classes for $C_{30}Y_2$ from bond class 3, and 30 atom and 45 bond classes for $C_{30}Y_2$ generated from bond class 4.

The information provided by the topological symmetry of a molecule may be used for the determination of the number and intensity patterns of ^{13}C NMR signals.[36] The number of carbon atoms equivalence classes of a molecular graph gives the number of ^{13}C NMR signals, considering that two carbon atoms situated in the same equivalence class are magnetically equivalent and thus would resonate at the same magnetic field. Because all carbon atoms in the same equivalence class resonate at the same applied magnetic field, the intensity ratio of the ^{13}C NMR signals is simply the ratio of the number of carbon atoms in each class. Thus, the identification of topological symmetry can be used in an expert system to generate ^{13}C NMR spectra starting from the molecular graph. On the basis of maximal point-group assigned to a fullerene graph, Manolopoulos and Fowler defined a simple procedure for computation of the number of ^{13}C NMR signals, IR-active and Raman-active normal modes.[37] In certain cases, pseudo closed-shell fullerenes (i.e., those fullerenes containing two electrons in every occupied bonding MO, but with a bonding LUMO, computed by the Hückel method) will present a second-order Jahn-Teller distortion.[38] The loss of symmetry will change the number of ^{13}C NMR signals as determined on the basis of topological symmetry. In the present investigation we will not consider the possibility of a Jahn-Teller distortion, and the discussion concerning ^{13}C NMR signatures of fullerenes will be made on the basis of topological symmetry alone. Considering the same $C_{30}D_{5h}$ fullerene, the inspection of the atom class information presented in Table 5 shows that each of the three sets of atoms contains 10 equivalent atoms. The idealized ^{13}C NMR spectrum (i.e., considering only the topological symmetry) should therefore consist of three lines with intensities in the ratio 1:1:1. In the characterization of higher fullerenes, the relationship between the topological symmetry

and the number of ^{13}C NMR signals and intensity patterns is useful to experimentalists, because it can give information on the fullerene isomer which can produce a given spectrum. Also, using the same information, one can determine if the sample contains only one isomer or a mixture of isomers. The computer generation of atomic equivalence classes based on the three-dimensional molecular structure was recently defined;[39] one must note that the present program can also be used for the exhaustive generation of three-dimensional atomic equivalence classes of any molecular structure. Recently, Balasubramanian obtained theoretical ESR hyperfine patterns for the radicals derived from C_{60} and C_{70} fullerenes from combinatorial techniques based on group-theoretical generating functions.[40] The program ABCLASS, since it can generate the atom equivalence classes in a reasonable amount of time, can be used for the efficient generation of the theoretical ESR hyperfine patterns for the radicals derived from a fullerene.

CONCLUSIONS

Although there are many molecular coding algorithms, a large number are in fact incomplete and are not able to give reliable results for highly symmetric graphs. On the other hand, complete coding algorithms are usually thought to be not applicable to large molecules with a high symmetry. We have presented a general, efficient coding algorithm which was implemented in Prolog and use a two stage procedure: (a) the atoms of the molecular graph are partitioned in classes according to a topological invariant, and (b) all canonical labeled trees and ring closure lists are generated, which in the end of the procedure offers the canonical code. Because it makes an exhaustive search of all labelings which can generate a minimal code, the algorithm is guaranteed to generate a good code. As a byproduct of the code, the algorithm identifies the topological symmetry elements of the molecular graph, i.e., the atom and bond equivalence classes. The efficiency of the proposed algorithm was tested for a set of 20 $C_{20}-C_{60}$ fullerenes. The determination of fullerene topological symmetry has both a theoretical and experimental importance. As presented in the paper, identification of the atom equivalence classes of a fullerene C_n allows one to generate all heterofullerenes isomers $C_{n-1}X$. An algorithm which permits the generation of all heterofullerene topological isomers with the formula $C_{n-m}X_m$ was presented; the details of the implementation are given elsewhere. Generation of the molecular structure for heterofullerene isomers is necessary in molecular and quantum mechanics studies on the structure and stability of fullerene derivatives. Similarly, the generation of bond equivalence classes of a fullerene C_n permits generation of all addition isomers C_nY_2 or cycloaddition isomers; the algorithm which permits the generation of all addition isomers of a fullerene is described; the implementation details are presented elsewhere. Generation of multiple addition isomers to fullerenes is important in studies related to the thermodynamics and isomer structure identification. Also, the number of atom classes and the number of atoms in each class gives important information on the number of ^{13}C NMR signals and intensity patterns in fullerenes. The same information can be used for the generation of ESR hyperfine patterns in fullerene radicals.

REFERENCES AND NOTES

(1) Kroto, H. W.; Heath, J. R.; O'Brien, S. C.; Curl, R. F.; Smalley, R. E. C_{60}: Buckminsterfullerene. *Nature* **1985**, *318*, 162−163.

(2) Hirsch, A. *The Chemistry of the Fullerenes*; Georg Thieme Verlag: Stuttgart, 1994.

(3) Henderson, C. C.; Rohlfing, C. M.; Assink, R. A.; Cahill, P. A. $C_{60}H_4$: Kinetics and Thermodynamics of Multiple Addition to C_{60}. *Angew. Chem., Int. Ed. Engl.* **1994**, *33*, 786−788.

(4) Birkett, P. R.; Hitchcock, P. B.; Kroto, H. W.; Taylor, R.; Walton, D. R. M.; Preparation and Characterization of $C_{60}Br_6$ and $C_{60}Br_8$. *Nature* **1992**, *357*, 479−481.

(5) Birkett, P. R.; Avent, A. G.; Darwish, A. D.; Kroto, H. W.; Taylor, R.; Walton, D. R. M. Preparation and ^{13}C NMR Spectroscopic Characterisation of $C_{60}Cl_6$. *J. Chem. Soc., Chem. Commun.* **1993**, 1230−1232.

(6) Matsuzawa, N.; Dixon, D. A.; Fukunaga, T. Semiempirical Calculations of Dihydrogenated Buckminsterfullerenes, $C_{60}H_2$. *J. Phys. Chem.* **1992**, *96*, 7594−7604.

(7) Clare, B. W.; Kepert, D. L. Structures and Stabilities of Hydrofullerenes $C_{60}H_n$. *J. Mol. Struct. (Theochem)* **1993**, *281*, 45−52.

(8) Henderson, C. C.; Rohlfing, C. M.; Cahill, P. A. Theoretical Studies of Selected $C_{60}H_2$ and $C_{70}H_2$ Isomers. *Chem. Phys. Lett.* **1993**, *213*, 383−388.

(9) Dunlap, B. I.; Brenner, D. W.; Schriver, G. W. Symmetric Isomers of $C_{60}H_{36}$. *J. Phys. Chem.* **1994**, *98*, 1756−1757.

(10) Bowser, J. R.; Jelski, D. A.; George, T. F. Stability and Structure of $C_{12}B_{24}N_{24}$: A Hybrid Analogue of Buckminsterfullerene. *Inorg. Chem.* **1992**, *31*, 154−156.

(11) Karfunkel, H. R.; Dressler, T.; Hirsch, A. Heterofullerenes: Structure and Property Predictions, Possible Uses and Synthesis Proposals. *J. Comput.-Aided Mol. Design* **1992**, *6*, 521−535.

(12) Trinajstic, N.; Nikolic, S.; Knop, J. V.; Müller, W. R.; Szymanski, K. *Computational Chemical Graph Theory*; Ellis Horwood: New York, 1991.

(13) Trinajstic, N. *Chemical Graph Theory*; CRC Press: Boca Raton, FL, 1992.

(14) Hong, H.; Xin, X. ESSESA: An Expert System for Structure Elucidation from Spectra. 4. Canonical Representation of Structures. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 730−734.

(15) Hu, C.-Y.; Xu, L. A New Scheme for Assignment of a Canonical Connection Table. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 840−844.

(16) Morgan, H. L. Generation of a Unique Machine Description for Chemical Structures—A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 107−113.

(17) Wipke, W. T.; Dyott, T. M. Stereochemically Unique Naming Algorithm. *J. Am. Chem. Soc.* **1974**, *96*, 4834−4842.

(18) Read, R. C.; Corneil, D. G. The Graph Isomorphism Disease. *J. Graph Theor.* **1977**, *1*, 39−363.

(19) Jochum, C.; Gasteiger, J. Canonical Numbering and Constitutional Symmetry. *J. Chem. Inf. Comput. Sci.* **1977**, *17*, 113−117.

(20) Carhart, R. E. Erroneous Claims Concerning the Perception of Topological Symmetry. *J. Chem. Inf. Comput. Sci.* **1978**, *18*, 108−110.

(21) Schubert, W.; Ugi, I. Constitutional Symmetry and Unique Description of Molecules. *J. Am. Chem. Soc.* **1978**, *100*, 37−41.

(22) Shelley, C. A.; Munk, M. J. An Approach to the Assignment of Canonical Connection Tables and Topological Symmetry Perception. *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 247−250.

(23) Balaban, A. T.; Mekenyan, O.; Bonchev, D. Unique Description of Chemical Structures Based on Hierarchically Ordered Extended Connectivities (HOC Procedures). I. Algorithms for Finding Graph Orbits and Canonical Numbering of Atoms. *J. Comput. Chem.* **1985**, *6*, 538−551.

(24) Liu, X.; Balasubramanian, K.; Munk, M. E. Computational Techniques for Vertex Partitioning of Graphs. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 263−269.

(25) Rücker, G.; Rücker, C. Computer Perception of Constitutional (Topological) Symmetry: TOPSYM, a Fast Algorithm for Partitioning Atoms and Pairwise Relations among Atoms into Equivalent Classes. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 187−191.

(26) Rücker, G.; Rücker, C. On Using the Adjacency Matrix Power Method for Perception of Symmetry and for Isomorphism Testing of Highly Intricate Graphs. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 123−126.

(27) Rücker, G.; Rücker, 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*, 422−427.

(28) Varmuza, K.; Scsibrany, H. Investigation of Molecular Similarities using a Turbo-Pascal Program on IBM-compatible Personal Computers. *Lab. Microcomput.* **1994**, *12*, 99−103.

(29) Liu, X.; Balasubramanian, K. Computer Generation of Edge Groups and Edge Colorings of Graphs. *J. Comput. Chem.* **1989**, *10*, 1−3.

(30) PDC Prolog, Prolog Development Center, H. J. Holst Vej 5A, DK-2605 Brondby-Copenhagen, Denmark.

(31) Balasubramanian, K. Laplacian Polynomials of Fullerenes (C_{20}−C_{40}). *Chem. Phys. Lett.* **1994**, *224*, 325−332.

(32) Manoharan, M.; Balakrishnarajan, M. M.; Venuvanalingam, P.; Balasubramanian, K. Topological Resonance Energy Predictions of the Stability of Fullerene Clusters. *Chem. Phys. Lett.* **1994**, *222*, 95−100.

(33) Balaban, A. T.; Liu, X.; Klein, D. J.; Babic, D.; Schmalz, T. G.; Seitz, W. A.; Randic, M. Graph Invariants for Fullerenes. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 396−404.

(34) Babic, D.; Balaban, A. T.; Klein, D. J. Nomenclature and Coding of Fullerenes. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 515−526.

(35) Laidboeur, T.; Cabrol-Bass, D.; Ivanciuc, O. Exhaustive Generation of Heterofullerenes Topological Isomers Using a Prolog Program. Manuscript in preparation.

(36) Liu, X.; Balasubramanian, K.; Munk, M. E. Computer-Assisted Graph-Theoretical Construction of ^{13}C NMR Signal and Intensity Patterns. *J. Magn. Reson.* **1990**, *87*, 457−474.

(37) Manolopoulos, D. E.; Fowler, P. W. Molecular Graphs, Point Groups, and Fullerenes. *J. Chem. Phys.* **1992**, *96*, 7603−7614.

(38) Fowler, P. W.; Sandall, J. P. B. Prediction of Spectral Signatures of Fullerenes. Second-Order Jahn-Teller Effects on the Structures of C_{44}, C_{56}, C_{68} and C_{92}. *J. Chem. Soc., Perkin Trans. 2* **1994**, 1917−1921.

(39) Balasubramanian, K. Computer Generation of Nuclear Equivalence Classes Based on the Three-Dimensional Molecular Structure. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 243−250.

(40) Balasubramanian, K. ESR Hyperfine Patterns for the Radicals Derived from C_{60} and C_{70}. *Chem. Phys. Lett.* **1993**, *210*, 153−158.

CI9500593