Isomer Generation: Syntactic Rules for Detection of Isomorphism

István Lukovits

Chemical Research Center, Hungarian Academy of Sciences, H-1525 Budapest, P.O. Box 17, Hungary

Received October 10, 1998

The problem of exhaustive, nonredundant generation of isomers of acyclic graphs was considered. A new canonical indexing algorithm, based on the Morgan indexing technique and a method to count all "Morgan-trees", was proposed. The adjacency matrix obtained by using the proposed naming algorithm can be written in a compact form. The compressed adjacency matrix is an example of a primitive "language" denoting acyclic structures. Syntactic rules related to this language were used to discard duplicate structures.

INTRODUCTION

Association of labels to vertices of a graph (a "graph" and its chemical counterpart the "structural formula" have identical meanings, similarly expressions "vertex" and "atom" are synonyms in this paper) is a prerequisite for manipulation of the graph. Yet the number of possibilities of labeling does increase exponentially with the number of vertices N. The total number of labeled trees[1] (i.e. acyclic graphs which will be considered hereafter) containing N vertices is N^{N-2}. Indiscriminate assignation of labels—usually numbers 1 through N—to vertices of a graph is a rather inconvenient procedure because of the large number of possibilities. The number of labeled graphs will be reduced drastically if, after assigning the first label (1), each consecutive label is assigned to a vertex which is adjacent to an already labeled vertex. Knop et al.[2] denoted such labeled trees as "physical trees", and they devised a computer program to generate all physical trees containing N vertices. Each vertex of a physical tree is adjacent to a single vertex with a lower sequential index (the only exception is vertex 1); therefore each column in the upper right triangle of the adjacency matrix[1] **A** will contain a *single* nonzero entry.[3] The number of physical trees containing N vertices is therefore $(N-1)!$.

Canonical indexing means that the order of vertices is not arbitrary, but the indices are associated to vertices according to a fixed scheme. Canonical indexing ensures that each code represents a single structure, and vice versa, each structure can be denoted by a unique code, only. Once the code of a structure is known, the structure can be set up unambiguously. Various coding systems have been suggested among these the Wiswesser line notation,[4] the DENDRAL system,[5] the Morgan algorithm[6] and its variants,[7−9] the two-row matrix representation,[10] the N-tuple code method (NTCM) and its variants,[11−17] the "binary-code" formalism,[18] the Strokov code,[19] and the DARC system.[20] All these systems can be used in principle to generate formulas of isomers of any N-vertex acyclic structure,—where N denotes the number of vertices in the graph—and some of them for cycle-containing structures, too.

The construction of isomers may be achieved by generating systematically all possible codes. Because the number of possible acyclic isomers containing N vertices grows exponentially with N, any algorithm used to generate isomers will be exponential itself. Each new code may or may not be a canonical code, but this fact can only be detected *after* the code itself has been created. If the code is noncanonical, then the underlying structure will be isomorphic with one of the "canonical" structures and has to be deleted. In this paper a method to generate all nonisomorphic acyclic structures containing N vertices will be proposed.

The method proposed in this paper is based on a version of the Morgan algorithm.[21] A computer program was created which generates adjacency matrices of all trees indexed in accordance with the Morgan indexing algorithm. It was shown that the adjacency matrix obtained by this algorithm can be compressed simplifying the encoding of structures. The mathematical properties of the compressed adjacency matrix (CAM) were investigated. The Morgan rules were augmented by the "lowest degrees first" (LDF) rule, and it was shown that any CAM derived by this combined system of rules is maximal. CAMs are "sentences" of a primitive "language" denoting acyclic structures. Many redundant structures can be discarded by considering syntactic rules of this language alone. Although syntactic rules are very efficient, they are not yet sufficient (for the time being) to discard *all* duplicates, and the remaining redundant structures have to be detected by using "semantic procedures", where "semantic" means that the indexing has to be checked by using the full adjacency matrix (and if necessary the distance matrix).

MORGAN-TREES

Kvasnička and Pospichal[21] devised a simplified interpretation of the Morgan indexing algorithm,[6] which consists of the following steps:

1. An arbitrary vertex is marked by 1.

2. If this vertex is of degree p, then all adjacent vertices are indexed by 2, 3, ..., $p + 1$, in whatever order.

3. Consider vertex 2 of degree q. $q - 1$ neighbors of vertex 2 are still nonlabeled vertices. Assign numbers $p + 2$, $p + 3$, ..., $p + q$ to the nonlabeled neighbors of vertex 2.

4. Continue the procedure for each vertex 3, 4, ..., $N-1$ keeping in mind that each time the nonlabeled neighbors of the vertex with the *lowest* serial number should be considered first.

10.1021/ci9801522 CCC: $18.00
Published on Web 04/17/1999

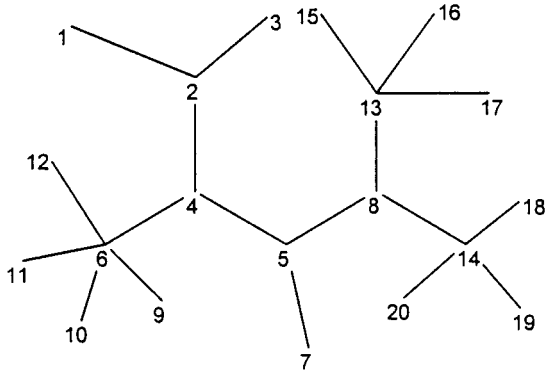

Figure 1. An example of the proposed numbering of an arbitrary tree.

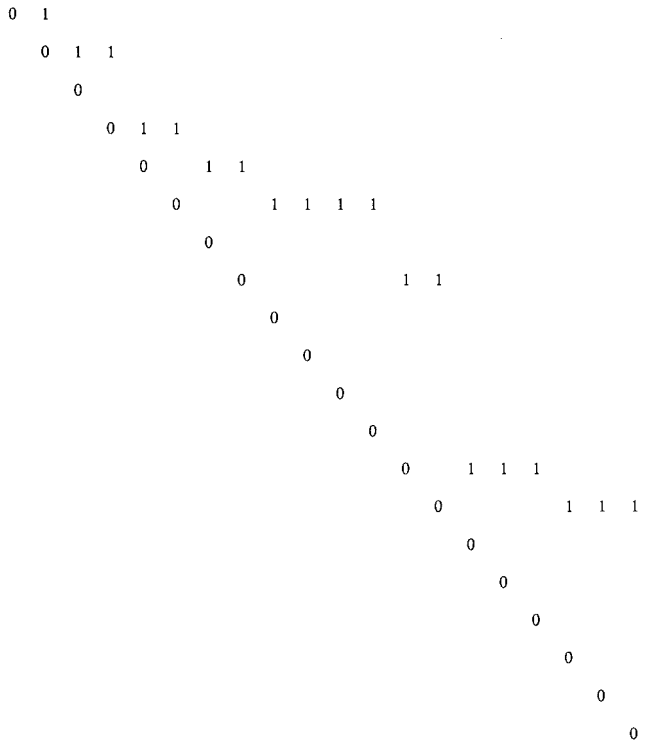

Figure 2. Schematic representation of the upper right triangle of the adjacency matrix of the structure depicted in Figure 1. Only zeros in the diagonal are shown, other zeros are suppressed for the sake of clearness.

Labeled trees that were indexed by using the Morgan algorithm[6] will be referred to as "Morgan-trees". Figure 1 depicts a Morgan-tree, and Figure 2 shows the structure of the respective adjacency matrix. The upper right triangle of matrix **A** will be considered hereafter, only. Note that a typical Morgan-tree will contain several nonzero entries in line 1, because usually the numbering starts at the vertex of the highest degree. Observe the architecture of matrix **A** which may be expressed by the following conditional: if $\mathbf{A}(i_1,j_1) = 1$, and $\mathbf{A}(i_2,j_2) = 1$, and $j_1 < j_2$, then $i_1 \leq i_2$. It is clear that the number of Morgan-trees is definitely lower than the number of physical trees.

To determine the number of Morgan-trees observe that $\mathbf{A}(1,2) = 1$ in all cases. The third column of matrix **A** also contains a single nonzero entry, and therefore there are two possibilities: either $\mathbf{A}(1,3) = 1$ or $\mathbf{A}(2,3) = 1$. If $\mathbf{A}(1,3) = 1$ then in the fourth column $\mathbf{A}(1,4) = 1$ or $\mathbf{A}(2,4) = 1$ or $\mathbf{A}(3,4) = 1$, i.e., there are three possibilities. On the other hand, if $\mathbf{A}(2,3) = 1$, then there are only two possibilities: $\mathbf{A}(2,4) = 1$ or $\mathbf{A}(3,4) = 1$, therefore $\mathbf{A}(1,4) = 1$ is realized only once, whereas $\mathbf{A}(2,4) = 1$ or $\mathbf{A}(3,4) = 1$ are realized in two cases. The same arguments apply for the fifth and all subsequent columns. The number of alternatives are listed in Table 1. $S(i,j)$ denotes the ith and jth entry in this table. Since

Table 1. Scheme Used To Determine the Number of Morgan-Trees[a]

	1	2	3	4	5	6	7	8	9	10	11
1	0	1	1	1	1	1	1	1	1	1	1
2		0	1	2	3	4	5	6	7	8	9
3			0	2	5	9	14	20	27	35	44
4				0	5	14	28	48	75	110	154
5					0	14	42	90	165	275	429
6						0	42	132	297	572	1001
7							0	132	429	1001	2002
8								0	429	1430	3432
9									0	1430	4862
10										0	4862
11											0

[a] For each entry $S(i,j) = S(i-1,j) + S(i,j-1)$, except the numbers of the first row and the diagonal.

$$S(1,j) = 1(j=2,N) \tag{1}$$

and

$$S(i,i) = 0(i=1,N) \tag{2}$$

it can easily be verified that

$$S(i,j) = S(i-1,j) + S(i,j-1) \tag{3}$$

and

$$S(i,j) = \sum_{k=1}^{k=i} S(k,j-1) \tag{4}$$

Equation 3 is a recurrent formula, which enables us to calculate the number of alternatives. The number of N-vertex Morgan-trees is equal to entry $S(N,N+1) = S(N-1,N+1)$.

LDF TREES

In this paper, for reasons to be explained later, the Morgan algorithm was replaced by the following—somewhat different—naming algorithm.

Step 1. Mark the endpoint (i.e., a vertex with a single neighbor) of the *longest* side chain by 1. (A side chain is a subgraph of the graph under consideration, consisting of one endpoint and—if there are any—vertices of degree two.) This rule will be referred to as the "endpoint convention", hereafter.

Step 2. Mark the first neighbor of vertex 1 by 2 and then start numbering neighbors of vertex 2. If there are several neighbors, then their order is *not* arbitrary: the vertex of the lowest degree (valence) has to be considered first. Remaining neighbors of higher degrees should also be considered in the order of increasing valencies. For vertices of the same degree, the second nonlabeled neighbors will decide the order. If all second neighbors have again identical degrees, continue with the inspection of the third, etc. nonlabeled neighbors.

Table 2. Number of Acyclic Structures Generated by Various Indexing Techniques in Terms of the Number of Vertices *N*

N	physical trees	Morgan-trees	rule 1	rules 1 and 2	no. of isomers
1	1	1	1	1	1
2	1	1	1	1	1
3	2	2	1	1	1
4	6	5	2	2	2
5	24	14	5	3	3
6	120	42	14	8	6
7	720	132	42	20	11
8	5040	429	132	90	23
9	40320	1430	429	279	47
10	362880	4862	1430	1170	106

Step 3. Continue the indexing with neighbors of vertex 3, then with neighbors of vertex 4, etc. Make sure that nonlabeled neighbors of the vertex with the lowest index are considered first, and within this set the neighbor of the lowest degree is numbered first.

The present method will be referred to as the "lowest degree first" (LDF) algorithm; it does not agree with rules used by other sources, which usually start indexing at a vertex with the highest degree.[7,8] Although Randić[18] also suggested that an endpoint (or in cyclic structures a vertex of the lowest degree) should be numbered first, in his scheme the first neighbor of vertex 1 gets the highest possible serial number (namely *N*), and therefore trees numbered by using his method are nonphysical (and therefore non-Morgan) trees.

Trees generated by the algorithm given by steps 1−3 will be called LDF trees. All LDF trees are also Morgan-trees, but not every Morgan-tree is a LDF tree, since LDF trees have maximal CAMs (see below). The first line of matrix **A** of any LDF tree contains a single nonzero entry; namely $\mathbf{A}(1,2) = 1$, and this will further reduce the number of allowed adjacency matrices.

Note that because of the LDF algorithm only adjacency matrices containing a single 1 in the first row need to be considered. The only nonzero entry in the first row is therefore $\mathbf{A}(1,2) = 1$, and the number of all *N*-vertex trees labeled by using the endpoint convention is equal to the number of $N - 1$ vertex Morgan-trees.

THE COMPRESSED ADJACENCY MATRIX

Each column of the upper right triangle of adjacency matrices of physical trees, Morgan-trees, and LDF trees contains a single nonzero entry. This property may be used to define a new, unique code for trees: let us consider the row index of each nonzero entry and collect these figures in a vector. The position of any number in the vector determines the column index of the respective nonzero entry; the *j*th entry refers to the $(j+1)$th vertex. If the LDF algorithm is taken into account, the first and second entries in this CAM will always be equal to one and two, respectively, and the third entry will be either 2 or 3, etc. The last (*N*-first) entry of the CAM may take on any value from 2 through $N - 1$. CAMs are codes which will be used to denote structures. As an example consider the CAM of the structure depicted in Figure 1: its CAM is (1, 2, 2, 4, 4, 5, 5, 6, 6, 6, 6, 8, 8, 13, 13, 13, 14, 14, 14). The underlying adjacency matrix **A** can be reconstructed from its CAM. If the *j*th entry of CAM is equal to *k*, then $\mathbf{A}(k,j+1) = \mathbf{A}(j+1,k) = 1$. The commas are only needed to separate numbers, but a more adequate number system, like the ASCII code, would make the use of commas unnecessary (provided that the tree has less than 256 vertices), and the numbers could be concatenated to form a single code (Table 3). CAM of any Morgan-tree is a monotonically increasing sequence of numbers, and the codes can be arranged in lexicographic order. CAMs of Morgan-trees will be considered hereafter, only.

CAMs may be used to determine the number of vertices of various degrees. Numbers less than or equal to $N - 1$ which do *not* appear in the respective CAM are endpoints, the only exception is digit 1 itself, since it denotes an endpoint by definition (cf. step 1). Similarly numbers appearing only once denote vertices of degree two; number 1 is again the only exception. In general the degree of vertex *k* may be obtained by counting the number of times *k* appears in the CAM and by increasing this sum by one. In our example (Figures 1 and 2) vertices 2, 4, 5, and 8 are trivalent, vertices 13 and 14 are tetravalent, and the degree of vertex 6 is equal to five. Vertices 3, 7, 9, 10, 11, 12, 15, 16, 17, 18, 19, and 20, which do not appear in CAM, are endpoints, and vertex 1 is also an endpoint. The following algorithm may be used to devise the vector v of degrees:

1. Set each entry of v equal to one, that is $v(i) = 1$ ($i = 1, 2, ..., N$).

2. Consider entries $j = 2, 3, ..., N - 1$ of the CAM in turn and denote the value of the *j*th entry by *k*. Increase the actual value of $v(k)$ by one.

Entries related to endpoints will not be affected by this procedure and remain equal to one.

The LDF codes of any tree can be obtained by the following procedure:

1. Consider vertex $k = 2$.

2. Determine the (first) neighbors of *k*.

3. Out of this set pick out the lowest number. (If LDF rules were used to index the vertices the lowest one would be less than *k* and denotes a vertex already referred to.)

4. Insert this number into the $(k-1)$th entry of the CAM.

5. Increase *k* by one and go to step 2 if $k < N$.

6. Terminate procedure if $k = N$.

Example: consider vertex 4 of Figure 1, it has three first neighbors, namely 2, 5, and 6. Number 2 is the lowest serial number in the actual set, and this figure will appear in the third entry of the CAM. (See first paragraph of this section.)

Let us compare CAM_1 with CAM_2: it will be said that CAM_1 is greater than CAM_2, if $a_i = b_i$ ($1 \le i < j$) and $a_j > b_j$ where a_i and b_i denote the entries in CAM_1 and CAM_2, respectively. The shorthand notation for this relation will be $CAM_1 > CAM_2$. The present definition of precedence is analogous to that given in the paper by Knop et al.[2] While each tree T may possess several CAMs, the LDF algorithm—as it will be shown below—yields the *maximal* CAM of all Morgan-trees. The maximal CAM was considered as the only valid code representing the underlying structure T. Maximal CAMs are LDF codes.

Let us consider vertices *j* and *k* attached to the *same* branching atom; their order will be determined by the LDF rules: if $v(j) < v(k)$, then $j < k$. Then index *j* appears less often than *k*, and the corresponding CAM is (1, 2, ..., *j*, *k*, *k*, ...). In this example it has been assumed that the degrees of *j* and *k* are two and three, respectively. If the order of *j* and *k* is reversed—and therefore the LDF rules are violated—the new CAM′ is less than CAM since CAM′ = (1, 2, ..., *j*, *j*, *k*,

Table 3. Concatenated Compressed Adjacency Matrices of all Morgan-Trees of Heptane Listed in Increasing Lexicographic Order[a]

no.	CAM	no. of maximal CAM	no. of rule used, if CAM was deleted
1	122222	max.	
2	122223	29	rule 2
3	122224	29	rule 2
4	122225	29	rule 2
5	122226	29	rule 2
6	122233	24	rule 2
7	122234	32	rule 2
8	122235	32	rule 2
9	122236	38	rule 2
10	122244	24	rule 2
11	122245	32	rule 2
12	122246	38	rule 2
13	122255	24	rule 2
14	122256	38	rule 2
15	122333	24	rule 2
16	122334	40	rule 2
17	122335	36	rule 2
18	122336	36	rule 2
19	122344	36	rule 2
20	122345	40	rule 2
21	122346	40	rule 2
22	122355	27	rule 2
23	122356	41	rule 2
24	122444	max.	
25	122445	36	rule 3
26	122446	36	rule 5
27	122455	max.	
28	122456	41	rule 5
29	123333	max.	
30	123334	32	rule 3
31	123335	32	rule 3
32	123336	max.	
33	123344	36	rule 4
34	123345	max.	
35	123346	40	rule 4
36	123355	max.	
37	123356	40	rule 5
38	123444	max.	
39	123445	40	rule 3
40	123446	max.	
41	123455	max.	
42	123456	max.	

[a] CAMs deleted by using rule 1 are not shown. Maximal (LDF) codes are underlined and the corresponding structures are shown in Figure 3.

...) < (1, 2, ..., *j*, *k*, *k*, ...) = CAM. Similar arguments apply for any degree. By showing this we have proved theorem 1.

Theorem 1. CAMs obtained by the LDF procedure are maximal. Table 3 lists all 42 CAMs of heptane; labeled structures violating the endpoint convention have been listed. Therefore all CAMs containing more than one digit "1"-like code (1, 1, 1, 1, 1, 1), which denotes a star (Figure 3) but which is equivalent to CAM = (1, 2, 2, 2, 2, 2), were not listed. The third column of this table lists the numbers of isomorphic structures, two structures namely 1 and 42 (the star and the chain) have no isomorphic variants because structure 1 is already a maximal Morgan-tree, and structure 42 has no variants because of the endpoint convention. It can be seen that all CAMs with a serial number less than 24—except the code denoting the star—are not maximal CAMs and were deleted. (In the next section this practice will be verified.) The structures corresponding to the maximal CAMs are shown in Figure 3.

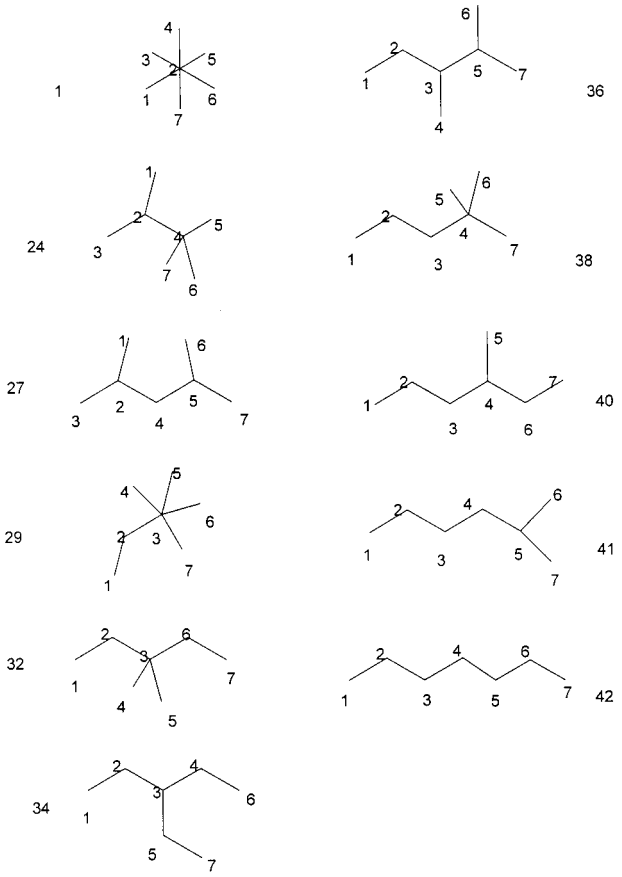

Figure 3. Canonically numbered isomers of heptane. The numbers agree with those given in the first column of Table 3.

SYNTACTIC RULES

CAMs may be interpreted as sentences of a primitive "language". The entries of the CAMs are the "words". The meaning of any such sentence is the structure of the underlying tree. In this language (as in any other language) there are syntactic rules, which—irrespectively of the meaning of the sentence—must be adhered to. By "syntactic rules" we shall denote all those prescriptions by which incorrect sentences can be discarded *without* investigating the underlying structures themselves. Methods which do require the investigation of the full adjacency matrix will be referred to as "semantical rules". The endpoint convention of the LDF algorithm provides therefore the first syntactic rule.

Rule 1. Delete (or do not generate) any CAM containing more than a single entry equal to one. Because Morgan-trees are considered only, the number equal to one must be the first entry in the CAM.

Structure 24 (Figure 3) contains two branching vertices and only a single edge connecting these. Such structures will be referred to as "bistars". It will be shown that the first maximal CAM after a star is a bistar, and *all* structures between these can be deleted. The CAM of any bistar, denoted by CAM_b, has the following structure: CAB_b = (1, 2, 2, ..., 2, *X*, *X*, ..., *X*), and for any *N*, digit 2 appears *k* times and *X* ($X = k + 2$) appears *m* times and $k = m = (N-2)/2$ if *N* is even, and $k + 1 = m = (N-1)/2$, if *N* is odd. Example: for $N = 7$ (heptanes) $k = 2$ and $m = 3$ (structure 24, Figure 3, and Table 3). Since CAB_b may be obtained by using the LDF algorithm, it is maximal. Theorem 2 ensures that any CAM_x such that (1, 2, ..., 2) < CAM_x < (1, 2, 2, ..., 2, *X*, *X*, ..., *X*) can be discarded.

Theorem 2. There is no maximal CAM_x such that $CAM_s < CAM_x < CAM_b$, where $CAM_s = (1, 2, ..., 2)$ and $CAM_b = (1, 2, 2, ..., 2, X, X, ..., X)$ and CAM_b contains k digits equal to two, and m numbers equal to X and $k = m = (N-2)/2$ if N is even and $k + 1 = m = (N-1)/2$, if N is odd.

Proof. Let us suppose that CAM_x is maximal, and $CAM_s < CAM_x < CAM_b$. If N is even, then none of the X's can be replaced by another number Y, such that $Y < X$, since this would violate the rule that numbering starts at an endpoint attached to a the longest side chain. If N is odd, then at most one X may be replaced by Y. But then the first endpoint will not belong to the longest side chain (step 1), since there is a path $2 - Y - Z$ of length two, where Z refers to an endpoint and $Z = k + 3$. (Example: for heptanes $k = 2$ and $Z = 5$.) Taking into account theorem 2 we may formulate rule 2.

Rule 2. Delete (or do not generate) any CAM_x such that $CAM_s < CAM_x < CAM_b$.

By inspecting Table 3 (structures 25, 30, 31, and 39) it may also be observed that if the last digit is not equal to $N - 1$ *and* it is preceded by at least two identical digits, then the CAM has to be deleted, because step 2 of the naming algorithm has been violated: vertex N is not attached to the last member of a set of equivalent vertices.

Rule 3. Delete any $CAM = (1, 2, ..., X, ..., X, Z)$ if $Z < N - 1$.

Both CAMs 33 and 35 (Table 3) contain duplicate numbers (namely number three) referring to the fourth and fifth vertex. Number 4 appears in the respective CAM, indicating that a vertex of degree two or more is attached to vertex 4, but digit 5 does not appear in these CAMs, indicating that vertex 5 is an endpoint. It should have been indexed "4" instead so the LDF rules have been violated. For such cases we can formulate rule 4.

Rule 4. Delete any $CAM = (1, 2, ..., Y, ..., Y, ...)$ with Y appearing k times if any Y in position k_1 is referred to (i.e. number $k_1 + 1$ appears in the string), whereas an Y in position k_2 ($k_2 > k_1$) is not.

The following syntactic rule applies if the last vertex is associated to a side chain which is longer than that containing vertex 1: cases 28 and 37 (Table 3) are examples.

Rule 5. Delete any $CAM = (1, 2, ..., k, V, ..., Y, Y, A_1, ..., A_m)$, if $A_1 = N - m$, and $A_{i+1} = A_i + 1$, ($i = 1, ..., m$) and k is the kth entry, and $k \neq V$, and $k \leq m + 1$.

DISCUSSION

The preceding rules are not sufficient to discard all redundant structures in a series of acyclic graphs with a higher number of vertices. Nevertheless syntactic rules are rather efficient, because the time needed to inspect a CAM increases linearly with N. It is clear that further syntactic rules could be found (and work is continued along this line), but ultimately syntactic rules alone might be insufficient to filter out *all* redundant structures. Therefore an algorithm based on semantic rules is still necessary. Here we outline such a procedure, but the full algorithm and the corresponding computer program—including the part which generates the CAMs—will be presented in another publication.[22]

The semantic procedure starts with the inspection of the branching vertices in reverse order. At each step the order of side chains must be checked, and if the order does not agree with the LDF algorithm, the CAM must be deleted. If the order is correct, the branching atom and all the side chains attached to it will be indexed temporarily. The corresponding CAM may contain more than one 1 in this case, since the indexing starts at the branching vertex. Then the next branching vertex will be checked. If the order is again correct, all portions already checked and connected to the actual vertex will be renumbered and a new CAM will be derived. The procedure is complete when the first branching vertex is also verified. Any violation of the LDF rules leads to omission of the respective CAM.

Instead of this algorithm, any other method designed to detect isomorphism may be used.[23] In this case the following steps are necessary: 1. Generate CAMs of Morgan-trees of increasing lexicographic order. 2. Convert each new code CAM_n into an adjacency matrix and check whether the underlying structure is isomorphic with any previous structure encoded by CAM_m and $m < n$. 3. If CAM_n and CAM_m encode the same structure, delete CAM_m since its "value" is less than that of CAM_n. 4. Go to step 1 unless all possible codes were generated.

The efficiency of the present method will be compared with that of the N-tuple code method,[12] (NTCM) since this method seems to have been used quite frequently. The NTCM consists of the following steps: 1. Creation of a code, this algorithm is $O(\alpha^N)$, where α is a constant. 2. Verification that the actual code denotes a tree; this procedure is $O(N^1)$ due to the authors.[12] 3. A procedure to test whether the code is maximal, the algorithm is $O(N^2)$. 4. Output. Most probably this step is $O(N^1)$.

It has to be emphasized that the NTCM does not generate Morgan-trees, only physical trees. The sum of entries of an N-tuple code is $N - 1$. The first entry may take on any value X_1 such that $2 \leq X_1 \leq N - 1$. Each following entry k may take on any value X_k, such that $0 \leq X_k < X_1$ and $\sum X_k \leq N - 1$. Therefore the number of alternatives to be created grows exponentially with N. Step 2 is not necessary for Morgan-trees, since CAMs always name acyclic structures. Syntactic rules derived so far and the corresponding algorithms are $O(N^1)$, i.e., are more efficient than step 3 of NTCM. In fact in both cases (NTCM and Morgan-trees) the time needed to generate all isomers for any N-tree would grow exponentially with N because of step 1. To determine whether the NTCM or the LDF method is more efficient, computer experiment would be necessary.

In conclusion we may state that the procedure of the reduction of CAMs of Morgan-trees is greatly simplified if first syntactic rules, and then semantic rules are used to delete all nonmaximal CAMs. The advantage of syntactic rules 1 and 2 is that many CAMs do not need to be generated. The LDF naming algorithm ensures that the codes obtained are maximal CAMs.

ACKNOWLEDGMENT

The author is indebted to Drs. Darko Babić, and Ante Graovac and Prof. Nenad Trinajstić (The Rudjer Bošković Institute, Zagreb, Croatia) for helpful discussions and recommending highly relevant literature.

REFERENCES AND NOTES

(1) Harary, F. *Graph Theory*; Addison−Wesley: Reading, 1969; p 179.

(2) Knop, J. V.; Müller, W. R.; Szymanski, K.; Nikolić, S.; Trinajstić, N. *Computer Generation of Certain Classes of Molecules*; SKTH/Kemija u industriji, Zagreb, 1985; p 3.
(3) Gutman, I.; Linert, W.; Lukovits, I.; Dobrynin, A. A. Trees with Extremal Hyper-Wiener Index: Mathematical Basis and Chemical Applications. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 349−354.
(4) Smith, E. G. *The Wiswesser Line-Formula Chemical Notation*; McGraw-Hill: New York, 1968; p 1.
(5) Lederberg, J.; Sutherland, G. L.; Buchanan, G. G.; Feigenbaum, E. A.; Robertson, A. M.; Duffield, A. M.; Djerassi. Applications of Artificial Intelligence for Chemical Inference. I. The Number of possible Organic Compounds. Acyclic Structures Containing C, H, O and N. *J. Am. Chem. Soc.* **1969**, *91*, 2973−2976.
(6) Morgan, H. L. The Generation of a Unique Description for Chemical Structures.−A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 107−113.
(7) Wipke, W. T.; Dyott, T. M. Stereochemically Unique Naming Algorithm. *J. Am. Chem. Soc.* **1974**, *96*, 4834−4842.
(8) Balaban, A. T. Topological Indices and Their Uses: A New Approach for the Coding of Alkanes. *J. Mol. Struct. (THEOCHEM)* **1988**, *165*, 243−253.
(9) Nagy, Z. M. How Can Parallel Algorithms Help to Find New Sequential Algorithms? *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 542−544.
(10) Bangov, I. P. Structure Generation from a Gross Formula. 7. Graph Isomorphism: A Consequence of the Vertex Equivalence. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 167−173.
(11) Read, R. C. The Enumeration of Acyclic Chemical Compounds. in *Chemical Applications of Graph Theory*; Balaban, A. T., Ed.; Academic Press: London, 1976; pp 25−61.
(12) Knop, J. V.; Müller, W. R.; Jeričević, Ž.; Trinajstić, N. Computer Enumeration and Generation of Trees and Rooted Trees. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 91−99.
(13) Contreras, M. L.; Rozas, R.; Valdivia, R.; Agüeras, R. Exhaustive Generation of Organic Isomers. 4. Acyclic Stereoisomers with One or More Chiral Carbon Atoms. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 752−758.
(14) Hansen, P.; Jaumard, B.; Lebatteux, C.; M. Zheng. Coding Chemical Trees with the Centered N-tuple Code. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 782−790.
(15) Hendrickson, J. B.; Toczko, A. Unique Numbering and Cataloguing of Molecular Structures. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 171−177.
(16) Randić, M.; Nikolić, S.; Trinajstić, N. Compact Codes; On Nomenclature of Acyclic Chemical Compounds. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 357−365.
(17) Kirby, E. C. Coding and Enumeration of Trees that can be Laid upon a Hexagon Lattice. *J. Math. Chem.* **1992**, *11*, 187−195.
(18) Randić, M. On Canonical Numbering of Atoms in a Molecule and Graph Isomorphism. *J. Chem. Inf. Comput. Sci.* **1977**, *17*, 171−180.
(19) Strokov, I. A Compact Code for Chemical Structure Storage and Retrieval. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 939−944.
(20) Dubois, J. E.; Carrier, G.; Panaye, A. DARC Topological Descriptors for Pattern Recognition in Molecular Database Management Systems and Design. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 574−578.
(21) Kvasnička, V.; Pospichal, J. Canonical Indexing and Constructive Enumeration in Molecular Graphs. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 99−105.
(22) Lukovits, I. Automatic Generation of Isomeric Structures: A Computer Procedure Employing Semantic Rules. *J. Chem. Inf. Comput. Sci.* (to be published).
(23) Faulon, J. L. Isomorphism, Automorphism, Partitioning, and Canonical Labeling Can be Solved in Polynomial Time for Molecular Graphs. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 432−444.

CI9801522