

***Handbook of Chemoinformatics***  
**Wiley–VCH**  
**2003**

**Canonical Numbering and Constitutional Symmetry**

**Ovidiu Ivanciuc**

Sealy Center for Structural Biology and Molecular Biophysics  
Department of Biochemistry and Molecular Biology  
University of Texas Medical Branch  
301 University Boulevard  
Galveston, Texas 77555-0857, USA  
Email: ovidiu\_ivanciuc@yahoo.com  
Email: iejmd@yahoo.com  
URL: <http://ivanciuc.org/>  
URL: <http://biochempress.com/>

**O. Ivanciuc**, Canonical Numbering and Constitutional Symmetry.  
In: *Handbook of Chemoinformatics*, Ed.: J. Gasteiger, Wiley–VCH,  
2003, pp. 139–160.

## 5 Processing Constitutional Information

Ovidiu Ivanciuc's biographical notes are given at the beginning of Chapter II, Section 4.

### 5.1 Canonical Numbering and Constitutional Symmetry

*Ovidiu Ivanciuc*

#### 5.1.1 Introduction

Chemoinformatics systems use a wide variety of algorithms for indexing and retrieving chemical compounds in databases, generating all isomers with a constitutional formula, or for computer-assisted organic synthesis. All these tasks involve three classes of algorithm for chemical graphs:

1. the canonical coding problem, for generating a unique representation of a chemical compound;
2. the automorphism partitioning problem (also known as constitutional symmetry perception, graph symmetry, or topological symmetry), for detection of equivalent atoms and bonds in a molecule; and
3. the graph isomorphism problem, for determining if two connection tables represent the same chemical compound.

The three problems are related and their considerable practical and theoretical importance has encouraged many mathematicians and chemists to investigate them. The main chemical applications of canonical coding and constitutional symmetry perception are briefly presented below:

1. In a chemical documentation system each substance must be characterized by a unique code, which is used for storage, retrieval and comparison of chemical compounds. The nomenclature systems used by man to communicate chemical information are not suitable for computer manipulation, and special chemical structure representations are developed for chemical database management and searching.
2. The computer generation of chemical compounds consistent with given structural constraints is used both in synthesis design and in structure elucidation. The chemical structure set generated must be exhaustive (to contain all compounds consistent with the generation rules) and non-redundant (to contain

each structure only once). The main problem of most structure-generation algorithms lies in generating a great number of redundant structures, which have to be found and eliminated. Canonical coding of structures is used to eliminate redundant compounds, while constitutional symmetry information is necessary to generate all possible structures.

3. Artificial intelligence systems for synthesis design use canonical codes and constitutional symmetry information to generate and evaluate reaction paths.
4. The synthesis by computer-aided molecular design of new compounds that conform to various physical property requirements can reduce the time and effort required using traditional empirical approaches. This process generates chemical structures compatible with experimental and structural restrictions. Canonical coding and symmetry perception are used to uniquely generate all possible isomers that adhere to the design constraints.
5. Constitutional symmetry is important for interpretation of NMR and ESR spectra, and computer-assisted structure elucidation systems use this information.
6. Quantitative structure–activity relationships are used to model the biological effect of a set of compounds and to propose new structures with optimized biological activity. Such a system makes extensive use of structure generation, substructure search and symmetry perception algorithms.
7. Modern drug-design procedures make extensive use of large combinatorial libraries and in silico screening of chemicals, both using algorithms based on canonical coding, graph isomorphism, and constitutional symmetry perception.

The problems of canonical coding, graph isomorphism, and graph automorphism have both mathematical and chemical significance. Mathematical formulation of these problems is covered briefly below, and some connections with their chemical counterparts are presented. In subsequent sections the main algorithms used in chemistry for canonical coding of molecular graphs and constitutional symmetry perception are presented and compared. We will use definitions introduced in the graph theory section (graphs, molecular graphs, molecular matrixes, characteristic polynomial and matrix spectra) (Chapter II, Section 4) and in the topological indices section (vertex and molecular invariants) (Chapter VIII, Section 1). This review is a modified and updated version of a chapter published in the Encyclopedia of Computational Chemistry [1].

### 5.1.2 Graph Labeling

All algorithms for canonical coding and graph symmetry determination generate, compare, and manipulate labeled graphs. The process of generating all labelings of a graph is presented below. A molecular graph  $G = G(V, E)$  consists of a non-empty set  $V$  of vertices representing atoms and a set  $E$  of edges representing chemical bonds; the number of vertices is  $N = |V(G)|$ . If the edge  $e_{ij} \in E$ , then the

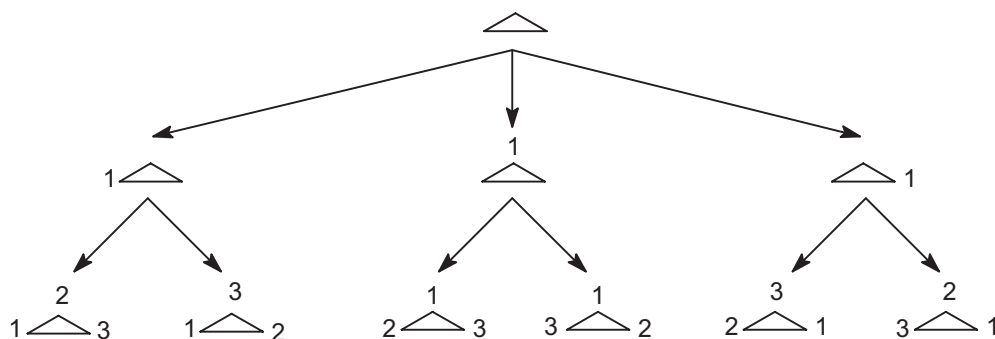


Fig. 5.1-1 The tree representation of the generation of the  $3! = 6$  permutation labelings of cyclopropane

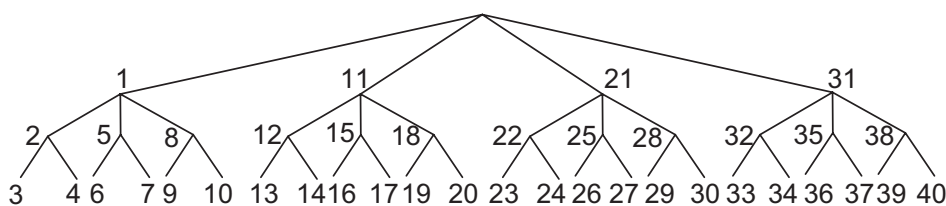


Fig. 5.1-2 The tree representing the depth-first permutation labeling of a graph with four vertices

two vertices  $v_i$  and  $v_j$  are adjacent, and  $e_{ij}$  is incident with  $v_i$  and  $v_j$ . The degree of a vertex  $v_i$ ,  $\text{deg}_i$ , is the number of edges adjacent with  $v_i$ .

A labeling  $Lb$  of the graph  $G$  composed of  $N$  vertices consists of a one-to-one mapping  $Lb : V(G) \rightarrow \{1, 2, \dots, N\}$ . The integer  $Lb(v) \in \{1, 2, \dots, N\}$  assigned to a vertex  $v \in V(G)$  is called the label of the vertex  $v$ . A graph  $G$  together with the mapping  $Lb$  is called a labeled graph and is denoted by  $G(Lb)$ . For a graph with  $N$  vertices there are  $N!$  permutation labelings.

The generation of all  $N!$  labelings of a graph can be represented as a rooted tree where the root node is the unlabeled graph, at each node a new label is added to an unlabeled vertex, and each terminal node is a completely labeled graph corresponding to one labeling. The tree representing the generation of the  $3! = 6$  permutation labelings of cyclopropane is presented in Figure 5.1-1.

Two basic approaches are used to construct the tree of permutation labelings: breadth-first or depth-first. Each one has advantages and disadvantages, and most coding, isomorphism, and automorphism algorithms use a mixture of these two approaches. The order of exploring the labeling tree for the depth-first method is presented in Figure 5.1-2; the breadth-first labeling process is presented in Figure 5.1-3. The problem of generating  $N!$  labelings in canonical coding algorithms can be reduced by using rules which allow cutting of some branches of the labeling tree.

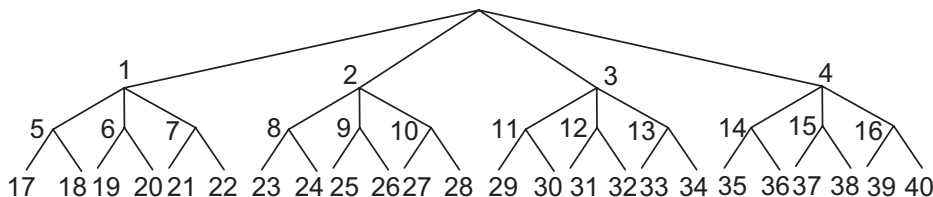


Fig. 5.1-3 The tree representing the breadth-first permutation labeling of a graph with four vertices

### 5.1.3

#### Constitutional Symmetry of Graphs

The problem of graph symmetry investigates the equivalence relationships between the elements of the molecular graphs (atoms, bonds, pairs of atoms, etc.). The geometrical information is neglected and only bonding relationships are considered. Consider two graphs  $G = G(V, E)$  and  $G' = G'(V', E')$  with  $N|V| = N|V'|$  and a mapping  $m : V \rightarrow V'$  which assigns to each vertex  $v \in V$  a vertex  $v' \in V'$  in such a way that if  $v_i \neq v_j$  then  $m(v_i) \neq m(v_j)$ . The two graphs  $G$  and  $G'$  are called isomorphic if there exists a mapping  $m$  between  $V$  and  $V'$  which preserves the adjacency of vertices, i.e. if  $e_{ij} \in E$ ,  $v_k = m(v_i)$ ,  $v_l = m(v_j)$  then  $e_{kl} \in E'$ . The problem of recognizing if two graphs  $G$  and  $G'$  are isomorphic or not is called the graph isomorphism problem [2]. The chemical counterpart of this problem is to determine if two molecular graphs represent the same chemical compound.

An isomorphism of a graph with itself is called an automorphism. An automorphism can be represented by a permutation (mapping) that transforms a graph labeling into another labeling and preserves the adjacency of the vertices. A permutation  $\mathbf{P}$  represented in a two-row notation has the following form (Eq. (1)):

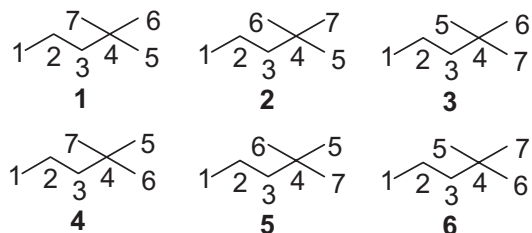
$$\mathbf{P} = \begin{pmatrix} 1 & 2 & 3 & \cdots & i & \cdots & N \\ p_1 & p_2 & p_3 & \cdots & p_i & \cdots & p_N \end{pmatrix} \quad (1)$$

with the meaning that atom 1 is permuted to atom  $p_1$ , atom 2 is permuted to atom  $p_2$ , atom 3 is permuted to atom  $p_3$ , atom  $i$  is permuted to atom  $p_i$ , and atom  $N$  is permuted to atom  $p_N$ . An orbit is the set of all atoms that are transformed from one into another by the actions of all automorphisms of a molecular graph. The set of all different orbits of a molecular graph  $G$  forms a partition of  $G$ . If  $\mathbf{P}$  represents an automorphism then atom  $i$  is symmetric (topologically equivalent) with its image  $p_i$ , and atoms  $i$  and  $p_i$  belong to the same orbit. This symmetry relationship in the molecular graph may or may not be true for the three-dimensional molecular structure.

Let  $Aut(G) = (A, B, C, \dots)$  be the set of automorphisms of the atoms in a graph  $G$  and let  $\otimes$  symbolize a binary operation on  $Aut(G)$ .  $Aut(G)$  is called an automorphism group if the following conditions are satisfied [3]:

1. For any two permutations  $\mathbf{A}, \mathbf{B} \in \text{Aut}(G)$  there exists a unique element  $\mathbf{C} = \mathbf{A} \otimes \mathbf{B}$ ,  $\mathbf{C} \in \text{Aut}(G)$ .
2. The operations respect the associative law:  $\mathbf{A} \otimes \mathbf{B} \otimes \mathbf{C} = \mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C}) = (\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C}$  for all permutations  $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \text{Aut}(G)$ .
3. For every permutation  $\mathbf{A} \in G$  there exists an inverse permutation  $\mathbf{A}^{-1} \in \text{Aut}(G)$  such that  $\mathbf{A} \otimes \mathbf{A}^{-1} = \mathbf{A}^{-1} \otimes \mathbf{A} = \mathbf{E}$ .
4. The set  $\text{Aut}(G)$  contains a unique permutation  $\mathbf{E}$  such that  $\mathbf{A} \otimes \mathbf{E} = \mathbf{E} \otimes \mathbf{A} = \mathbf{A}$  for all  $\mathbf{A} \in \text{Aut}(G)$ . The permutation  $\mathbf{E}$  is called the identity permutation of the group  $\text{Aut}(G)$ .

The automorphism group describes all symmetry properties of a graph. The determination of the automorphism group of molecular graphs is important in enumeration and generation of isomers and interpretation and simulation of spectra. The combination of two automorphisms with the operation  $\otimes$ ,  $\mathbf{A} \otimes \mathbf{B} = \mathbf{C}$ , considers in the first step the second permutation, which transforms an atom  $i$  into its image  $\mathbf{B}(i)$ . The second step considers permutation  $\mathbf{A}$  and correlates an atom  $\mathbf{B}(i)$  from the first row with its image  $\mathbf{A}(\mathbf{B}(i))$  from the second row. The third step generates the permutation  $\mathbf{C}$  which correlates an original atom  $i$  with its image  $\mathbf{A}(\mathbf{B}(i))$ .



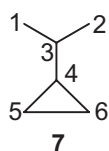
Consider the labeled graph of 2,2-dimethylpentane **1** and its five permutation labelings **2**, **3**, **4**, **5**, and **6**. The five orbits of 2,2-dimethylpentane are:  $X_1 = \{1\}$ ,  $X_2 = \{2\}$ ,  $X_3 = \{3\}$ ,  $X_4 = \{4\}$ ,  $X_5 = \{5, 6, 7\}$ . The identity permutation  $\mathbf{E}$  of **1** and the five automorphisms  $\mathbf{P}_1$ ,  $\mathbf{P}_2$ ,  $\mathbf{P}_3$ ,  $\mathbf{P}_4$ , and  $\mathbf{P}_5$ , which transform **1** in **2**, **3**, **4**, **5**, and **6**, respectively, are presented below:

$$\begin{aligned} \mathbf{E} &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{pmatrix} & \mathbf{P}_1 &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 7 & 6 \end{pmatrix} \\ \mathbf{P}_2 &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 7 & 6 & 5 \end{pmatrix} & \mathbf{P}_3 &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 6 & 5 & 7 \end{pmatrix} \\ \mathbf{P}_4 &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 7 & 5 & 6 \end{pmatrix} & \mathbf{P}_5 &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 6 & 7 & 5 \end{pmatrix} \end{aligned}$$

The combination of two automorphisms of 2,2-dimethylpentane, namely  $P_1 \otimes P_2 = P_5$ , is presented below:

$$\begin{aligned} P_1 \otimes P_2 &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 7 & 6 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 7 & 6 & 5 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 6 & 7 & 5 \end{pmatrix} \end{aligned}$$

Because the result of  $P_1 \otimes P_2$  is another automorphism of **1**, the above equation is an application of property (1) of automorphism groups.



Consider the molecular graph of isopropylcyclopropane **7**, its orbits  $X_1 = \{1, 2\}$ ,  $X_2 = \{3\}$ ,  $X_3 = \{4\}$ ,  $X_4 = \{5, 6\}$ , and its automorphisms:

$$\begin{aligned} E &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix} & A &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 3 & 4 & 5 & 6 \end{pmatrix} \\ B &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 6 & 5 \end{pmatrix} & C &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 3 & 4 & 6 & 5 \end{pmatrix} \end{aligned}$$

The combination of these permutations gives the following multiplication table:

	E	A	B	C
E	E	A	B	C
A	A	E	C	B
B	B	C	E	A
C	C	B	A	E

Each permutation has an inverse element as presented in the following table:

P	E	A	B	C
P <sup>-1</sup>	E	A	B	C

From the above properties of the permutations **E**, **A**, **B**, and **C**, one may conclude that they form the automorphism group of the graph of isopropylcyclopropane **7**. This automorphism group describes all constitutional symmetry relationships of isopropylcyclopropane.

A permutation of the vertices of a graph  $G$  can be described by a permutation matrix  $\mathbf{P}$  whose element  $[\mathbf{P}]_{ij} = 1$  if the vertex  $v_i$  is permuted to  $v_j$ , and  $[\mathbf{P}]_{ij} = 0$  otherwise. If a permutation is in the automorphism group then the following equation is valid (Eq. (2)):

$$\mathbf{A} = \mathbf{P}^T \cdot \mathbf{A} \cdot \mathbf{P} \quad (2)$$

The matrix representation of the permutation  $\mathbf{C}$  of 7 is:

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

To verify with Eq. (2) that the permutation  $\mathbf{C}$  is in the automorphism group of 7 we have to show that  $\mathbf{C}^T \mathbf{A} \mathbf{C} = \mathbf{A}$ , i.e.:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \\ = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

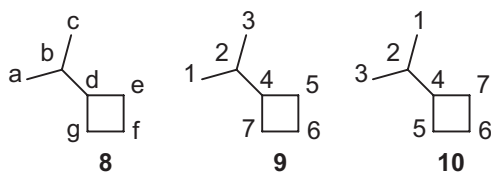
For a graph with  $N$  vertices there are  $N!$  permutation matrixes, and a brute force generation of  $Aut(G)$  requires  $N!$  tests to verify if Eq. (2) is satisfied. Algorithms that greatly reduce the number of tests in the determination of the automorphism group use information on constitutional symmetry (orbits) and avoid the generation of non-automorphic permutations [4, 5]. Balasubramanian developed algorithms to determine the automorphism groups of edge-weighted graphs [6] and to generate nuclear equivalence classes based on the three-dimensional molecular structure [7].



## 5.1.4

## Canonical Coding of Graphs

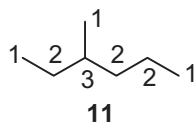
A code  $Cd(G, Lb)$  of the labeled graph  $G(Lb)$  is a string obtained from  $G$  by a set of rules. The code is not a structural invariant, because different labelings of  $G$  usually give different codes. A code is a complete representation of  $G(Lb)$  because the labeled graph can be reconstructed from  $Cd(G, Lb)$ . The code of a chemical compound is a numerical representation of the chemical structure suitable for computer manipulation. An important property of codes is that the lexicographical relation (or numerical relation, in the case of numerical codes) between two strings induces an order of the codes. Two labelings  $Lb_1$  and  $Lb_2$  of graph  $G$  are called equivalent,  $Lb_1 = Lb_2$ , if the corresponding codes are identical,  $Cd(G, Lb_1) = Cd(G, Lb_2)$ . Because a code depends on the graph labeling, if  $Cd(G, Lb_1) \geq Cd(G, Lb_2)$  then  $Lb_1 \geq Lb_2$  and  $G(Lb_1) \geq G(Lb_2)$ . In this way it is possible to order two labeled graphs. The set of all possible labelings  $Lb$  of graph  $G$  is denoted  $LbS(G)$ . A maximal canonical labeling  $Lb_{Mcan} \in LbS(G)$  of graph  $G$  generates a maximal canonical code  $Cd_{Mcan}$  with the property that  $\forall Lb \in LbS(G) : Cd(G, Lb) \leq Cd_{Mcan}(G, Lb_{Mcan})$ . A minimal canonical labeling  $Lb_{mcan} \in LbS(G)$  of graph  $G$  generates a minimal canonical code  $Cd_{mcan}$  with the property that:  $\forall Lb \in LbS(G) : Cd(G, Lb) \geq Cd_{mcan}(G, Lb_{mcan})$ . Both minimal and maximal canonical codes are used in chemistry, depending on the definition selected to represent the chemical structures. From the above definition it is clear that for a given molecular graph the canonical code is unique. This property is used in graph isomorphism testing and in storage, retrieval and comparison of chemical compounds, because two molecular graphs with identical canonical codes represent the same chemical compound.



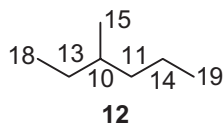
For two permutation labelings giving the same matrix or code, two vertices with the same label are automorphic. In the case of isopropylcyclobutane **8** the two labelings **9** and **10** have identical adjacency matrixes, showing that two pairs of vertices are automorphic, namely vertices ( $a$   $c$ ) and ( $e$   $g$ ). A rigorous method to derive the canonical code and automorphism partitioning is to construct all  $N!$  permutations, to generate their codes and compare them to extract the one-to-one correspondence as presented in the above example. The permutation labelings corresponding to the canonical code are identified by a lexicographical comparison of the  $N!$  codes, followed by the selection of the maximal (or minimal) code. The process of generation of the canonical code by investigating automorphism permutations is called canonical code generation by automorphism permutation (CCAP).

All coding algorithms use a heuristic approach to reduce the number of permutation labelings that have to be investigated in order to detect the canonical labelings, however. Any vertex invariant can be used to obtain a preliminary partition of the vertices from a molecular graph. Consider a vertex invariant of graph  $G$  with  $N$  vertices,  $In = (In_1, In_2, \dots, In_N)$ , which assigns a value  $In_i$  to vertex  $v_i$ . A vertex graph invariant is any vertex property, computed on the basis of the graph structure, whose value does not depend on the graph labeling. Examples of vertex invariants are the degree and distance sum. A partitioning of the vertex set  $V$  is induced by the invariant  $In$  by including vertices  $v_i$  and  $v_j$  in the same atom invariant class (AIC) if  $In_i = In_j$ ; the number of vertices in the class  $i$  is denoted by  $n_i$ . Two vertices from different AIC cannot be automorphic, while two vertices from the same AIC are not necessarily automorphic. Despite numerous efforts, no vertex graph invariant is known which is sufficient to establish the automorphism partitioning, because for certain graphs non-automorphic vertices are partitioned in the same class. The process of atom partitioning in AIC induced by a certain atomic invariant is called graph invariant atom partitioning (GIAP), and represents an important step in the generation of the canonical code.

To determine the canonical code and automorphism partitioning, each AIC resulting from GIAP procedure is investigated to detect non-automorphic vertices by using the definition of automorphism. The partitioning of vertices in  $k$  classes, with  $n_1, n_2, \dots, n_k$  vertices in each class, reduces the complexity of canonical coding of a graph with  $N$  vertices from  $N!$  to the construction of  $n_1!n_2! \dots n_k! = \prod n_i!$  permutation labelings. The classes of vertices are ordered with some rules, then vertices in the first class receive labels  $1, 2, \dots, n_1$ , vertices in the second class are labeled  $n_1 + 1, n_1 + 2, \dots, n_1 + n_2$ , and so on.



The efficiency of the GIAP depends on the discriminating power of the atomic invariant. Consider the molecular graph of 3-methylhexane **11** and its vertex partitioning into three classes induced by atom degrees: three atoms with degree 1, three atoms with degree two, and one atom with degree 3. The graph **11** is an identity tree because all vertices are topologically distinct. The brute force determination of the canonical code and constitutional symmetry of 3-methylhexane **11** requires the comparison of the codes generated by  $7! = 5040$  permutation labelings. The atom degree partitioning reduces the number of labelings needed to determine the constitutional symmetry to  $3!3!1! = 36$ .



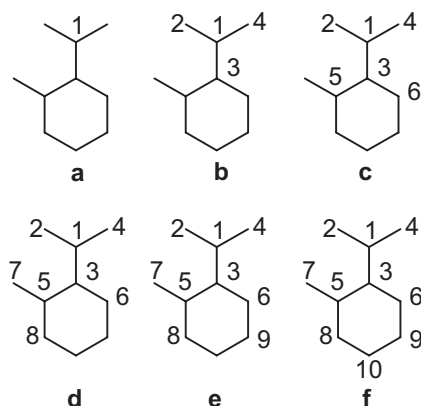


Fig. 5.1-4 The cooperative labeling of 1-isopropyl-2-methylcyclohexane

This number can be further reduced by the use of more powerful vertex invariants, such as the distance sum. The distance sum,  $DS_i$ , of a vertex  $v_i$  is the sum of the topological distances from vertex  $v_i$  to all other vertices in the molecular graph. The partitioning in  $DS$  equivalence classes of the atoms from 3-methylhexane is presented in 12, showing that all atoms have distinct  $DS$  values. Using this partitioning, the constitutional symmetry of 3-methylhexane can be determined with one labeling. The use of vertex graph invariants in determining the constitutional symmetry is hampered by the degeneracy (i.e., two or more topologically distinct vertices have identical numerical values for a certain vertex graph invariants) of almost all known graph invariants.

The cooperative labeling used by Morgan in his canonical coding algorithm [8] restricts the enormous number of labelings investigated in order to find the canonical code of graph  $G$ . In this procedure an arbitrary vertex of  $G$  is selected and labeled by 1. The  $r$  vertices adjacent with vertex 1 are labeled by 2, 3, ...,  $r + 1$  in an arbitrary combination. In the next step the vertex with index 2 is considered and its  $s$  adjacent vertices (not labeled yet) are labeled by  $r + 2, r + 3, \dots, r + s + 1$ . The labeling process continues until all vertices are labeled. The process of cooperative labeling is illustrated in Figure 5.1-4 for 1-isopropyl-2-methylcyclohexane.

The use of the cooperative indexing can reduce further the number of labelings generated in the process of canonical coding by stopping the exploration of a branch as soon as it becomes clear that the corresponding code is not the canonical one. Consider that the minimal code of a chemical structure is searched for; the code is generated during the labeling process and at each step it is compared with the current minimal code stored. If the investigated labeling gives a partial code that is greater than the minimal one, the branch is not further investigated and other branch is explored.

Many coding algorithms used in chemistry use a combination of cooperative labeling and atom partitioning with a discriminant graph invariant. An algorithm for canonical coding can be separated in two steps:

1. GIAP: compute a discriminant atom invariant and establish with it an initial atom partitioning.
2. CCAP: using the atom partitioning established in the first step identify the canonical code by investigating all automorphism permutations.

The use of the GIAP step is based on the property that two atoms with distinct values for the same invariant cannot be automorphic. On the other hand, the assumption that atoms in the same GIAP class are automorphic is not correct. Carhart pointed out that a rigorous canonical coding and constitutional symmetry perception algorithm must contain both GIAP and CCAP steps [9]. The same idea was formulated for the problem of graph isomorphism by Read and Corneil [2]. Even after these warnings, the illusion of obtaining a “better” and “faster” canonical coding algorithm by eliminating the step CCAP continued and one can still find in the literature papers that propose algorithms for the computation of vertex invariants, with the wrong assumption that vertices with identical values of the invariant are automorphic and belong to the same orbit [10–13]. Even when such AIC algorithms give vertex partitionings that coincide with the automorphism partitioning for a certain set of graphs, this fact is not a demonstration that the algorithm can generate the automorphism partitioning for any graph. This type of “canonical coding” algorithm is incomplete, and its use in a chemical database, in synthesis design or structure elucidation systems is unreliable.

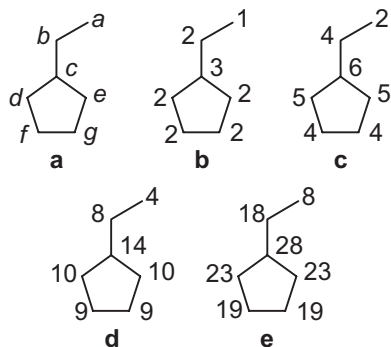
### 5.1.5

#### The MORGAN Algorithm

The extended connectivity algorithm (EC) defined by Morgan is the first efficient algorithm for the partitioning of atoms in equivalence classes [8]. In a modified form, it is used at the CAS in the chemical registry system in order to generate a unique, canonical code of a chemical compound. The EC algorithm is the basis of an important branch of methods for graph partitioning used in various coding methods. The EC algorithm attempts to make a graph partitioning by considering the connectivity of adjacent atoms:

1. Set the level 1 EC of each atom to the value of its degree.
2. Determine the number of different EC<sup>1</sup> values, NECV<sup>1</sup>.
3. The level  $n + 1$  EC of each atom is equal to the sum of EC <sup>$n$</sup>  values of the adjacent atoms.
4. Determine NECV <sup>$n+1$</sup> .
5. If NECV <sup>$n+1$</sup>  > NECV <sup>$n$</sup>  go to step (3).
6. The EC <sup>$n$</sup>  values are the final ones.

While the EC algorithm presented by Morgan does not always allow the complete classification of atoms in equivalence classes, it gives a good starting point for the generation of the canonical code. The calculation of the extended connectivity



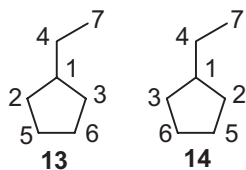
**Fig. 5.1-5** Computation of the extended connectivity values with the Morgan algorithm for ethylcyclopentane (a). The level 1–4 EC values are presented in diagrams b–e

values is illustrated in Figure 5.1-5 for the molecular graph of ethylcyclopentane. Level 1 to 4 EC are presented in Figure 5.1-5b–e.

The final partition induced by the EC<sup>n</sup> values is used for the cooperative labeling of the molecular graph:

1. The atom with the highest EC value is labeled with 1 and is considered the current atom.
2. If the current atom has any unlabeled adjacent atom then assign the next label to it. If the current atom has more than one unlabeled adjacent atoms select the one with the highest EC value. Repeat step (2) until all atoms adjacent to the current atom are labeled.
3. Stop if the graph is completely labeled, else increase the label of the current atom and go to step (2).

Using the above rules and the EC values from Figure 5.1-5e one obtains two cooperative labelings of ethylcyclopentane, namely **13** and **14**. The Morgan EC procedure described above is able to reduce the search for a canonical labeling of ethylcyclopentane from  $7! = 5040$  to only two labelings.



In the final stage of the Morgan algorithm the molecular structure is transformed into a linear code formed by several lists. The FROM list contains for every atom the label of the atom from which it was labeled. This list describes a spanning tree of the molecular graph. The RING-CLOSURE list defines the ring closure bonds by the labels of the atoms connected by the bonds. The atom types are

specified in the ATOM-TYPE list following the order of their labels. The BOND-TYPE lists the bond types in the order in which the bonds were defined in the FROM and RING-CLOSURE lists. Morgan defined as canonical code the minimal code obtained from the above four lists. The canonical code is determined with a CCAP procedure, with a rigorous examination of all cooperative labelings that can be canonical.

The two cooperative labelings **13** and **14** give two identical Morgan codes, representing the canonical code of ethylcyclopentane:

```
FROM: 1 1 1 2 3 4
RING-CLOSURE: 5 6
ATOM-TYPE: C C C C C C
BOND-TYPE: 1 1 1 1 1 1
```

Because both labelings 13 and 14 give identical codes, two vertices with the same label are automorphic. This property allows one to generate the orbits of ethylcyclopentane, using the notation from Figure 5.1-5a:  $X_1 = \{a\}$ ,  $X_2 = \{b\}$ ,  $X_3 = \{c\}$ ,  $X_4 = \{d, e\}$ ,  $X_5 = \{f, g\}$ .

### 5.1.6

#### The Augmented Connectivity Molecular Formula

CAS uses a chemical registry system that is a computer system for the unique representation of the molecular structure in a connection table. The chemical registry system generates the canonical connection table with the algorithm defined by Morgan. The CAS registry system uses also a compact and easy calculated numerical representation of a chemical structure called the augmented connectivity molecular formula (ACMF) [14]. The ACMF molecular representation is a quick and simple way to determine if a compound is absent from the database: if the ACMF of a compound is not found in the database, the compound is new. On the other hand, because the ACMF is not a canonical representation of the molecular structure, two different molecules can have identical ACMF. The ACMF algorithm consists of the following steps:

1. Assign to each atom in the molecular graph a level 1 value that depends on its chemical nature. The elements are considered in alphabetic order, each element having a value  $A_v$  between 32 and 240: actinium, 32; carbon, 60; hydrogen, 106; lutetium, 130; lawrencium, 132; oxygen, 158; zirconium, 240.
2. Assign to each bond in the molecular graph a value that depends on its type. The bonds are numerically characterized by bond values  $B_v$  which are prime numbers: cyclic single bond, 3; cyclic double bond, 5; cyclic tautomer bond, 7; cyclic delocalized bond, 11; cyclic alternating bond, 13; cyclic triple bond, 17; acyclic single bond, 19; acyclic double bond, 23; acyclic tautomer bond, 29; acyclic delocalized bond, 31; acyclic triple bond, 37.

- Assign to each atom in the molecular graph a level 2 augmented connectivity value equal to (Eq. (3)):

$$AC_i^2 = \sum_j Av(v_j)Bv(e_{ij}) \quad (3)$$

where the summation goes over all atoms  $j$  adjacent with atom  $i$ ,  $Ac(v_j)$  represents the Av parameter of atom  $j$  and  $Bv(e_{ij})$  represents the Bv parameter of the bond between atoms  $i$  and  $j$ .

- The  $AC^{n+1}$  values are obtained from the set of  $AC^n$  values (Eq. (4)):

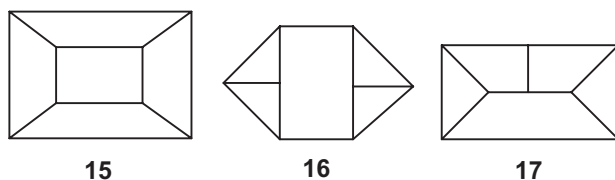
$$AC_i^{n+1} = \sum_j AC_j^n \quad (4)$$

where the summation goes over all atoms adjacent with atom  $i$ .

- If  $n < 4$ , increment  $n$  by 1 and go to step (4).
- Denote by  $NAC^n$  the number of distinct  $AC^n$  values; if  $NAC^n < NAC^{n+1}$ , increment  $n$  by 1 and go to step (4).
- Assign as final atomic value the  $AC^n$  value.

In the following steps the ACMF linear representation of the molecular structure is obtained on the basis of  $AC^n$  values by concatenating them and generating a 64-bit hash code that represents the final ACMF description of the chemical structure. The calculation of the augmented connectivity values is illustrated in Figure 5.1-6 for the molecular graph of 2-cyclopenten-1-ylacetic acid.

The level 1 values (Av parameters) associated with each non-hydrogen atom are presented in Figure 5.1-6a, the Bv parameters are presented in Figure 5.1-6b, while Figure 5.1-6c presents the  $AC^2$  values computed by summing the products of level 1 atom values and the corresponding bond value for all adjacent atoms. Level 3, 4, and 5 AC values are computed by summing the AC values of the previous level for all adjacent atoms; the corresponding diagrams are presented in Figure 5.1-6d–f. The number of distinct values at level 1,  $NAC^1$ , is 2 and their number increases to 8 for  $NAC^4$  and  $NAC^5$ . Because  $NAC^4 = NAC^5$  the algorithm converged to a stable partition and the AC values from level 4 are used to generate the ACMF linear representation of 2-cyclopenten-1-ylacetic acid. The ACMF algorithm is not able to distinguish all non-equivalent atoms, as computed for the two oxygen atoms in Figure 5.1-6d–f, although at level 2 they are characterized by different  $AC^2$  values.



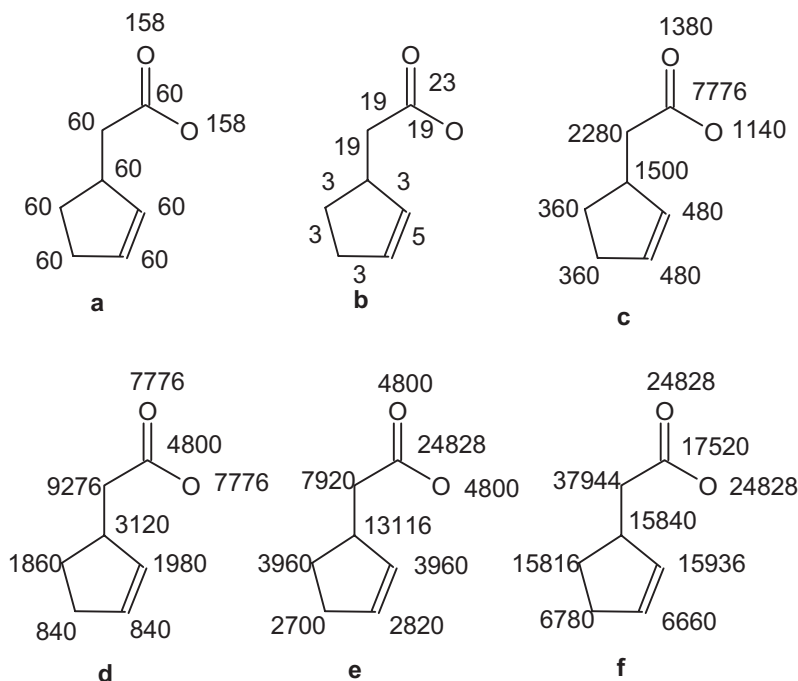


Fig. 5.1-6 Computation of the augmented connectivity molecular formula for 2-cyclopenten-1-ylacetic acid

Because the ACMF algorithm considers only structural information of adjacent atoms, all atoms in a cubic molecular graph will have identical  $AC^k$  values, for all  $k$  levels. Therefore, the  $(CH)_{2k}$  saturated valence isomers of annulenes, represented by cubic molecular graphs, will have identical augmented connectivity for all atoms, giving an identical ACMF representation for the whole set of  $(CH)_{2k}$  isomers with a given  $k$ . The three  $(CH)_8$  isomers 15, 16, and 17 are characterized by identical ACMFs, showing an important limitation of this procedure. Another important class of chemical compounds, fullerenes, are not discriminated by the ACMF algorithm. In a fullerene all carbon atoms are characterized by identical AC values, and as a consequence of this fact all fullerene isomers with a given number of carbon atoms have identical ACMFs. This problem is common to all GIAP algorithms derived from the EC procedure that use only the connectivity information of adjacent atoms.

Although the ACMF is not a unique molecular representation, it has a good discriminating power and from the CAS database that contains several million chemical compounds only less than 0.04% of them represent distinct substances with identical ACMFs. ACMF is a molecular graph invariant, whose value is independent of a particular labeling. In the CAS Chemical Registry System the ACMF is used to determine if a chemical compound is new and its ACMF is not found in the database. This algorithm represents a practical tool for the selection from a



chemical database of the compounds that have identical ACMFs with a potential new compound, and the comparison of the unique canonical code is performed only for this small subset of molecules from the entire database.

### 5.1.7

#### Modifications of the Extended Connectivity Algorithm

The EC method is by far the most investigated GIAP algorithm, and many chemical information systems use this atom partitioning method in various implementations that improve its discriminant power.

Wipke and Dyott have modified the EC algorithm and used it in their simulation and evaluation of chemical synthesis (SECS) program [15]. In this implementation the EC values are computed as in the Morgan definition, unless for a primary atom, in which case its EC value is 1. The coding system proposed by Wipke and Dyott, called stereochemical extension of morgan algorithm (SEMA) has two strings coding the stereochemical information: the *double bond configuration* list and the *atom configuration* list. The canonical code is formed by all lists appended linearly together and gives distinct names for stereoisomers.

The GIAP method defined by Morgan fails sometimes to distinguish between topologically non-equivalent atoms. Some causes of the degeneracy of EC descriptors were identified and some improved algorithms were defined. Consider the EC values of the atoms *b* and *f* in ethylcyclopentane (Figure 5.1-5a). Although the two atoms are topologically distinct, they are characterized by identical EC values in all iterations. By inspecting the EC diagrams in Figure 5.1-5b–e it is clear that the degenerate EC values are obtained by summing different terms, e.g.,  $EC^2_b = 1 + 3 = 4$  and  $EC^2_f = 2 + 2 = 4$ . The canonical coding and constitutional symmetry algorithm developed by Shelley and Munk [16, 17] and used in the CASE (computer-assisted structure elucidation) system substitutes the summing of adjacent atoms EC values with an ordered list of EC values. Their algorithm is described below:

1. Assign a Class Identifier (CI) to each non-hydrogen atom. The CI value is a two-digit integer, the first one being equal with the degree of the atom and the second designates atom type, i.e., C = 2, N = 3, O = 4.
2. Count the number of different CI values, NCI, and assign new CI values between one and NCI to each atom. The atoms with the smallest CI value receive label one and the atoms with the largest CI value receive label NCI.
3. If NCI is equal to the number of non-hydrogen atoms then go to (7).
4. Assign a trial class identifier (TCI) to each atom. The TCI is a string of five two-digit integers with the leftmost field containing the CI of the respective atom. The remaining four fields contain a list of the adjacent atoms CIs in descending order from right to left. If there are less than four adjacent atoms then the list is zero filled.
5. Count the number of different TCI values (NTCI) and assign new TCI values

- between one and NTCI to each atom: the atoms with the smallest TCI value receive label one and the atoms with the largest TCI value receive label NTCI.
- If NTCI is not greater than NCI then go to (7), else set the CI of each atom to its TCI value and set NCI to NTCI.
  - The GIAP is finished.

A similar algorithm, HOC (hierarchically ordered extended connectivities), was proposed by Balaban, Mekenyan and Bonchev [18]. The HOC algorithm considers also the stereochemical information [19], and is followed by a CCAP algorithm that provides the canonical code. Moreau defined a discriminant EC algorithm, but disregarded the CCAP procedure [10]. Another implementation of the Morgan algorithm was proposed by Figueras [20]. A modified Morgan algorithm was used for the determination of topological equivalence classes of atoms and bonds in  $C_{20}$ – $C_{60}$  fullerenes [21].

### 5.1.8

#### Other Symmetry Perception Algorithms

Many other GIAP and canonical coding algorithms were proposed both in chemical and mathematical literature. A succinct enumeration of them is presented in this section.

For the computer manipulation of organic reactions and reaction mechanisms Fujita defined the concept of imaginary transition structure (ITS), which is a structural formula obtained by the superposition of the molecules of the reagents and products [22, 23]. Each ITS can be numerically represented as a connection table and transformed into a canonical code which can be manipulated by a computer and stored in databases.

A complete algorithm for partitioning the atoms in a molecule into equivalence classes was introduced by Jochum and Gasteiger using the NOON (number of outermost occupied neighbor) sphere of an atom defined as the minimal number of neighbor spheres necessary to accommodate all atoms of a molecule starting at that atom [24]. A related algorithm, the atom environment matrix method, was developed by Bersohn for partitioning the atoms into equivalence classes [25].

A group of efficient canonical coding algorithms were developed starting from the adjacency matrix. The adjacency matrix of a molecular graph is not a graph invariant, because it depends on the particular labeling of the chemical structure. Randić defined the UAC (upper adjacency code) notation of the labeled graph  $G$  as a string composed of  $N(N-1)/2$  digits of the upper triangle of the adjacency matrix  $A$ :  $UAC = (a_{12} a_{13} a_{14} \dots a_{1N} a_{23} \dots a_{NN-1})$  [26, 27]. From the UAC code the adjacency matrix can be unambiguously reconstructed. Different labelings of the molecular graph generate different UAC strings. SUAC (smallest upper adjacency code) is a unique representation of the adjacency matrix, which was proposed by Randić as an efficient method for canonical labeling, and perception of constitutional symmetry.

The system *Chemics*, developed for the automated structure elucidation of organic compounds, uses the connectivity stack to generate all possible chemical structures that are consistent with given structural information [28]. The connectivity stack uses a notation for each substructure, e.g., C3 for a methyl and LC for chlorine, and with a set of rules the molecule is linearly coded into a list of substructures and a list of connections. The canonical code is defined as the labeling which offers the maximal linear representation of the upper triangle of the adjacency matrix.

The *Syngen* synthesis design program developed by Hendrickson uses another molecular code generated from UAC, namely the greatest upper adjacency code (GUAC), which is a canonical labeling of the molecular graph giving the maximal UAC representation of the upper triangle of the adjacency matrix [29]. GUAC is used to establish a catalog of available starting material molecules and to give a unique representation to the molecules generated by the program. In practical use, there are some advantages in using GUAC over SUAC code because the GUAC labeling is cooperative and there are fewer atoms with a maximal valence than atoms with a minimal valence. Therefore GUAC makes fewer trials than SUAC and requires less time to code the graph and uses less memory space.

Kvasnicka and Pospichal introduced an algorithm of canonical indexing for the non-redundant and exhaustive constructive enumeration of graphs [30, 31]. Their algorithm uses the LAC (lower adjacency code) notation of the labeled graph  $G$  which is a string composed of  $N(N+1)/2$  digits of the lower triangle of the adjacency matrix  $A$ :  $LAC = (a_{11} a_{21} a_{22} \dots a_{N1} \dots a_{NN})$ . Unlike UAC, the LAC notation contains the diagonal of the adjacency matrix. The greatest lower adjacency code (GLAC) is generated by the canonical labelings of  $G$  that give a maximal LAC.

To obtain canonical codes of fullerenes, Liu and Klein introduced a vertex invariant defined on the basis of matrix spectral theory [32]. Denote by  $\lambda$  an eigenvalue of the adjacency matrix  $A$  and by  $|s\lambda\rangle$  the corresponding set of orthonormal eigenvectors, where  $s$  is a degeneracy index with values from 1 to the degeneracy  $g(\lambda)$  of the eigenvalue  $\lambda$ . Then (Eq. (5)):

$$A|s\lambda\rangle = \lambda|s\lambda\rangle \quad (5)$$

and the projection operator for the  $\lambda$ th eigenspace is (Eq. (6)):

$$P(\lambda) = \sum_{s=1}^{g(\lambda)} |s\lambda\rangle\langle s\lambda| \quad (6)$$

where  $\langle s\lambda|$  is the row-vector corresponding to the column vector  $|s\lambda\rangle$ , and  $(a|b)$  represents the inner product between two vectors  $|a\rangle$  and  $|b\rangle$ . The invariant  $T$ , with row  $i$  representing the vertex  $i$  and column  $\lambda$  representing the eigenvalue  $\lambda$ , is (Eq. (7)):

$$\mathbf{T}_{ii} = (i|\mathbf{P}(\lambda)|i) = \sum_{s=1}^{g(\lambda)} |(i|s\lambda)|^2 \quad (7)$$

where  $|i\rangle$  is a vector with the  $i$ th element equal to 1 and all other equal to 0. The rows of the matrix  $\mathbf{T}$  are ordered lexicographically to obtain a vertex invariant.

A hash code is a fixed length representation of a data structure used as an index or key to a direct access file. The input structure cannot be restored from a hash code, and due to the limited range of values two different data structures may be represented by the same hash code. Ihlenfeldt and Gasteiger proposed to represent chemical structures with hash codes by using a hierarchical algorithm: atom hash codes are computed first, merged into molecule hash codes and the molecule hash codes are combined to give a molecular ensemble hash code [33]. The input of the algorithm is a connection table, and a prime number table is used to select initial values for atoms. The atom hash codes contain information on the number of hydrogen and non-hydrogen neighbor atoms, the atomic number  $Z$ , number of atoms in molecule modulo 257, stereochemical descriptors, isotope information,  $\pi$ -system size, and charge. In the next step the hash codes of the atoms are combined with the hash codes of the neighbors by rotation and exclusive-or operations applied on the 64-bit representation. Using simple bit operations atom hash codes are combined to yield molecule and ensemble hash codes. The hash codes were used to classify chemical catalogs, in the Wodca synthesis planning system [34], in the Eros 6 reaction prediction program [35, 36], and in the Massimo mass spectra prediction system [37].

Uchino used the matrix multiplication method for obtaining the canonical code and automorphisms of a molecular graph [38]. He considered the adjacency, distance, and open walks matrixes in a series of efficient algorithms that offer the automorphism partition of graphs. The incidence matrix of molecular hypergraphs was used for the canonical coding of non-classical molecular structures with polycentric delocalized bonds [39, 40]. The canonical adjacency matrix was used in the MOLGEN program for the generation of constitutional and configurational isomers [41]. Projection operators of graph spectra were proposed for canonical coding [42]. The canonical coding algorithm of Schubert and Ugi [43] was implemented in synthesis design systems [44, 45]. Graph-matching algorithms were found useful for automorphism, isomorphism, and maximal common substructure determination [46, 47]. The algorithms developed for determining constitutional symmetry were extended to obtain the three-dimensional symmetry [48]. New results were obtained for the coding of configurational isomers [49].

Faulon developed polynomial-time algorithms for the problems of isomorphism, automorphism partitioning, and canonical labeling of molecular graphs, which represent a particular class of graphs for which these problems can be efficiently solved [50]. Satoh et al. introduced a canonical coding method for representing three-dimensional structures, CAST (canonical representation of stereochemistry), which can differentiate between conformers, enantiomers, or diastereomers [51, 52]. The illusory enterprise of obtaining a canonical coding algorithm that contain

only the GIAP step continues, and Ouyang et al. have proposed a modified Morgan EC algorithm for “topological symmetry perception and unique numbering” of atoms [53]. However, their algorithm is a simple graph invariant atom partitioning method, without the CCAP step, which makes it unfit for the “canonical coding algorithm” label. Even though such algorithms are able to correctly identify the canonical labeling of a large number of graphs, they are not complete and fool-proof. Also, Ouyang et al. continues the series of false assertions stating that the Morgan algorithm is not suitable for canonical labeling of molecular graphs [53]. In fact, the original Morgan algorithm contains the two steps, GIAP and CCAP, required for a canonical coding algorithm [8]: the GIAP step consists of the EC algorithm, while in the CCAP step the final EC atom partition is used to generate all possible connection tables, and the labeling that gives the minimal code represents the canonical labeling. Although the EC algorithm is not very efficient, overall the Morgan algorithm is demonstrated to always give the canonical code.

Contreras et al. developed a stereoisomer generation system, Camgec2, using a linear coding system that encodes also the stereo information [54]. A problem related to graph isomorphism is the detection of the maximal common substructure between two molecules. Raymond et al. proposed a molecular similarity measure based on the maximal common edge subgraph [55].

### 5.1.9

#### Conclusions

The determination of the constitutional symmetry of chemical compounds is a highly investigated research topic, and the various automorphism partitioning algorithms presented in this section show the evolution from simple techniques to methods with a strong mathematical background. The partitioning of atoms in orbits represents the common starting point for three structure representation and manipulation techniques: graph isomorphism, determination of the automorphism group, and canonical coding.

The algorithms for automorphism partitioning of atoms consist of two steps: (1) an atom invariant is used to establish an initial atom partitioning (GIAP); (2) the GIAP result is used to establish the automorphism partition by investigating all automorphism permutations. The Morgan extended connectivity algorithm [8] inspired a large number of variants, but the method fails to discriminate atoms in fullerenes,  $(\text{CH})_{2k}$  saturated valence isomers of annulenes, and other regular graphs. More powerful algorithms, many inspired from vertex topological indices, are presently used to obtain a better atom partitioning based on graph invariants. A balance must be maintained between the efficiency and the computational complexity of the GIAP algorithm: methods that solve a system of linear equations, use matrix multiplication, or compute matrix eigenvectors can be a good choice, but path enumeration algorithms are not efficient.

The selection of the canonical code definition is equally important: the use of the cooperative labeling can greatly reduce the number of labelings investigated in the

CCAP step. A canonical code is generated and manipulated by a computer, and its definition and representation must consider the data structures offered by the programming languages.

## References

- O. IVANCIUC, Canonical Numbering and Constitutional Symmetry, in: *The Encyclopedia of Computational Chemistry*, P. v. R. SCHLEYER, N. L. ALLINGER, T. CLARK, J. GASTEIGER, P. A. KOLLMAN, H. F. SCHAEFER III, AND P. R. SCHREINER (Eds.), John Wiley and Sons, Chichester, 1998, pp. 167–183.
- R. C. READ, D. G. CORNEIL, *J. Graph Theor.* 1977, 1, 339–363.
- I. GUTMAN, O. E. POLANSKY, *Mathematical Concepts in Organic Chemistry*; Springer, Berlin, 1986, Chapter 9, 108–116.
- M. RAZINGER, K. BALASUBRAMANIAN, M. E. MUNK, *J. Chem. Inf. Comput. Sci.* 1993, 33, 197–201.
- S. BOHANEC, M. PERDIH, *J. Chem. Inf. Comput. Sci.* 1993, 33, 719–726.
- K. BALASUBRAMANIAN, *J. Chem. Inf. Comput. Sci.* 1994, 34, 1146–1150.
- K. BALASUBRAMANIAN, *J. Chem. Inf. Comput. Sci.*, 1995, 35, 761–770.
- H. L. MORGAN, *J. Chem. Doc.* 1965, 5, 107–113.
- R. E. CARHART, *J. Chem. Inf. Comput. Sci.* 1978, 18, 108–110.
- G. MOREAU, *Nouv. J. Chim.* 1980, 4, 17–22.
- C. Y. HU, L. XU, *Anal. Chim. Acta* 1994, 295, 127–134.
- C.-Y. HU, L. XU, *J. Chem. Inf. Comput. Sci.* 1994, 34, 840–844.
- H. HONG, X. XIN, *J. Chem. Inf. Comput. Sci.* 1994, 34, 730–734.
- R. G. FREELAND, S. A. FUNK, L. J. O'KORN, G. A. WILSON, *J. Chem. Inf. Comput. Sci.* 1979, 19, 94–98.
- W. T. WIPKE, T. M. DYOTT, *J. Am. Chem. Soc.* 1974, 96, 4834–4842.
- C. A. SHELLEY, M. J. MUNK, *J. Chem. Inf. Comput. Sci.* 1977, 17, 110–113.
- C. A. SHELLEY, M. J. MUNK, *J. Chem. Inf. Comput. Sci.* 1979, 19, 247–250.
- A. T. BALABAN, O. MEKENYAN, D. BONCHEV, *J. Comput. Chem.* 1985, 6, 538–551.
- A. T. BALABAN, O. MEKENYAN, D. BONCHEV, *J. Comput. Chem.* 1985, 6, 562–569.
- J. FIGUERAS, *J. Chem. Inf. Comput. Sci.* 1992, 32, 153–157.
- T. LAIDBOEUR, D. CABROL-BASS, O. IVANCIUC, *J. Chem. Inf. Comput. Sci.* 1996, 36, 811–821.
- S. FUJITA, *J. Chem. Inf. Comput. Sci.* 1986, 26, 205–212.
- S. FUJITA, *J. Chem. Inf. Comput. Sci.* 1988, 28, 128–137.
- C. JOCHUM, J. GASTEIGER, *J. Chem. Inf. Comput. Sci.* 1979, 19, 49–50.
- M. BERSOHN, *Comput. Chem.* 1987, 11, 67–72.
- M. RANDIĆ, *J. Chem. Phys.* 1974, 60, 3920–3928.
- M. RANDIĆ, G. M. BRISSEY, C. L. WILKINS, *J. Chem. Inf. Comput. Sci.* 1981, 21, 52–59.
- H. ABE, Y. KUDO, T. YAMASAKI, K. TANAKA, M. SASAKI, S. SASAKI, *J. Chem. Inf. Comput. Sci.* 1984, 24, 212–216.
- J. B. HENDRICKSON, A. G. TOCZKO, *J. Chem. Inf. Comput. Sci.* 1983, 23, 171–177.
- V. KVASNICKA, J. POSPICHAL, *J. Chem. Inf. Comput. Sci.* 1990, 30, 99–105.
- V. KVASNICKA, J. POSPICHAL, *J. Math. Chem.* 1992, 9, 181–196.
- X. LIU, D. J. KLEIN, *J. Comput. Chem.* 1991, 12, 1243–1251.
- W. D. IHLENFELDT, J. GASTEIGER, *J. Comput. Chem.* 1994, 15, 793–813.
- J. GASTEIGER, W. D. IHLENFELDT, R. FICK, J. R. ROSE, *J. Chem. Inf. Comput. Sci.* 1992, 32, 700–712.
- J. GASTEIGER, W. D. IHLENFELDT, P. RÖSE, R. WANKE, *Anal. Chim. Acta* 1990, 235, 65–75.
- P. RÖSE, J. GASTEIGER, *Anal. Chim. Acta* 1990, 235, 163–168.

- 37 J. GASTEIGER, W. HANEBECK, K.-P. SCHULZ, *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 264–271.
- 38 M. UCHINO, *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 201–206.
- 39 E. V. KONSTANTINOVA, V. A. SKOROBOGATOV, *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 472–478.
- 40 E. V. KONSTANTINOVA, V. A. SKOROBOGATOV, *Discr. Math.* **2001**, *235*, 365–383.
- 41 T. WIELAND, A. KERBER, R. LAUE, *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 413–419.
- 42 I. V. STANKEVICH, E. G. GAL'PERN, A. L. CHISTYAKOV, I. I. BASKIN, M. I. SKVORTSOVA, N. S. ZEFIROV, O. B. TOMILIN, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 1105–1108.
- 43 W. SCHUBERT, I. UGI, *J. Am. Chem. Soc.* **1978**, *100*, 37–41.
- 44 I. UGI, J. BAUER, J. BRANDT, J. FRIEDRICH, J. GASTEIGER, C. JOCHUM, W. SCHUBERT, *Angew. Chem. Int. Ed. Engl.* **1979**, *18*, 111–123.
- 45 I. UGI, J. BAUER, C. BLOMBERGER, J. BRANDT, A. DIETZ, E. FONTAIN, B. GRUBER, A. VON SCHOLLEY-PFAB, A. SENFF, N. STEIN, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 3–16.
- 46 H. SCSIBRANY, K. VARMUZA, *ToSiM: PC-Software for the Investigation of Topological Similarities in Molecules*. C. JOCHUM (ed.), Software-Development in Chemistry 8; Gesellschaft Deutscher Chemiker: Frankfurt am Main, 1994, 235–249.
- 47 J. XU, *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 25–34.
- 48 T. LAIDBOEUR, D. CABROL-BASS, O. IVANCIUC, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 87–91.
- 49 M. RAZINGER, M. PERDIH, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 290–296.
- 50 J.-L. FAULON, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 432–444.
- 51 H. SATOH, H. KOSHINO, K. FUNATSU, T. NAKATA, *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 622–630.
- 52 H. SATOH, H. KOSHINO, K. FUNATSU, T. NAKATA, *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 1106–1112.
- 53 Z. OUYANG, S. YUAN, J. BRANDT, C. ZHENG, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 299–303.
- 54 M. L. CONTRERAS, J. ALVAREZ, M. RIVEROS, G. ARIAS, R. ROZAS, *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 964–977.
- 55 J. W. RAYMOND, E. J. GARDINER, P. WILLETT, *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 305–316.