Nomenclature and Coding of Fullerenes

D. Babić*

Institute Rugjer Bošković, HR-41001 Zagreb, P.O.B. 1016, Croatia

A. T. Balaban[‡] and D. J. Klein

Texas A&M University at Galveston, Galveston, Texas 77553-1675

Received October 27, 1994[⊗]

An algorithm for deriving IUPAC names of fullerenes is described. Several possible ways for coding of fullerenes are also discussed: the known spiral ring code and its modified version, the Weinberg code, the code derived from the Eulerian trail, the code derived from the Brugesser–Mani theorem, two others based on a Hamiltonian cycle and a Hamiltonian path, and the code based on a path-cotree. A special nomenclature for fullerenes and their derivatives is under consideration in CAS and IUPAC bodies, based upon a spiral passing through all atoms in a molecule. Some isomers relevant for this scheme are pointed out: two highly symmetric isomers with no such spiral and one isomer with such a spiral in only one direction.

1. INTRODUCTION

The discovery,[1] and especially the successful preparation,[2] of spherical carbon molecules called fullerenes has led to an explosion of research[3] in many branches of science. Fullerenes are a family of molecules rich in isomers with a great structural similarity, and their existence has accentuated the problem of unique and efficient naming of organic chemical compounds. Although not strictly defined, fullerenes are thought of as spherical molecules with only five- and six-membered rings. Related structures including also four-membered rings were theoretically investigated, but no evidence for their existence is obtained yet. In the present paper we will be concerned only with those involving only pentagons and hexagons, but many concepts are applicable to other types as well. The present IUPAC nomenclature is based on Adolf von Baeyer's rules for bridged polycyclic systems.[4] In principle the current nomenclature is quite applicable to fullerenes as well; however, its use is rather difficult and error prone as exemplified by a series of articles.[5–9] The limited availability of different isomeric fullerenes turns out to be a lucky circumstance. The naming of fullerenes according to these rules is unfeasible without a computer. Even then the task may become intractable, as the problem involved in deriving a unique name belongs to the so-called NP-complete class.[10] This class of problems is characterized by nonpolynomial complexity which means that the time needed for performing the task increases with the size of the considered object faster than any polynomial function. Sooner or later, no program and no computer will be capable to handle the task properly. In the present case the problem is to find all largest cycles in the fullerene graph. Even for the special case of cubic graphs, in which all the vertices have degree three, the problem is known to be NP-complete.[10] Fullerenes are a subclass of cubic graphs, but, probably, this does not ease the solvability.

The general problem of fullerene nomenclature has been considered first by R. Taylor.[11] He proposed a numbering of atoms in those fullerene isomers that were isolated so far. However, the proposal was intended mainly to bring some order into naming fullerene derivatives which were being produced at an increasing rate. As to the fullerenes themselves, there are no new rules which would be generally applicable.

In the first part of the present paper we describe our computer program which can be used to obtain the name according to the present IUPAC nomenclature. In the second part we discuss some possible alternatives to the existing system. Finally, as there is already a nomenclature proposal in preparation,[12] which relies on a "face spiral" containing all atoms of a fullerene molecule, we present two fullerenes of high symmetry without such a spiral and one isomer with such a spiral in only one direction. These examples could be of interest to investigators in the field.

2. DERIVING FULLERENE NAMES ACCORDING TO THE IUPAC NOMENCLATURE

The following rules for deriving the name of polycyclic carbon compounds[4] are applied sequentially until a unique choice is obtained.

(1) The main ring should contain as many carbon atoms as possible. Two atoms serve as the bridgeheads for the main bridge.

(2) The main bridge should be as large as possible.

(3) The main bridge should divide the main ring as symmetrically as possible.

(4) Vertices on the main ring are numbered sequentially by starting with the longest of the two paths connecting the bridgeheads of the main bridge and following with the shorter path. Any vertex in the main bridge is then numbered sequentially starting with the vertex closest to the bridgehead vertex numbered first.

(5) Secondary bridges are denoted in decreasing order of their size, by their size number and two superscripts corresponding to positions of the two bridgeheads.

(6) The superscripts locating the secondary bridges should be as small as possible.

The most difficult part is finding all the maximal rings in the molecule. A systematic examination of fullerene isomers with up to 70 carbon atoms, and of isomers with no abutting

[‡] Permanent address: Dept. of Organic Chemistry, Polytechnic University, Bucharest, Roumania.

[⊗] Abstract published in *Advance ACS Abstracts*, April 15, 1995.

0095-2338/95/1635-0515$09.00/0

pentagons going up to 100 carbon atoms revealed that all of them possess Hamiltonian cycles.[13] A Hamiltonian cycle is characterized by passing through each atom exactly once. Therefore, all the examined isomers possess a maximal ring which contains all atoms. A literature search revealed that Hamiltonian cycles are conjectured to be present in the class of graphs which contains fullerenes,[14,15] and the conjecture is still open. This hypothesis, if true, facilitates the finding of IUPAC names since one knows in advance the size of the maximal ring, which appropriately narrows the search; therefore no bridge contains any vertex. The program presented here involves this assumption for the same reasons (but notifies the user if the hypothesis is not valid in a particular case). It is possible that in this way a counterexample may be found, but, in that case, it will be a serendipitous result.

A general program for obtaining the IUPAC name of a polycyclic molecule has been recently developed by Rückers.[6] It is also quite applicable to fullerenes. However, due to the general character of the algorithm, the CPU times needed for processing fullerenes appear to be quite large (181.8 min for C_{60} on IBM 3090). The program described here is designed specifically for fullerenes but can be equally well applied to any molecule in which all vertices of the hydrogen depleted graph are trivalent. Its efficiency may be demonstrated by the same C_{60} for which only 7 s were needed on PC 486/66 MHz. The program consists of two separate procedures. The first one performs the search for all Hamiltonian cycles, and the second one derives the names from each cycle and compares them with the best one obtained from the previous search. If the new one conforms better to the rules 1−6, it replaces the old one which was kept so far. The first part of the program is independent of the second part and can be used separately for other purposes as well.

The program is written in FORTRAN 77 and can be obtained from D.B. upon request.[16] Basically, the algorithm works by trying to prolong the path until the Hamiltonian cycle is produced. The path starts at any vertex and proceeds by any of the incident edges. In each step there are two vertices, CHOICE, by which the path can be prolonged, at least in principle. Whenever the choice is possible, one of the vertices is chosen, and the other is put on stack, NGBHD(PTR). If the path gets stuck, BACKTRACK procedure is performed. It returns the path to the most recent stacked branching and proceeds by the stacked vertex. Particular care is taken in order to detect early a blind alley. As each vertex is trivalent, whenever the choice of the next vertex includes one whose neighbor is already in the path (NGBHD), the path must proceed via this vertex, or else it will get stuck sooner or later. The exceptions are the vertices in ENDS, one of which must be the ending vertex of the path. When the path passes by one of these vertices, ENDFLAG is made true to denote that the ending vertex is fixed. If the same would happen with another vertex in ENDS, BACKTRACK is called.

For the sake of space, we give only the pseudocode of the main procedure which performs the search for all Hamiltonian cycles. The second procedure (EXAMINE_FOR_THE_NAME) is omitted as its realization is not complicated. The algorithm is structured around a few particular data types. CONT(I,J) = {CONT1, CONT2} denotes a set of two vertices by which the path going from I to J can be continued. In other words: I, CONT1, and CONT2 are the neighbors of J. The variables ENDS and CHOICE are also sets of two vertices. NGBHD is a one-dimensional array, here considered also as a set of vertices adjacent to any of the vertices in PATH. Whenever the set is implied, it is symbolized by NGBHD(1:PTR). The same applies to PATH; when referred to as a set, it implies the elements indexed by 1 through LEN, as denoted by PATH(1:LEN). PATH is a sequence of vertices currently occupied by the path of the length LEN. TRACK is used to count the branchings of the path which are not yet explored. PTR indexes the sequence of vertices adjacent to the path. Whenever a new branching appears, the current LEN and PTR are stored in PATHPTR and NGBPTR, respectively. Thus in BACKTRACK, when the path is returned to the most recent branching, the values of PTR and LEN are easily restored.

The algorithm works in two sweeps. First, the path starts at an arbitrary vertex V, proceeds by any of its three neighbors, say U, and returns to the starting vertex by any of the two remaining neighbors. This time, one of the two remaining neighbors of V, say W, is put into NGBHD with a special purpose. Namely, NGBHD has two functions: as the set containing vertices passed by the path, which have to be distinguished in order to early detect blind alleys, and as an ordered stack of not yet examined branches. The last function is used in the BACKTRACK. When the program starts, the first vertex put into NGBHD (line 6 of the pseudocode) needs to be there in order to appropriately start the second sweep in which the vertex U will be substituted by W, and the path will be allowed to close only by the last of the three neighbors of the starting vertex V. However, if the vertex U would be recognized as an element of NGBHD in the first sweep, it would prevent return of the path through this vertex. In that order variable BTM is in the first sweep set equal to 2, and the first element in NGBHD cannot be "seen" in the statements checking vertices as elements of NGBHD. However, in the second sweep, which starts when TRACK for the first time assumes the value of 1, this is not adequate any more, and in BACKTRACK, BTM becomes equal to 1.

For a few best known fullerenes[11] shown in Figure 1, the IUPAC names are given in Table 1. CPU times needed for their derivation are given in the first column of Table 8. Only nontrivial parts of the names are given in Table 1. The IUPAC name for a fullerane, which is a hydrocarbon $(CH)_N$, i.e., a valence isomer of [N]annulene, assumed to possess a Hamiltonian circuit, consists of the following:

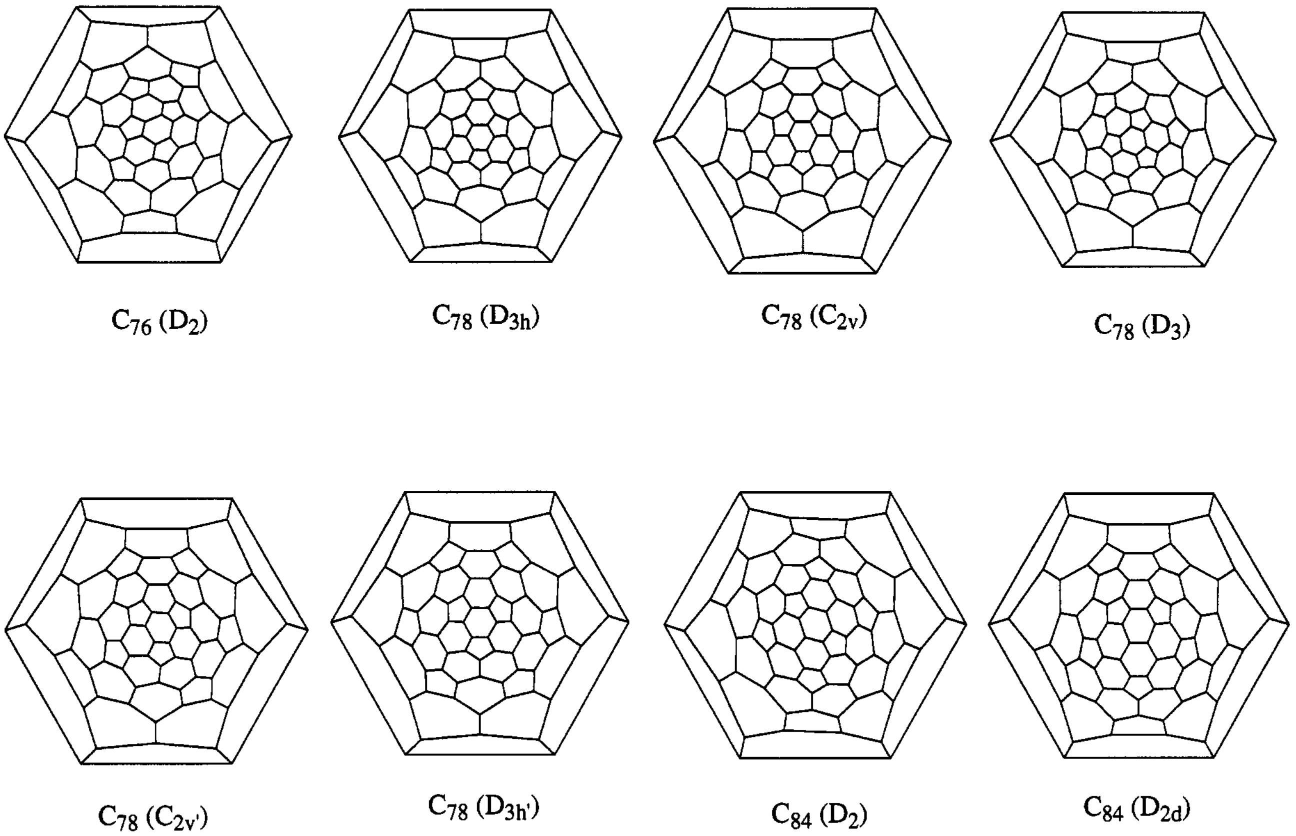
(1) the Greek prefix for (N/2 + 1);

(2) the modifier "cyclo";

(3) in square brackets, the numbers of CH groups on the two halves of the main ring, not counting the two bridgeheads (in all cases examined here, these numbers are both equal to $N/2-1$); the CH groups are numbered from 1 to N starting with one of the bridgeheads;

(4) in continuation, the number of CH groups in the main bridge (in all cases examined here, this being zero);

(5) in continuation, separated by periods like all symbols in square brackets, the numbers of CH groups in the secondary bridges, having superscripts separated by a comma, the labels of the two CH groups to which each bridge is anchored (in all cases considered here, these are zeros with



C_{76} (D_2) C_{78} (D_{3h}) C_{78} (C_{2v}) C_{78} (D_3)

C_{78} ($C_{2v'}$) C_{78} ($D_{3h'}$) C_{84} (D_2) C_{84} (D_{2d})

Figure 1. Schlegel diagrams of the few best known fullerenes, chosen as in the ref 11. C_{60} and C_{70}, which are widely known, are omitted.

Table 1. Essential Parts of IUPAC Names of Fullerene Isomers in Figure 1[a]

C_{60}	29	2	3	4	5	6	7	8	9	11	13	15	16	17	19	22	23	26	27	29	32	33...
	29	14	12	59	10	58	55	53	21	20	18	30	28	25	24	52	50	49	47	45	44	60...
			...34	35	36	37	38	39	40	42												
			...57	43	56	41	54	51	48	46												
C_{70}	34	2	3	4	5	7	8	9	11	12	13	14	15	16	17	18	19	20	21	22	24	25...
	34	6	34	31	10	69	63	61	30	60	58	29	56	27	54	26	23	53	40	37	35	33...
			...28	38	39	41	42	43	44	45	46	47	49	50	64							
			...32	70	67	52	66	51	48	65	62	59	57	55	68							
C_{76} (D_2)	37	2	3	4	5	7	8	9	11	12	14	15	16	18	19	20	22	23	24	26	27	29...
	37	6	37	13	10	75	73	67	66	17	36	34	21	65	63	25	33	31	28	62	59	57...
			...30	32	35	38	40	41	42	43	44	45	46	48	50	61	64	68				
			...55	53	52	51	49	76	47	74	71	60	58	56	54	70	69	72				
C_{78} (D_{3h})	38	2	3	4	5	7	8	9	10	12	14	15	17	18	20	21	24	26	27	30	31	33...
	38	6	38	16	13	11	78	75	25	23	22	19	37	34	32	29	28	74	55	54	51	49...
			...35	36	39	41	42	43	46	47	50	52	53	56	58	59	60	61	71			
			...48	45	44	77	70	68	67	65	64	63	57	73	62	72	69	66	76			
C_{78} (C_{2v})	38	2	3	4	5	7	8	9	11	12	14	15	17	18	20	21	23	25	26	27	28	32...
	38	6	16	13	10	39	37	31	30	24	22	19	78	75	73	71	69	29	68	54	33	36...
			...34	35	38	41	42	43	44	48	50	51	55	57	58	61	62	64	66			
			...53	47	46	45	77	60	49	52	59	56	67	65	63	76	74	72	70			
C_{78} (D_3)	38	2	3	4	5	7	8	9	11	12	14	15	17	18	20	21	22	23	27	29	30	33...
	38	6	16	13	10	39	36	34	32	26	25	19	78	75	24	74	67	28	31	66	53	52...
			...35	37	38	41	42	43	44	47	49	51	54	55	58	61	64	68	72			
			...50	48	46	45	77	71	60	59	57	56	65	63	62	70	69	73	76			
C_{78}($C_{2v'}$)	38	2	3	4	5	7	8	9	11	12	14	15	17	18	21	22	24	26	28	29	30	31...
	38	6	38	13	10	77	74	20	19	16	37	27	25	23	73	66	64	62	36	61	35	59...
			...32	33	34	42	43	44	45	46	49	51	52	53	54	55	56	67	71			
			...48	41	39	47	78	76	70	50	58	69	57	68	65	63	60	72	75			
C_{78} ($D_{3h'}$)	38	2	3	4	5	7	8	9	10	12	14	15	17	18	19	20	21	22	23	24	26	27...
	38	6	38	16	13	11	78	75	62	60	59	56	37	55	36	25	54	51	49	28	35	32...
			...29	30	31	33	34	42	44	45	47	50	52	53	58	61	63	64	65			
			...48	46	43	41	39	77	76	73	71	70	68	57	67	66	74	72	69			
C_{84} (D_2)	41	2	3	4	5	7	8	9	11	12	14	15	17	18	20	21	23	25	27	29	30	32...
	41	6	16	13	10	42	31	28	26	24	22	19	84	81	79	76	75	73	71	70	34	41...
			...33	35	36	37	39	40	45	46	48	50	52	53	54	56	59	60	62	64	65	67
			...38	69	51	49	47	44	83	58	57	55	68	66	63	61	82	80	78	77	74	72
C_{84} (D_{2d})	41	2	3	4	5	7	8	9	11	12	14	15	17	18	20	21	23	25	26	29	30	32...
	41	6	16	13	10	42	31	28	27	24	22	19	84	81	79	76	75	73	71	70	34	41...
			...33	35	36	37	39	40	45	46	48	50	52	53	54	56	59	60	62	64	65	67
			...38	69	51	49	47	44	83	58	57	55	68	66	63	61	82	80	78	77	74	72
C_{380}											?											

[a] Following the symbol of an isomer are the sizes of the two sides of the main ring, followed by the two-line row with zero superscripts denoting positions of other (zero-atom) bridges. The superscripts are written in pairs placed one above the other.

Chart 1. Pseudocode of the Main Procedure

```
begin
   choose any vertex as the starting vertex V;
   PATHPTR(1):=0; NGBPTR(1):=0; PATH(1):=V;
   choose any neighbor of V as the next vertex U;
   PATHPTR(2):=1; NGBPTR(2):=0; PATH(2):=U;
   NGBHD(1):=either CONT(U,V); BTM=2;
   ENDS:=CONT(U,V); ENDFLAG:=false;
   TRACK:=2; LEN:=2; PTR:=1;

   while TRACK > 0 do
       begin
           CHOICE:=CONT(PATH(LEN-1),PATH(LEN));
           if either CHOICE is in PATH(1:LEN) then
               choose the other CHOICE as NEXT
           else
               if both CHOICE are in NGBHD(BTM:PTR) then BACKTRACK
               else
                   if either CHOICE is in NGBHD(BTM:PTR), say V, then
                       if V is in ENDS then BACKTRACK
                       else
                               begin
                                   choose V as NEXT;
                                   increment PTR;
                                   NGBHD(PTR):=the other CHOICE;
                               end
                       else
                           begin
                               choose any CHOICE as NEXT;
                               increment TRACK;
                               PATHPTR(TRACK):=LEN;
                               NGBPTR(TRACK):=PTR;
                               increment PTR
                               NGBHD(PTR):=the other CHOICE;
                           end;

               increment LEN; PATH(LEN):=NEXT;
               if LEN=N then EXAMINE_FOR_THE_NAME;

               if NEXT is in ENDS then
                   if ENDFLAG=true then decrement LEN; BACKTRACK
                   else ENDFLAG:=true;
           end;
    end.


       PROCEDURE BACKTRACK

       begin
           PTR:=NGBPTR(TRACK)+1;
           NEXT:=NGBHD(PTR);
           NGBHD(PTR):=PATH(PATHPTR(TRACK)+1);
           for I:=PATHPTR(TRACK)+1 until LEN do
               if PATH(I) is in ENDS then ENDFLAG:=false;
           LEN:=PATHPTR(TRACK); TRACK:=TRACK-1;
           if TRACK=1 then BTM=1;
       end.
```

superscripts indicated pairwise in Table 1). The square bracket is then closed;

(6) the Greek name for N;

(7) the ending "-ane".

For a corresponding fullerene C_N, the name is identical through item 6, but item 7 is replaced by the following:

(7a) the location of the $N/2$ double bonds, indicated by the smaller numbering of the two endpoints of a double bond on the Hamiltonian circuit or by the numberings of both endpoints if these numberings do not differ by one unit;

(7b) the Greek name for $N/2$;

(7c) the ending "-ene".

Of course, there are in general many possible Kekulé structures with many possible ways for locating double bonds. E.g., there are[17] 12 500 Kekulé structures for buckminsterfullerene, and, more generally, given any Hamiltonian circuit, one immediately has three: two corresponding to the different alternating patterns around the circuit and one more corresponding to making all those bonds not in the circuit double. Even more generally (without assumption of Hamiltonian circuits), it can be shown[18] that every cubic planar graph has at least three Kekulé structures. Including positions of double bonds for nomenclature purposes into what are known to be appreciably (but not completely) delocalized systems is a matter of convention. In the planar aromatic series, the word "annulenes" has solved this problem for monocyclics obeying the Hückel rule, but in the corresponding ions or heterocycles, usual names ascribe localized positions for double bonds that are known to be largely delocalized. The reactivity of buckminsterfullerene indicates that the bonds exocyclic to five-membered rings have a pronounced double bond character, and one can safely assume this to be true for all isolated-pentagon fullerenes, which are expected to be more stable than their abutted-pentagon isomers. Therefore, this "chemical choice" for positions of double bonds should prevail: for isolated-pentagon fullerenes, the number of double bonds exocyclic to five-membered rings should be maximal, and whenever more than one such Kekulé structure is possible, the positions of the double bonds should be determined by the lexicographically smallest name. For the abutting-pentagon fullerenes, only the lexicographically smallest name should be taken into account. Thus, the full name for buckminsterfullerene[6,7] (granted the chemical choice for the location of double bonds) is hentriacontacyclo[29.29.0.$0^{2,14}$.$0^{3,12}$.$0^{4,59}$.$0^{5,10}$.$0^{6,58}$.$0^{7,55}$.$0^{8,53}$.$0^{9,21}$.$0^{11,20}$.$0^{13,18}$.$0^{15,30}$.$0^{16,28}$.$0^{17,25}$.$0^{19,24}$.$0^{22,52}$.$0^{23,50}$.$0^{26,49}$.$0^{27,47}$.$0^{29,45}$.$0^{32,44}$.$0^{33,60}$.$0^{34,57}$.$0^{35,43}$.$0^{36,56}$.$0^{37,41}$.$0^{38,54}$.$0^{39,51}$.$0^{40,48}$.$0^{42,46}$]hexaconta-1,3,5(10),6,8,11,-13(18),14,16,19,21,23,25,27,29(45),30, 32(44),33,35(43),36,-38(54),39(51),40(48),41,46,49,52,55,57,59-triacontaene.

For the famous C_{380} fullerene[19] depicted in Figure 2 the IUPAC name could not be derived owing to our computer limitations, though it is known[9] to possess Hamiltonian circuits.

3. ALTERNATIVE WAYS OF SPECIFYING FULLERENES

In what other way could one specify a fullerene molecule? The first code theorematically guaranteed to be valid for planar three-connected graphs was devised by L. Weinberg.[20] Fullerenes belong to this class of graphs, which is characterized (1) by the possibility to draw the graph in a plane without crossing lines and (2) by the necessity to break at least three edges in order to decompose the graph into separate components. A planar drawing of such a graph is called a Schlegel diagram, and we shall use it in the following text to illustrate some possibilities for coding of fullerenes. However, none of the methods depends on such a representation which can be drawn in many various ways. Actually, the only requirement is embeddability of a fullerene graph into a surface of a 3D-sphere, what is granted by definition. All the described algorithms need as input only the adjacency list with neighbors listed in a uniform manner (e.g., in a clockwise direction). Let us note that the described method for deriving the IUPAC name does not need anything more

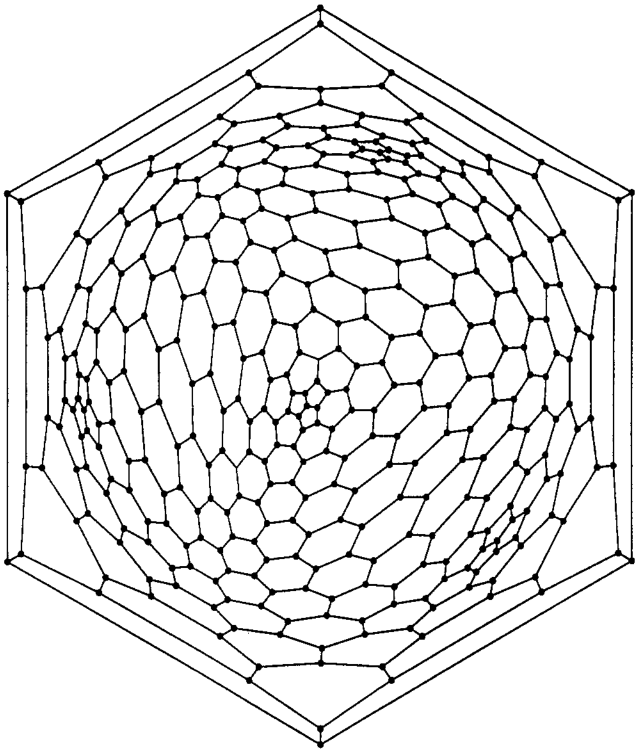
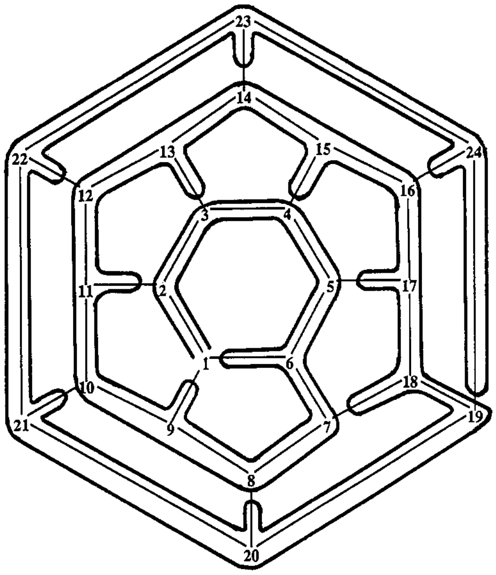


Figure 2. The C_{380} fullerene isomer with no ring spiral.[19]

than an ordinary adjacency list, that is, no embedding is necessary. However, note that some coding schemes discussed in the following text allow one to get insight into the symmetry properties of the fullerene.[20] The Weinberg procedure is based on an Eulerian cycle through the graph. An Eulerian cycle is a closed path passing through each edge exactly once. In the Weinberg procedure each edge is understood as a pair of oppositely directed edges—this ensures that the Eulerian path exists. The graph is drawn in a plane so that no two edges are crossing. The Eulerian cycle is started from any vertex by any of its incident edges. The starting vertex gets the label "1", and when a new, not yet visited vertex is reached, it gets the label greater by one then the last used label. When the trail arrives at vertex v, the choice of the next edge depends upon whether the vertex v is a new or an old (already visited) vertex. If v is a new vertex, the trail is continued by the rightmost edge with respect to the incoming edge. If v is an old vertex, the path turns back along the incoming edge, if it was not already traversed in the opposite direction. If this is not possible, the rightmost free edge is chosen. An example of an Eulerian cycle on the fullerene C_{24} graph performed in the described way, together with the corresponding Eulerian code, is depicted in Figure 3.

The code is obtained from the Eulerian cycle by writing down the vertex labels in the order in which the vertices are visited in the cycle. Cycles starting at different vertices and edges generally produce different codes. A unique code can be obtained by (deliberately) choosing lexicographically the smallest one. The time for coding is proportional to the squared number of vertices in the fullerene graph.[20] However, since the described method is developed for every planar three-connected graph, it might be unnecessarily robust for coding of only fullerene graphs, which form quite a special subclass. Besides, the relation of the Weinberg code to the actual structure is not transparent, and the chemists' wish for a "better" (e.g., shorter) code is easy to understand.



Eulerian code:

1.2.3.4.5.6.1.6.7.8.9.1.9.10.11.2.11.12.13.3.13.14.15.4.15.16.17.
5.17.18.7.18.19.20.8.20.21.10.21.22.12.22.23.14.23.24.16.24.19.
24.23.22.21.20.19.18.17.16.15.14.13.12.11.10.9.8.7.6.5.4.3.2.1

Figure 3. A simple fullerene C_{24} drawn with an Eulerian cycle according to the Weinberg algorithm[20] and the corresponding Eulerian code. The code is not lexicographically the smallest.

The first practical coding of fullerenes appeared shortly after their successful preparation.[21] It is based on the assumption that every fullerene can be viewed as a spiral arrangement of pentagons and hexagons.[22] This hypothesis was later shown to be false by the same authors[19] who found the counterexample shown in Figure 2, but nevertheless it has served well for isomer generation of smaller fullerenes (up to 100 carbon atoms).[23] The code is produced by writing down the sizes of rings as they are ordered in the spiral. As the spiral can start from various rings, producing thus different codes for the same graph, one must choose a distinct one to obtain a unique name. Again, the lexicographically smallest code can serve. Decoding is straightforward and produces immediately an acceptable drawing of the molecular structure. Coding can be accomplished by hand, but this is not suitable for everyday work, and the use of a computer is necessary. Although it is not generally applicable (not every fullerene possesses a ring spiral,[19] see Figure 2), this code will probably remain in at least informal use whenever possible. So far it is the most elegant code for fullerenes. The names of the fullerene isomers from Figure 1 derived in the above described way are given in Table 2. The appropriate spiral for C_{60} is drawn in Figure 4.

Table 2. Ring Spiral Codes of Fullerene Isomers from Figure 1

C_{60}	56666656565656566656565656566666665
C_{70}	5666665656565656566666666666565656565
C_{76} (D_2)	566666565656566665666666665666656565656
C_{78} (D_{3h})	566666565656665665666566566666666566656565
C_{78} (C_{2v})	5666665656565666666666656656656566656565656
C_{78} (D_3)	5666665656565666666566665665666565656565666
C_{78} ($C_{2v'}$)	5666665656566566666666566656565666566666565
C_{78} ($D_{3h'}$)	5666665656656566666665666656565666566666565
C_{84} (D_2)	566666566566566665666566565666566566656666665
C_{84} (D_{2d})	566666566566566665666566565666566666665665656
C_{380}	–

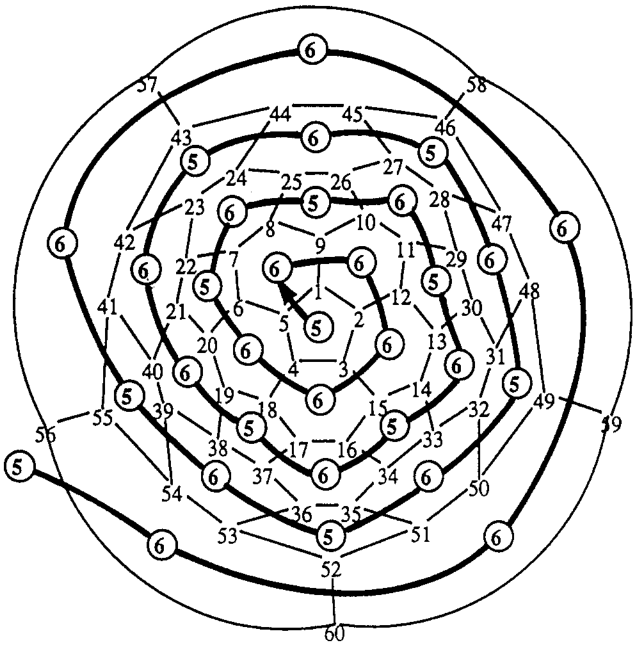


Figure 4. A graph of buckminsterfullerene with the ring spiral corresponding to the code in Table 2. The numbering of vertices corresponds to the code given in Table 4.

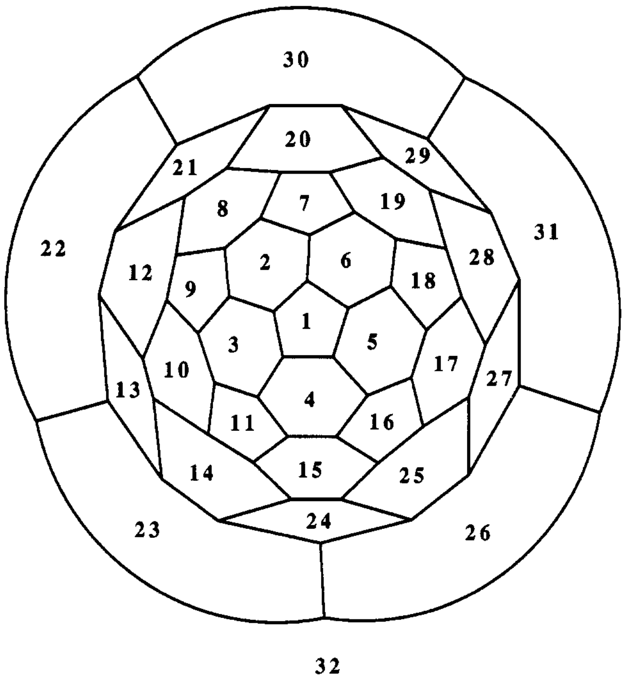


Figure 5. A graph of buckminsterfullerene with rings numbered in the order they appear in the code given in Table 3.

Another way of coding occurred as accompanying an alternative method for the generation of fullerene isomers.[24,25] The algorithm recursively adds ring by ring so that in each stage there is a "compact" buckydisk with a single boundary (represented by a single cycle). The algorithm starts with the simplest buckydisk: a pentagon or a hexagon. Whenever the new ring can be both a pentagon and a hexagon, the two options are separately examined. From the Brugesser–Mani theorem[26] it follows that this procedure can produce every fullerene isomer. However, the precise location of a boundary segment onto which the new ring will be adjoined might be important and not randomly chosen. In this algorithm the position for the new ring was chosen among only the longest boundary segments. After a new ring is added by bridging the two boundary vertices which terminate the chosen segment, an updated boundary is searched for the longest segment. The search starts at the segment containing the left of the two vertices bridged in the preceding step and proceeds to the right along the boundary. Whenever the longer segment is met, its length and position are saved. After the search is completed, the next ring is added at the segment whose position was last saved. Thus in each stage the position of the new ring is known in advance, similarly as in the spiral ring algorithm.[22] Therefore the isomer may be coded by simply writing down the ring sizes in the order they were added. Once again, different codes are possible for the same structure, depending on the ring from which the construction starts. The common solution by taking lexicographically the smallest one was applied here as well. The names obtained for the fullerenes from Figures 1 and 2 are given in Table 3, and the order in which rings are enlisted in the code of C_{60} is denoted by numbers in Figure 5.

The next code can be also related to another algorithm for the generation of fullerene isomers[27,28] which is similar to the preceding one but it does not use any assumption about the boundary segment onto which the new ring is added. Instead, each possible position is separately tried. This makes the algorithm safe, as no fullerene isomer can escape this procedure. But a price is paid in high redundancy: each isomer is produced many times and many unproductive paths are also repeatedly examined. Nevertheless, the algorithm was successfully applied for generating fullerene isomers with up to 70 carbon atoms[29] as well as for isomers with no abutting pentagons up to 96 carbon atoms[30a] after some initial difficulties.[30bc] The results confirmed those obtained by other algorithms.

The code derived from this algorithm turns out to be remarkably efficient. It could be applied even to large

Table 3. Codes of the Fullerene Isomers from Figures 1 and 2 Derived from the Algorithm for Generation of Fullerene Isomers Reported in ref 24

C_{60}	56666656565656656566566566565665
C_{70}	5666665656565666566656665665666565665
C_{76} (D_2)	5666665656565666566666656665665666565665
C_{78} (D_{3h})	56666656565656665666666656665665666565665
C_{78} (C_{2v})	56666656565656665666666656665665666565665
C_{78} (D_3)	56666656565656665666666666566566566565665
C_{78} ($C_{2v'}$)	56666656565666665666656565666566656666565
C_{78} ($D_{3h'}$)	56666656566665656666565666656566656656665
C_{84} (D_2)	56666656656656666656566666566566565666656665
C_{84} (D_{2d})	56666656656656666656566666566656565666566665
C_{380}	55566666666666666666666666666666666666666666666666666 66666666666666666666666666666666666666666666666666666 66666666666666666666666666666655656666666666666666666 666666666556566666666666666666555

fullerenes with as many as 380 atoms without difficulty. To obtain the code, the vertices introduced in the new ring are numbered in a fixed way. We chose to do this by orienting the arc so that always a clockwise-oriented boundary is produced. Any new vertices are numbered in the order they appear along the arc while going from its beginning toward its end. It is important to verify that the ring is simply connected to the boundary, that is, the added ring must be closed by the continuous boundary segment and a single arc. The code is produced by writing down the label of the initial vertex of the arc followed by the size of the added ring, in the order it was done during the generation process. The efficiency of this coding scheme depends on the way one chooses the unique name among all possible codes which may be generated. It is important that the buckydisk present in each stage be as compact as possible, becauses this diminishes the possibility of getting stuck during the generation process. If one prefers lexicographically the least code, this leads to an irregular growth with many returns from blind alleys. On the contrary, lexicographically the greatest code *with respect to the initial vertex labels* was quite efficient and was successfully applied even to C_{380}. *In respect to ring sizes*, there was no difference in usage of lexicographically the smallest and the greatest codes In order to keep the relation to the two previous codes as close as possible, when there was a choice, we preferred the lexicographically smaller code *with respect to the ring sizes*. The number of different codes is huge, but the algorithm showed a surprising efficiency due to the "learning" capability. Namely, the algorithm keeps only the "best" name found so far, and the paths leading to valid but "worse" codes (in the lexicographical sense) can be eliminated early. Note that the first ring is added to an "empty" buckydisk and that the initial vertex of the arc introduced by the next ring can always be that labeled by the largest number. Therefore for the first two rings their sizes are implicit as well as for the last ring which is automatically closed by the preceding one. Table 4 shows the names obtained for fullerenes from Figures 1 and 2. Comparison of the ring-size sequences in Tables 2 and 4 reveals identical order in both codes. Figure 4 shows the numbering of vertices in C_{60} obtained by the above coding. The order in which the rings are added is the same as in the spiral ring code.

The next two ways of naming fullerenes were not derived from the algorithms for generation of isomers. The first one is based on the assumption that a fullerene graph always possesses a Hamiltonian cycle (an H-cycle) and the second one on the assumption that a fullerene graph possesses a Hamiltonian path (an H-path). A Hamiltonian path also passes through each vertex of a graph exactly once, but there is no mandatory adjacency of the first and last vertices as in the case of an H-cycle.

The codes are derived in the following way. First, the graph is drawn without crossing lines on a plane (this is also called a Schlegel diagram), and a Hamiltonian cycle is outlined. This cycle divides the fullerene graph into two separated areas. Choose one of them as the "inside" of the cycle and the other as the "outside". On each vertex, an edge not included in the cycle points either to inside or to outside of the cycle. Go along the cycle and code each inside edge by "0" and each outside edge by "1". The code obtained in this way is N digits long, with N denoting the number of vertices in the graph. Depending on the starting

Table 4. Codes of the Fullerene Isomers from Figures 1 and 2 Derived from the Safe Algorithm for Generation of Fullerene Isomers[27]

C_{60}	$5\ 6\ 9^6\ 12^6\ 15^6\ 18^6\ 20^5\ 22^6\ 25^5\ 26^6\ 29^5\ 30^6\ 33^5\ 34^6\ 37^5\ 38^6\ 40^6\ 42^5\ 44^6\ 45^5\ 47^6\ 48^5\ 50^6\ 51^5\ 53^6\ 54^5\ 55^6\ 57^6\ 58^6\ 59^6\ 60^6\ 5$
C_{70}	$5\ 6\ 9^6\ 12^6\ 15^6\ 18^6\ 20^5\ 22^6\ 25^5\ 26^6\ 29^5\ 30^6\ 33^5\ 34^6\ 37^5\ 38^6\ 40^6\ 42^6\ 45^6\ 46^6\ 49^6\ 50^6\ 53^6\ 54^6\ 57^6\ 58^6\ 60^5\ 61^6\ 64^5\ 63^6\ 66^5\ 65^6\ 68^5\ 67^6\ 70^5\ 69^6\ 5$
C_{76} (D_2)	$5\ 6\ 9^6\ 12^6\ 15^6\ 18^6\ 20^5\ 22^6\ 25^5\ 26^6\ 29^5\ 30^6\ 33^5\ 34^6\ 37^6\ 39^6\ 41^6\ 43^5\ 45^6\ 46^6\ 49^6\ 50^6\ 53^6\ 54^6\ 57^6\ 59^5\ 60^6\ 62^6\ 64^6\ 65^6\ 68^5\ 67^6\ 70^5\ 69^6\ 72^5\ 73^6\ 74^5\ 75^6\ 76^5\ 6$
C_{78} (D_{3h})	$5\ 6\ 9^6\ 12^6\ 15^6\ 18^6\ 20^5\ 22^6\ 25^5\ 26^6\ 29^5\ 30^6\ 33^5\ 34^6\ 37^6\ 39^6\ 41^6\ 43^6\ 46^6\ 47^5\ 49^6\ 50^6\ 53^6\ 54^5\ 56^6\ 58^6\ 60^5\ 61^6\ 63^6\ 66^6\ 67^6\ 68^6\ 71^6\ 72^6\ 74^5\ 75^6\ 76^5\ 73^6\ 70^5\ 69^6\ 5$
C_{78} (C_{2v})	$5\ 6\ 9^6\ 12^6\ 15^6\ 18^6\ 20^5\ 22^6\ 25^5\ 26^6\ 29^5\ 30^6\ 33^5\ 34^6\ 37^6\ 39^6\ 41^6\ 43^6\ 46^6\ 47^6\ 50^6\ 51^6\ 54^6\ 55^5\ 57^6\ 59^6\ 61^5\ 62^6\ 64^6\ 67^5\ 66^6\ 69^5\ 68^6\ 71^6\ 72^6\ 74^5\ 75^6\ 76^5\ 77^6\ 78^5\ 6$
C_{78} (D_3)	$5\ 6\ 9^6\ 12^6\ 15^6\ 18^6\ 20^5\ 22^6\ 25^5\ 26^6\ 29^5\ 30^6\ 33^5\ 34^6\ 37^6\ 39^6\ 41^6\ 43^6\ 46^6\ 47^5\ 49^6\ 50^6\ 53^6\ 54^6\ 57^5\ 58^6\ 60^6\ 62^5\ 63^6\ 66^6\ 67^6\ 68^5\ 70^6\ 71^5\ 73^6\ 74^5\ 75^6\ 76^5\ 78^6\ 77^6\ 6$
C_{78} ($C_{2v'}$)	$5\ 6\ 9^6\ 12^6\ 15^6\ 18^6\ 20^5\ 22^6\ 25^5\ 26^6\ 29^5\ 30^6\ 33^6\ 35^5\ 37^6\ 39^6\ 41^6\ 43^6\ 46^6\ 47^6\ 50^6\ 51^5\ 53^6\ 55^6\ 57^6\ 59^5\ 60^6\ 62^5\ 63^6\ 66^5\ 65^6\ 68^6\ 69^6\ 71^5\ 72^6\ 74^6\ 75^6\ 76^6\ 78^5\ 77^6\ 5$
C_{78} (D_{3h}')	$5\ 6\ 9^6\ 12^6\ 15^6\ 18^6\ 20^5\ 22^6\ 25^5\ 26^6\ 29^6\ 31^5\ 33^6\ 35^5\ 37^6\ 39^6\ 41^6\ 43^6\ 46^6\ 47^6\ 50^5\ 51^6\ 53^6\ 55^6\ 57^6\ 59^5\ 60^6\ 62^5\ 63^6\ 66^5\ 65^6\ 68^6\ 69^6\ 71^5\ 72^6\ 74^6\ 75^6\ 76^6\ 78^5\ 77^6\ 5$
C_{84} (D_2)	$5\ 6\ 9^6\ 12^6\ 15^6\ 18^6\ 20^5\ 22^6\ 25^6\ 27^5\ 29^6\ 31^6\ 34^5\ 35^6\ 38^6\ 40^6\ 42^6\ 44^5\ 46^6\ 48^6\ 50^6\ 52^5\ 53^6\ 56^6\ 57^5\ 59^6\ 61^5\ 62^6\ 64^6\ 66^6\ 68^5\ 69^6\ 71^6\ 72^5\ 74^6\ 75^6\ 77^6\ 78^5\ 79^6\ 81^6\ 82^6\ 83^6\ 84^6\ 5$
C_{84} (D_{2d})	$5\ 6\ 9^6\ 12^6\ 15^6\ 18^6\ 20^5\ 22^6\ 25^6\ 27^5\ 29^6\ 31^6\ 34^5\ 35^6\ 38^6\ 40^6\ 42^6\ 44^5\ 46^6\ 48^6\ 50^6\ 52^5\ 53^6\ 56^6\ 57^5\ 59^6\ 61^5\ 62^6\ 64^6\ 66^6\ 68^5\ 69^6\ 71^6\ 72^6\ 75^6\ 76^6\ 78^6\ 79^5\ 80^6\ 82^6\ 83^5\ 81^6\ 84^5\ 6$
C_{380}	$5\ 5\ 8^5\ 10^6\ 13^6\ 16^6\ 18^6\ 21^6\ 23^6\ 25^6\ 28^6\ 30^6\ 33^6\ 35^6\ 37^6\ 40^6\ 42^6\ 44^6\ 46^6\ 49^6\ 51^6\ 53^6\ 56^6\ 58^6\ 60^6\ 62^6\ 65^6\ 67^6\ 69^6\ 71^6\ 73^6\ 76^6\ 78^6\ 80^6\ 82^6\ 85^6\ 87^6\ 89^6\ 91^6\ 93^6\ 96^6\ 98^6\ 100^6\ 102^6\ 104^6\ 106^6\ 109^6\ 111^6$ $113^6\ 115^6\ 117^6\ 120^6\ 122^6\ 124^6\ 126^6\ 128^6\ 130^6\ 133^6\ 135^6\ 137^6\ 139^6\ 141^6\ 143^6\ 145^6\ 148^6\ 150^6\ 152^6\ 154^6\ 156^6\ 158^6\ 161^6\ 163^6\ 165^6\ 167^6\ 169^6\ 171^6\ 173^6\ 176^6\ 178^6\ 180^6\ 182^6\ 184^6\ 186^6\ 188^6\ 190^6$ $193^6\ 195^6\ 197^6\ 199^6\ 201^6\ 203^6\ 205^6\ 208^6\ 210^6\ 212^6\ 214^6\ 216^6\ 218^6\ 220^6\ 222^6\ 225^6\ 227^6\ 229^6\ 231^6\ 233^6\ 235^6\ 237^6\ 239^6\ 241^6\ 244^6\ 246^6\ 248^6\ 250^6\ 252^6\ 254^6\ 256^6\ 258^6\ 261^6\ 263^6\ 265^6\ 267^6$ $269^6\ 271^6\ 273^6\ 275^6\ 277^6\ 280^6\ 282^6\ 284^6\ 286^6\ 288^6\ 290^6\ 292^6\ 294^6\ 296^6\ 298^6\ 301^5\ 302^5\ 303^6\ 305^6\ 307^6\ 309^6\ 311^6\ 313^6\ 315^6\ 318^6\ 320^5\ 321^5\ 322^6\ 324^6\ 326^6\ 328^6\ 330^6\ 332^6\ 334^6\ 337^6\ 339^5$ $340^5\ 341^6\ 343^6\ 345^6\ 347^6\ 349^6\ 351^6\ 353^6\ 352^6\ 356^6\ 360^6\ 364^6\ 363^6\ 371^6\ 359^6\ 350^6\ 348^6\ 346^6\ 338^5\ 336^6\ 335^6\ 333^6\ 331^6\ 329^6\ 327^6\ 319^5\ 317^6\ 316^6\ 314^6\ 312^6\ 310^6\ 308^6\ 306^6\ 304^6\ 5$

Table 5. Codes of Fullerene Isomers in Figure 1 Derived from Their Hamiltonian Cycles[a]

C_{60}	1090	000010010010101010101101101111000100101001100110011010110111
C_{70}	2790	0000100100101001011010101111010000110010110011000010111101010010110111
C_{76} (D_2)	6643	0000100100100101010101011011011001111010000101010010110011001100110110101111
C_{78} (D_{3h})	8244	000010010010010101010101111010010100001011110101000010111100001100110101101111
C_{78} (C_{2v})	7300	000010010010010101010100110110110011110100001010010100110011001100110101101111
C_{78} (D_3)	6988	000010010010010101010101111001010100001011110101000010111100001101010110101111
C_{78} (C_{2v}')	8120	000010010010010101011001111010010100110000101111010110000111000011100110101111
C_{78} (D_{3h}')	7344	000010010010100101101011110100001100101111010100001010101100001110110010110111
C_{84} (D_2)	17622	000010010010010101010100111000011100111000011110100001010100110011001100110110101111
C_{84} (D_{2d})	15488	000010010010010101011001110010111100001010011001100001001110110011000011100110101111
C_{380}	?	?

[a] The numbers in the second column indicate the numbers of Hamiltonian cycles in the fullerene graphs.

Table 6. Codes of Fullerene Isomers in Figure 1 Derived from Their Hamiltonian Paths

C_{60}	0000100010100101001100101010101010101101010110101101101111
C_{70}	00001000100100101001100100110011001010101101010110110101101101011101
C_{76} (D_2)	00001000100100100100110010011001100110010101101010110110110011011010111101
C_{78} (D_{3h})	0000100010010010010010101001100110011001100110101001101101111000110001111011
C_{78} (C_{2v})	0000100010010010010010101001100110011001100110101001101101100110111010111101
C_{78} (D_3)	0000100010010010010011001001100110011001010110101001101101111000110000111101
C_{78} (C_{2v}')	0000100010010010010011001001100110011001100110101011011001101101101011101111
C_{78} (D_{3h}')	0000100010010010100110010011001100101010011100001111011000010011110110011011
C_{84} (D_2)	0000100010010010010011001001100110011001100110101001110000111001100110110111100011
C_{84} (D_{2d})	0000100010010010010011001001100110011001100110101011001110000111001101101011100111
C_{380}	?

vertex in the cycle, on the chosen definition of inside and outside of the cycle, and on the choice between two possible directions for tracing the cycle, different binary codes can be obtained by writing sequentially digits 0 and 1 encountered along the cycle. Take lexicographically the smallest number as the final code of the given cycle. Repeat the same operations for all Hamiltonian cycles and take again the lexicographically smallest among all these codes as the name of the fullerene.

In the case of an H-path, the procedure is similar. The path is traced on a planar drawing of a fullerene graph. Except for the starting and ending vertices, at each vertex an edge not included in the path points out to the left or to the right of the path. Each left-going edge is coded by "0" and each right-going edge by "1". The first and last vertices are omitted from this procedure—thence the code length is N-2 digits. The Hamiltonian path is thus coded into a sequence of 0's and 1's. Depending on the initial and final vertex and the choice of "left" and "right" sides, one gets four different codes for each path. The lexicographically smallest among all the codes of all paths represents the name of the fullerene.

It is not obvious that the codes based on an H-cycle or an H-path represent uniquely the fullerene. To prove this, place a vertex inside each face of the fullerene and connect those new vertices which lie on opposite sides of an edge NOT INCLUDED in the cycle or the path, respectively. The new graph thus obtained is composed of two treelike components when derived from the H-cycle (one component being inside and the other outside of the cycle), and it is a spanning tree of the fullerene dual in the case of the H-path. Now, consider the terminal vertices of these trees: the corresponding rings in the fullerene share all but one boundary edge with an H-cycle or an H-path, respectively. On an H-cycle and H-path these segments have three or four vertices in a sequence with all edges pointing onto the same side, say to the inside of the Hamiltonian cycle, or to the left of the Hamiltonian path. By bridging the two boundary vertices of such a sequence, a pentagon or hexagon is reconstructed. Recursive application of this procedure reconstructs the starting fullerene.

Table 5 contains the names of fullerenes exemplified in Figure 1 derived from their Hamiltonian cycles, and in Table 6 the names are derived from Hamiltonian paths. In neither of the two ways could we get the name for C_{380} (Figure 2). While this seems reasonable in the case of Hamiltonian cycles, it is somewhat surprising in the case of Hamiltonian paths since the algorithm also possesses a kind of "learning" capability. A superficial examination shows that the problem is with finding Hamiltonian paths. Higher fullerenes are complex structures in which Hamiltonian paths are difficult to find, especially when the algorithm is tuned for a systematic search for all Hamiltonian paths. Maybe a hybridization with some efficient heuristic algorithm for generating Hamiltonian paths could be helpful, since after some paths are found, the learning capability may become effective.

The Hamiltonian cycle corresponding to the code of C_{60} is drawn in Figure 6. The analogous Hamiltonian path is shown in Figure 7. It is interesting to note that the Hamiltonian cycle and the Hamiltonian path are wound into perfect spirals. However, this is not a general property. It seems that the criteria used for choosing a unique name somehow prefer a spiral configuration.

The last code discussed here has arisen as a byproduct of a new algorithm[31] for computation of matching polynomials, that is, in its application to fullerene isomers. The algorithm relies on a spanning tree of a given graph, such that the corresponding cotree has the smallest possible Hosoya index.[32] A spanning tree is a connected acyclic subgraph which contains all vertices of a graph. A cotree[33] is the subgraph spanned by the edges not included in the spanning tree. The Hosoya index of the cotree, defined as the sum of absolute values of coefficients of the matching polynomial, equals the number of terms which have to be processed in order to compute the matching polynomial of the graph. The

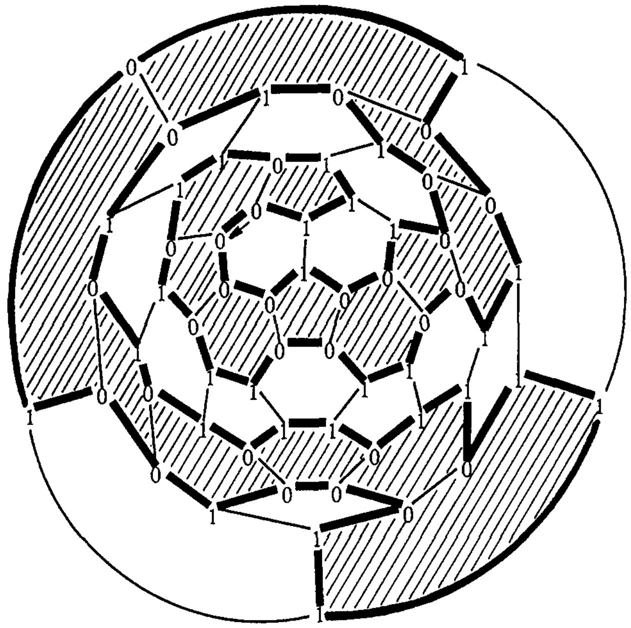


Figure 6. A graph of buckminsterfullerene with the Hamiltonian cycle corresponding to the code given in Table 5. Labels 0 and 1 denote vertices with inside and outside pointing edges, respectively. The inside of the cycle is shaded. A start of the code given in Table 5 is indicated by a small arrow.

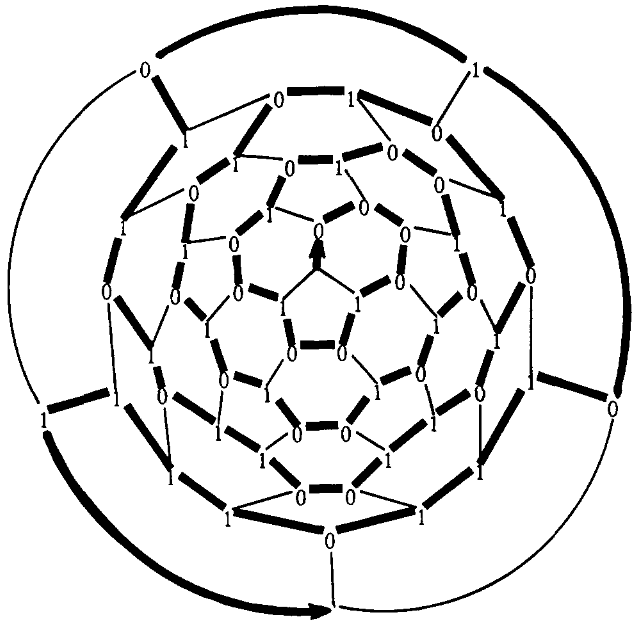


Figure 7. A graph of the buckminsterfullerene with the Hamiltonian path corresponding to the code given in Table 6. Starting and ending edges are indicated by arrows. Vertices at which edges point toward left and right are labeled by 0 and 1, respectively.

optimal cotree for computation of a matching polynomial of the fullerene is a path.[31] We have found that all graphs of fullerene isomers with up to 70 vertices possess such a path-cotree, and we conjecture that this is a general property for them.

The path-cotree may be used for coding of a fullerene in the following way, starting with its Schlegel diagram. Note that every ring (elementary cycle) in a fullerene graph contains at least one edge of a cotree (otherwise the spanning tree would not be a tree). One may establish the correspondence between the edges of a path-cotree and rings so that each edge is associated with exactly one ring, and vice versa, with the exception of a terminal edge (further considered as the "starting" edge) which is associated with both neighboring rings. The way of establishing correspondence is obvious by choosing only between neighboring rings while going from starting toward ending edge of the path. The code is started by writing down sizes of the two rings associated with the starting edge. By choice, first is written the size of the ring on the right side of the edge (which is imagined as oriented toward the rest of the path). For each subsequent edge, first is written down the size of the associated ring, and then the symbol of a side of the edge on which the ring is situated. By choice, the left side is coded by 1 and the right side by 0. The code is completed after the last edge is reached. Construction of the parent graph from the generated code is straightforward. One should only prove that the rings being added during the construction are only "simply" linked to the boundary of the already constructed part. Otherwise the boundary would split into two separate cycles and the part inside either of the two boundary components could not be created. However, this contradicts the described correspondence between rings and edges of the path-cotree and is to be abandoned.

As before there are many different possible path-cotrees and the representative one is chosen as lexicographically the smallest. One must take care that both possible choices of left and right in a given planar embedding of the graph are checked. The codes obtained for exemplary fullerene isomers from Figure 1 are given in Table 7. The symbols 1 and 0 are superscripted in order to visually distinguish them from ring sizes. Symptomatically, for the C_{380} isomer we could not find any path-cotree in a reasonable CPU time (several days on a DEC Station 5000) since due to the large size of the graph, only a tiny amount of possibilities has been searched through. The path-cotree corresponding to the code of C_{60} in Table 7 is shown in Figure 8. Note that the minimization criterion used for getting the canonical code resulted again in spiral winding of the path.

Table 8 shows times needed on a PC 486/66 MHz under MS DOS for deriving the names given in Tables 1−7. As far as the time requirements are concerned, all the names except those based on Hamiltonian cycles are acceptable. Obtaining the name for small N values from Hamiltonian paths needed also a small amount of time, but the time increases steeply after the size N passes approximately 100 vertices and soon surpasses any reasonable limit.

The space requirements are the smallest for the spiral-ring code and its modification: this needs only $n/2 + 2$ bits. However, there exist isomers for which this code is not applicable. In comparison to them, the Weinberg code needs a lot of space. Probably it could be compressed in some way, but this code is also remote from a chemist's view. An interesting compromise could be the code given in Table 4. It is safe, efficient, reasonably compact, and not so far from the chemist's approach.

The numbering of atoms is inherent to all these coding schemes or can be easily built up, and thus the names of addition derivatives are easy to obtain, e.g., for products of partial hydrogenation or halogenation. One could try to use some of these names (spiral ring name, or its modification, or those based on Hamiltonian cycle or path) as vectors characterizing the structure of isomers, for purposes of quantification of structural similarity. However, this does not seem promising due to the minimizing criterion used in deriving these names (lexicographically the smallest version)

Table 7. Codes of Fullerene Isomers in Figure 1 Derived from the Path-Cotrees

C_{60}	$566^05^06^05^06^16^05^06^16^05^06^16^05^16^06^15^06^16^05^16^06^15^16^06^15^16^06^15^16^15^0$
C_{70}	$566^05^06^05^06^16^05^06^16^05^06^16^05^16^06^16^06^15^06^16^05^06^16^16^05^06^16^16^05^06^16^16^05^16^15^0$
C_{76} (D_2)	$566^05^06^05^06^16^05^06^16^05^06^16^06^15^06^06^16^06^15^06^16^05^06^16^16^06^06^15^16^06^16^15^06^16^05^16^15^16^0$
C_{78} (D_{3h})	$566^05^06^05^06^16^05^06^16^05^06^16^06^16^06^06^15^06^16^16^05^06^16^05^06^06^16^15^16^06^16^16^06^06^15^16^15^06^05^0$
C_{78} (C_{2v})	$566^05^06^05^06^16^05^06^16^05^06^16^06^16^06^06^15^06^16^06^16^05^06^16^16^05^06^16^16^06^06^15^16^15^06^05^16^15^16^0$
C_{78} (D_3)	$566^05^06^05^06^16^05^06^16^05^06^16^06^15^06^06^16^06^15^06^16^06^06^15^16^06^16^15^06^06^16^15^06^16^05^16^15^16^06^0$
C_{78} $(C_{2v'})$	$566^05^06^05^06^16^06^06^15^06^06^15^06^16^05^06^16^06^15^16^16^05^16^06^15^16^06^06^16^06^15^06^06^05^16^16^15^06^16^0$
C_{78} $(D_{3h'})$	$566^05^06^05^06^16^05^06^16^06^06^15^06^16^05^06^16^06^15^16^16^05^16^06^15^16^06^06^15^06^16^06^06^05^16^16^15^06^16^0$
C_{84} (D_2)	$566^05^06^06^06^15^06^06^16^05^06^16^06^15^06^06^15^06^16^06^15^06^05^16^16^06^15^06^16^05^16^06^16^15^06^16^06^16^15^06^16^16^0$
C_{84} (D_{2d})	$566^05^06^06^06^15^06^06^16^05^06^16^06^15^06^06^15^06^16^06^16^06^05^16^16^06^15^06^16^05^16^06^16^15^06^16^06^15^16^16^06^05^0$
C_{380}	?

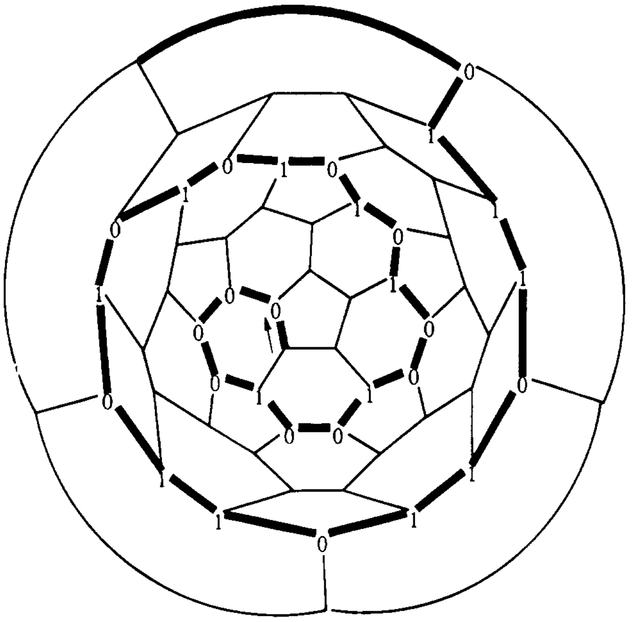


Figure 8. A graph of the buckminsterfullerene with the path-cotree corresponding to the code given in Table 7I. The label on the vertex denotes that the ring corresponding to the next edge is on the left (1) or right (0) side of the edge. It also corresponds to left (0) or right (1) turn of the path.

Table 8. CPU Times, in Seconds, Spent on PC 486/66 MHz under MS DOS, for Deriving the Names Given in Tables 1−7[a]

	table number						
	1	2	3	4	5	6	7
C_{60}	7	1	2	2	8	1	1
C_{70}	29	1	2	2	35	1	2
C_{76} (D_2)	84	1	3	2	96	1	1
C_{78} (D_{3h})	119	1	3	3	138	1	1
C_{78} (C_{2v})	118	1	2	2	133	2	2
C_{78} (D_3)	116	1	2	2	132	1	2
C_{78} $(C_{2v'})$	115	1	2	2	132	1	2
C_{78} $(D_{3h'})$	113	1	3	2	129	1	1
C_{84} (D_2)	321	1	3	2	365	1	2
C_{84} (D_{2d})	319	1	2	2	358	1	1
C_{380}	?		4	14	?	?	?

[a] Values are rounded off to the nearest integer. The Arabic numbers in the top row denote the table numbers.

which may put structurally identical parts onto different positions in two names and cause them to appear uncorrelated.

4. SPIRAL NUMBERING OF ATOMS IN FULLERENES

In a recent proposal from Chemical Abstracts Service, the atoms in fullerenes should be numbered along a spiral passing through all the atoms,[12] whenever possible. The spiral is defined as the Hamiltonian path which at every vertex, where a choice is possible, turns to the same side. Since there are two possible sides, we speak about two spiral directions or orientations. Here we wish to point out a few interesting examples. Two of them have high symmetry: T_d and I_h and no such spiral! They are depicted on Figure 9. Here C_{80} (I_h) is a generalized leapfrog derivative of dodecahedral C_{20}, with $a = 2$, $b = 0$, according to Sah's specification.[34] It has isolated pentagons, but it is already known as an open-shell isomer[35] and therefore is not expected to be of practical importance. On the other hand, the C_{56} (T_d) isomer is a closed shell compound, but has four triples of fused pentagons, a structural detail which is energetically unfavorable.[36] Thus both of these examples are not very real; however, they show that high symmetry does not guarantee the existence of a spiral passing through all atoms.

In order to verify the above finding, the reader has to try to construct a spiral running through all atoms. Due to the high symmetry of C_{80} (I_h), one needs to check only four different ways to begin a spiral. Let us denote vertices belonging to any of pentagons by P and those shared by three hexagons by H. Then the four different beginnings are (1) P−P−P−, (2) P−P−H−, (3) P−H−P−, and (4) H−P−P−. Similarly, in C_{56} (T_d) there are 14 different beginnings. Let us first label equivalent vertices. Label the four vertices shared by triples of pentagons by 1. All their neighbors should be labeled by 2. Proceed with labeling as yet unlabeled neighbors of m vertices with $m + 1$ until all vertices are labeled. The maximal label to be used is 5. The 14 beginnings to be checked are (1) 1−2−3−, (2) 2−3−3−, (3) 2−3−4−, (4) 2−1−1−, (5) 3−3−2−, (6) 3−3−4−, (7) 3−4−3−, (8) 3−4−5−, (9) 3−2−2−, (10) 3−2−1−, (11) 4−3−2−, (12) 4−3−3−, (13) 4−5−4−, and (14) 5−4−3−.

The third example, shown in Figure 10, has D_3 symmetry and is chiral. Its peculiar property is the existence of a spiral wound up in only one direction; this fact proves that one has to take care about both possible spiral orientations when searching for such spirals in chiral fullerenes. In order to check this, some more work is needed. First, let us again mark the different kinds of atoms. The vertices along the spiral marked in Figure 10 may be labeled in the following order (it is not important from which end one starts): 1 2 3 4 2 3 4 2 3 4 5 6 7 8 5 6 7 8 5 6 7 8 9 10 10 9 9 10 10 9 9 10 10 9 8 7 6 5 8 7 6 5 8 7 6 5 4 3 2 4 3 2 4 3 2 1. Since only clockwise orientations are searched (counterclockwise direction is given in Figure 10), the spiral is fixed by specifying first two vertices. Vertices labeled by 1 have three equivalent neighbors, while all others have neighbors of different type. This amounts to the total number of 28 different beginnings which have to be tried.

The shown examples are by no means isolated cases. In all fullerene isomers with up to 90 atoms, we have found 23

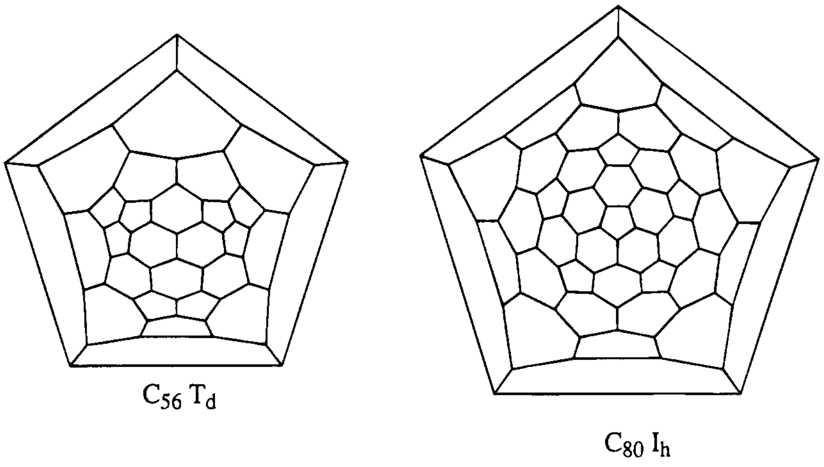


Figure 9. Two fullerene isomers with no atom spiral.

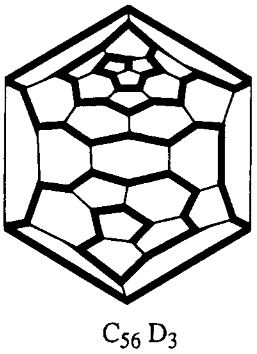


Figure 10. Fullerene isomer with atom spiral going in only one direction.

cases without a spiral path, and 98 cases with a spiral wound up in only one direction.

5. CONCLUSION

We have presented several schemes for specifying fullerene isomers. In the first part we have described the algorithm for obtaining the IUPAC name. In the second part, alternative schemes are given and compared in order to present a variety of possible codes, some of which might serve as a seed of a better way of naming fullerene isomers. They differ in their proximity to common chemical sense, universality of application, time needed for finding the canonical unique code, and space needed for storage. One of these coding schemes seems to be a compromise solution which could be of practical interest. In connection to the proposal for naming fullerenes by using a spiral passing through all the atoms, we have shown a few isomers with peculiar properties: two isomers with high symmetry and without such a spiral and one isomer with the spiral going in only one direction.

ACKNOWLEDGMENT

We acknowledge with thanks constructive comments made by Drs. A. L. Goodson and especially by C. Rücker. D.B. wants to thank Prof. Chih-Han Sah and Prof. Joseph Malkevitch for useful discussions. The support from the Ministry of Science, Technology and Informatics of the Republic of Croatia through Grant 1-07-159 is acknowledged. A.T.B. and D.J.K. acknowledge support from the Welch Foundation of Houston, Texas and from the donors of the Petroleum Research Fund, administered by the American Chemical Society.

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) Krätschmer, W.; Lamb, L. D.; Fostiropoulos, K.; Huffmann, D. R. Solid C_{60}: a new form of carbon. *Nature* **1990**, *347*, 354–358.

(3) Braun, T. Epidemic Spread of Fullerene Research. *Angew. Chem., Int. Ed. Engl.* **1992**, *31*, 588–589.

(4) Rigaudy J.; Klesney S. P. *IUPAC Nomenclature of Organic Chemistry, Sections A, B, C, D, E, F and H*; Pergamon Press: Oxford, 1979; pp 32–34.

(5) Castells, J.; Serratosa, F. Goal! An Exercise in IUPAC Nomenclature. *J. Chem. Educ.* **1983**, *60*, 941. Castells, J.; Serratosa, F. Replaying the ball: soccerane revisited. *J. Chem. Educ.* **1986**, *63*, 630.

(6) Rücker, C.; Rücker, G. Nomenclature of organic polycycles out of the computer—how to escape the jungle of the secondary bridges. *Chimia* **1990**, *44*, 116–120.

(7) Kroto, H. W., Allaf, A. W., Balm, S. P. C_{60}: Buckminsterfullerene. *Chem. Rev.* **1991**, *91*, 1213–1235.

(8) Eckroth, D. R. The Correct von Baeyer Name for (Buckminster)-fullerane. *J. Chem. Educ.* **1993**, *70*, 609–611.

(9) Balaban, A. T.; Babić, D; Klein, D. J. W. R. Hamilton, his circuits and IUPAC nomenclature for fulleranes. *J. Chem. Educ.*, in press.

(10) Papadimitriou, C. H.; Steiglitz, K. *Combinatorial Optimization—Algorithms and Complexity*; Prentice Hall: Englewood Cliffs, NJ, 1982; p 395.

(11) Taylor, R. C_{60}, C_{70}, C_{76} and C_{84}: Numbering, π-Bond Order Calculations and Addition Pattern Considerations. *J. Chem. Soc., Perkin Trans. II* **1993**, 813–824.

(12) At a recent meeting of the IUPAC Committee for Nomenclature of Organic Chemistry, it was decided that IUPAC would present both Taylor's and the Chemical Abstracts Service's nomenclature proposals, and that for the time being there would be no IUPAC fullerene nomenclature (Alan L. Goodson, Chemical Abstracts Service, private communication).

(13) Babić, D.; Trinajstić, N. On Assembling Fullerenes from Identical Fragments. *Fullerene Sci. Technol.* **1994**, *2*, 343–356.

(14) Goodey, P. R. A Class of Hamiltonian Poytopes. *J. Graph Theory* **1977**, *1*, 181–185.

(15) Malkevitch, J. Polytopal Graphs. In *Selected Topics in Graph Theory 3*; Beineke, L. W., Wilson, R. J., Ed.; Academic Press: New York, 1988; p 180.

(16) E-mail address of D. Babić: dbabic@olimp.irb.hr.

(17) Klein, D. J.; Schmalz, T. G.; Hite, G. E.; Seitz, W. A. Resonance in C_{60}, Buckminsterfullerene. J. *Am. Chem. Soc.* **1986**, *108*, 1301–1302.

(18) Klein, D. J., Liu, X. Theorems for carbon cages. *J. Math. Chem.* **1992**, *11*, 199–205.

(19) Manolopoulos, D. E.; Fowler, P. W. A fullerene without a spiral. *Chem. Phys. Lett.* **1993**, *204*, 1–7.

(20) Weinberg, L. A simple and efficient algorithm for determining isomorphism of planar triply connected graphs. *IEEE Trans. Circuit Theory* **1966**, *CT-13*, 142–148.

(21) Manolopoulos, D. E.; May, J. C.; Down, S. E. Theoretical studies of the fullerenes: C_{34} to C_{70}. *Chem. Phys. Lett.* **1991**, *181*, 105–111.

(22) Manolopoulos, D. E.; Fowler, P. W. Molecular graphs, point groups, and fullerenes. *J. Chem. Phys.* **1992**, *96*, 7603–7614.

(23) Fowler, P. W.; Manolopoulos, D. E. *An Atlas of Fullerenes*; Oxford University Press: Oxford, in press.

(24) Babić, D.; Trinajstić, N. Pyracylene rearrangement classes of fullerene isomers. *Computers Chem.* **1993**, *17*, 271–275.

(25) Babić D.; Klein, D. J.; Sah, C. H. Symmetry of fullerenes. *Chem. Phys. Lett.* **1993**, *211*, 235–241.

(26) Brugesser, H.; Mani. P. Shellable Decompositions of Cells and Spheres. *Math. Scand.* **1971**, *29*, 197–205.

(27) Liu, X.; Klein, D. J.; Schmalz, T. G.; Seitz, W. A. Generation of carbon-cage polyhedra. *J. Comput. Chem.* **1991**, *12*, 1252–1259.

(28) Klein, D. J.; Schmalz, T. G. Fullerene structures. In *Buckminsterfullerenes*; Billups, W. E., Ciufolini, M. A., Eds.; VCH Publishers: New York, 1993; pp 83–101.

(29) Liu, X.; Klein, D. J.; Babić, D.; Schmalz, T. G., manuscript in preparation.

(30) (a) Liu, X.; Schmalz, T. G.; Klein, D. J. Favorable structures for higher fullerenes. *Chem. Phys. Lett.* **1992**, *188*, 550–554. (b) Manolopoulos, D. Comment on "Favorable structures for higher fullerenes". *Chem. Phys. Lett.* **1992**, *192*, 330. (c) Liu, X., Schmalz, T. G., Klein, D. J. Reply to Comment on "Favorable structures for higher fullerenes". *Chem. Phys. Lett.* **1992**, *192*, 331.

(31) Babić, D. manuscript in preparation.

(32) Hosoya, H. Topological index. A newly proposed quantity characterizing the topological nature of structural isomers of saturated hydrocarbons. *Bull. Chem. Soc. Jpn.* **1971**, *44*, 2332–2339.

(33) Harary, F. *Graph Theory*; Addison-Wesley: Reading, MA, 1972; p 39.

(34) Sah, C. H. A generalized leapfrog for fullerene structures. *Fullerene Sci. Technol.* **1994**, *2*, 445–458.

(35) Fowler, P. W.; Cremona, J. E.; Steer, J. I. Systematics of bonding in non-icosahedral carbon clusters. *Theor. Chim. Acta* **1988**, *73*, 1–26.

(36) Schmalz, T. G.; Seitz, W. A.; Klein, D. J.; Hite, G. E. Elemental carbon cages. *J. Am. Chem. Soc.* **1988**, *110*, 1113–1127.

CI940109O