# The Automation of Structural Group Contribution Methods in the Estimation of Physical Properties

NEIL JOCHELSON,* C. MICHAEL MOHR,** and ROBERT C. REID
Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Mass.

Received January 26, 1968

**The AIChE has developed a computer routine to estimate the properties of materials. A serious limitation of their system is its inability to handle property estimation methods of the structural increment type. The present paper discusses the applicability of using a modified Sussenguth structure-matching algorithm to determine and count relevent subgroups of a molecule for input to such structural increment estimation methods. The program was tested by estimating critical properties and ideal gas functions. Computation times and method inefficiencies are discussed.**

Physical and thermodynamic properties of gases and liquids are of vital concern to the chemical engineer, whether he is involved in research, in development, in plant design, or in production. But for many compounds of interest to the chemical engineer, the only data available to him are the molecular weight and the boiling point, and for some of the newer, more exotic compounds, even the boiling point may not have been reported. Even for well-known compounds, the data available will often not cover the region of interest to the chemical engineer.

It is thus not surprising that in recent years literally thousands of papers have been published which present methods of estimating a variety of properties. Many of the methods are based on theoretical considerations and allow the user to calculate one set of properties from another set already known. Others require only a knowledge of the structure of the compound under consideration and use empirically derived contributions for various atomic or functional subgroups within the molecule. Such contributions are then manipulated algebraically to estimate the required output properties. "Structural increment" methods are extremely useful if no other property data exist for the compound, since calculated output properties can then be in other methods to estimate a variety of other properties.

In a review of the available estimation techniques for the properties of gases and liquids (*23*), recommended methods are shown for various properties depending upon the input data available, the type of compound, and the range of conditions such as temperature and pressure. Many of these methods are of the "structural group increment" type.

To provide an efficient tool by which the chemical engineer can estimate physical properties with a minimum of background knowledge on the multitude of available techniques and with a minimum of repetitive calculations, the Machine Computation Committee of the AIChE initiated in 1960 a project to program the best methods for digital computer operation. The AIChE Physical Property Estimation System was completed in 1965 and is presently in partial use by the industrial sponsors (*1, 3*).

One serious drawback of the present AIChE program is its inability to handle satisfactorily property estimation methods of the structural increment type. It cannot accept the structural formula of a compound and produce required group counts which are necessary in structural increment methods. Indeed, many excellent methods were excluded from the system because they required such detailed structural information.

The problem of counting the subgroups in a molecule by computer is similar in many aspects to the problem of automatic chemical information retrieval. There exists, at present, a large number of automated search-and-match techniques for chemical structures which were primarily developed for chemical information retrieval systems (*20*). Many of these techniques are directly applicable to the problem of counting the number of atomic subgroups in the molecule whose properties are to be estimated and can provide the basis of a computer program for structural group counting. Such a program was recently developed at MIT (*16*).

## THE AIChE PHYSICAL PROPERTY ESTIMATION SYSTEM

The AIChE Physical Property Estimation System is described by Meadows (*19*) and Norris (*21*), and more recently by Heitman and Harris (*13*). It makes extensive use of a "road-mapping" feature to select the optimum route from the input properties to the desired output properties. To accomplish this, the system takes into account the input data that each method requires and the output properties that it generates, as well as those types of chemical compounds for which a method is not applicable. Based on the estimated errors in the input data and the average method errors, the system selects the optimum route to give the desired output properties with minimum error (*3*).

In most cases, the critical properties of a compound are the starting point for the "road map," and if these have not been supplied as input data, they must be estimated by the Lydersen group-contribution method (*18*). In the present form of the system, the user must

* Present address: Caltex Oil (S. A.) Ltd., Milnerton Refinery, Cape Town, South Africa.
** Present address: Arthur D. Little, Inc., Acorn Park, Cambridge, Mass.

supply the Lydersen method structure counts for the molecules under investigation via the input property cards. This is true of the four other methods which are presently included in the system and which use structural group counts as their input.

## CHEMICAL STRUCTURE INFORMATION RETRIEVAL TECHNIQUES

In the field of automated information retrieval, the problem is to extract from a file of compounds (usually stored on magnetic tape) all those that contain a desired chemical substructure. This desired substructure is called a "query group" and may be as simple as a methyl group or as large as a steroid structure. The retrieval system usually produces a bibliographic reference list of all compounds containing the query group in question. To communicate the structure of the compound or query group to the computer, a notation system or structure code is used to encode the chemical structure.

A thorough, comprehensive review of chemical notation systems or structure codes was published in 1964 (*20*), so only the principles of some of the techniques are outlined here.

Chemical notation systems may be classified according to certain qualities the notation bears with respect to the compound. If only one notation is possible for a given compound, the system is unique, while, if more than one notation is possible, the system is nonunique. A notation system is unambiguous if it will regenerate only the original compound and ambiguous if it will regenerate more than one compound.

Chemical notation systems vary considerably in the details of the compound structure that they record. Classification codes simply classify the compound as to its chemical type (homologous series). Code terms are assigned to broad classes of structures, such as heterocyclic, bicyclic, etc. Fragmentation codes are more detailed and specifically list the groups of atoms and bonds present according to a defined set of groups—e.g., $-NH_2$, —COOH, —N=N—NH—. Topological codes are completely detailed in that they assign code terms to each atom and bond present and record the actual interconnection of the various atomic groups in the molecule. If rules for ordering the code symbols are added to a topological code so that the code terms for a given structure must always be cited in the same order, a topological code becomes a unique and unambiguous notation.

Of the many unique and unambiguous codes, the Wiswesser (*31, 32*) and Dyson (IUPAC) (*6, 15*) linear codes are probably the best known. Both systems have received considerable attention and are in use by industrial companies, universities, and information service groups. There are a number of simple rules to learn, but still the effort required is often enough to discourage an infrequent user from mastering the encoding technique.

Generic searches on unique and unambiguous notations are made by programming the computer to scan the notation for the explicit structural fragments present or to analyze the notation for substructures implicit in the notation.

A typical example of the methods involving complete topological matching of compound and query structures is the iterative "node-by-node" searching technique described by Ray and Kirsch in 1957 (*22*). Atoms are called nodes in their paper. The objective is to match each atom in the query structure with a similar atom in the compound structure. To do this, their algorithm proceeds by matching an atom in the query structure with a similar atom in the compound structure. The program then attempts to match the atoms bonded to the first matched atom in the query structure with atoms bonded in the same manner to the first matched atom in the compound structure. If this series of matches is successful, the program proceeds to test whether atoms bonded to these "second level" atoms can be matched. The point is eventually reached at which either all atoms in the two structures are matched, or no further successful matches can be made. When the latter case arises, the program back-tracks, perhaps several levels, until an untried branch is reached. If all possible branches have been tried without success, the program terminates, indicating that the query structure could not be matched. A great deal of back-tracking may be necessary to determine a match or no-match condition. The process of marking matched atoms and erasing those marks during the back-tracking is very time-consuming. The time required to find a no-match condition between two similar structures varies as $2^n$, where $n$ is the number of atoms in the smaller structure.

A method developed by Gluck (*12*) refined the simple topological matching technique to increase the efficiency of the iterative process. The input format of the Gluck system for a typical compound is shown in Figure 1. The atoms are numbered arbitrarily and then listed in numerical order. The atom type—e.g., C, H, O, N—of each is listed as are the numbers of the other atoms to which each atom is connected. The bond types connecting each atom to its neighbors are represented by numbers—e.g., 1 for single bond, 2 for double bond. Each bond is only entered once as the algorithm fills in the corresponding entry. For example, the connection of atom 1 to atom 2 is coded, but the connection of atom 2 to atom 1 need not be. Hydrogen atoms are not listed as the program assumes that unfulfilled valences are attached to hydrogen atoms. The last three atom cards in Figure 1 show the molecular formula by listing the total number of carbon, oxygen, and hydrogen atoms ($C_{11}O_4H_{18}$). The input atom-by-atom list is reordered until a unique form is produced. This unique list is then in storage—e.g., magnetic tape as the structural description of the compound. Ordering in the list is determined by node value (atom type), node connectivity (types of bonds connected to the given atom), and node degree (number of other atoms connected to a given atom). The position of an atom in the list enables the matching procedure to determine whether or not it is possible for the atom to be bonded to the same types of atoms by the same types of bonds, as required by the query structure. The Gluck system has been modified by Chemical Abstracts Service for use in their automated information service programs (*17*).

The Salton and Sussenguth "graph-theoretic" algorithm (*26, 27, 28*) avoids the time-consuming back-tracking inherent in the iterative node-by-node matching technique. It treats a chemical structure as a graph—i.e., a network of nodes (atoms) connected by branches (chemical bonds). To generate sets of atoms which must match in the query

| | Atom | | | Bond 1 | | Bond 2 | | Bond 3 | | Bond 4 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Retrieval Number | No. | Code | Group | Type | Atom no. | Type | Atom no. | Type | Atom no. | Type | Atom no. |
| 1 2 3 4 5 6 7 | 1 | C | | 1 | 2 | | | | | | |
| | 2 | C | | 1 | 3 | | | | | | |
| | 3 | C | | 2 | 4 | | | | | | |
| | 4 | C | | 1 | 5 | | | | | | |
| | 5 | C | | 1 | 6 | 1 | 11 | | | | |
| | 6 | O | | 1 | 7 | | | | | | |
| | 7 | C | | 1 | 8 | 2 | 10 | | | | |
| | 8 | C | | 1 | 9 | | | | | | |
| | 9 | C | | | | | | | | | |
| | 10 | O | | | | | | | | | |
| | 11 | O | | 1 | 12 | | | | | | |
| | 12 | C | | 1 | 13 | 2 | 15 | | | | |
| | 13 | C | | 1 | 14 | | | | | | |
| | 14 | C | | | | | | | | | |
| | 15 | O | | | | | | | | | |
| | 11 | C | 9 | | | | | | | | |
| | 4 | O | 9 | | | | | | | | |
| | 18 | H | 9 | | | | | | | | |

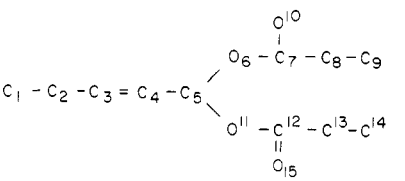



Figure 1. Sample input format for Gluck system (12)

| Card No. | Compound Name | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ACETIC ACID | | | | | | | | | | | |
| | Compound ID No. | | No. of Cards | No. of Atoms | | No. of Rings | | | | | | |
| 2 | 501 | | 8 | 8 | | 0 | | | | | | |
| | Atom No. | Atom Type | Connected to Atom No. | By Bond of Type | Connected to Atom No. | By Bond of Type | Connected to Atom No. | By Bond of Type | Connected to Atom No. | By Bond of Type | Connected to Atom No. | By Bond of Type |
| 3 | 1 | C | 2 | S | 3 | S | 4 | S | 5 | S | | |
| 4 | 2 | H | 1 | S | | | | | | | | |
| 5 | 3 | H | 1 | S | | | | | | | | |
| 6 | 4 | H | 1 | S | | | | | | | | |
| 7 | 5 | C | 1 | S | 6 | D | 7 | S | | | | |
| 8 | 6 | O | 5 | D | | | | | | | | |
| 9 | 7 | O | 5 | S | 8 | S | | | | | | |
| 10 | 8 | H | 7 | S | | | | | | | | |

| | Ring No. | No. of Atoms in Ring |
|---|---|---|
| (11) | Ring | |
| | List of Atoms in Ring 1 | |
| (12) | | |

( Only Present for Ring Compounds )

Bond Code

S = Single
D = Double
DC = Double Cis
DT = Double Trans
R = Resonant
T = Triple
I = Ionic

Figure 2. Sussenguth input format for acetic acid

and compound structures, the algorithm makes use of four properties of these atoms: node value (atom type), node degree (number of atoms connected to a given atom), branch value (bond types associated with a given atom—e.g., single bond, double bond), and node interconnection (other atoms connected to a given atom). The time required to determine a match or no-match condition between two structures is estimated to be proportional to $(n - 1)^2$, where $n$ is the number of atoms in the smaller structure. The input format, which is shown in Figure 2 for acetic acid, is similar to that of Gluck (*12*) and is described in detail later. The Sussenguth method is probably the most suitable existing technique for use in group counting programs.

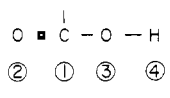
## THE SUSSENGUTH STRUCTURE-MATCHING ALGORITHM

The graph-theoretic algorithm of Sussenguth was used to perform the structural-group matching in group-counting programs recently developed at MIT (*16, 25*). This method was chosen because of its versatility in handling various types of query groups, the simplicity of the input format, and the efficiency in identifying the presence or absence of query structural groups in a compound structure. In this matching technique, the chemical compound is represented by a graph consisting of a set of nodes (equivalent to the chemical atoms) and a set of branches connecting pairs of nodes (which correspond to the chemical bonds joining the atoms). The algorithm determines whether a given query structure is present in the compound structure by attempting to find the nodes in the compound graph which correspond to the nodes of the query graph. Only when all of the query nodes have been so identified in a one-to-one correspondence has the algorithm proved the presence of the query graph as a subgraph of the compound graph.

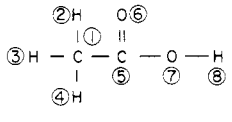
To determine the correspondences between the query and compound nodes, the algorithm uses certain properties of the nodes, such as atomic type (node value), bond type, and atom interconnection, to generate mathematical sets of nodes which are equivalent in the query and compound structures. These sets are then intersected in an attempt to reduce the number of nodes until a one-to-one correspondence is established for each query node.

The following example illustrates the over-all concepts of the algorithm. The problem is to search for the carboxylic acid group in acetic acid. Both the query and compound structures are shown below with the atoms arbitrarily numbered.

Carboxyl Group
(Query Graph)



Acetic Acid
(Compound Graph)



The input format for coding the chemical structure of acetic acid is shown in Figure 2. It is very similar to the Gluck (*12*) method of coding [especially the version adopted by Chemical Abstracts Service (*17*)] and is very easy to learn. The cards are numbered in the leftmost column. The first two cards identify the compound. These are known as header cards. The atoms are then listed showing the atom type of each and the type of bond by which each atom is connected to the other atoms in the structure. The symbols used for coding the allowable bond types (single, double, double-cis, double-trans, resonant, triple, and ionic) are also shown in Figure 2. Provision is also made for listing those atoms which are present in a ring. The second header card indicates the number of rings present. The cards listing the atoms in these rings follow after the last atom card. They are not relevant for acetic acid.

The input structural data are condensed into a compact binary matrix form for storage and searching purposes. Sussenguth used a form which takes advantage of the binary nature of the 7090 class of machine.

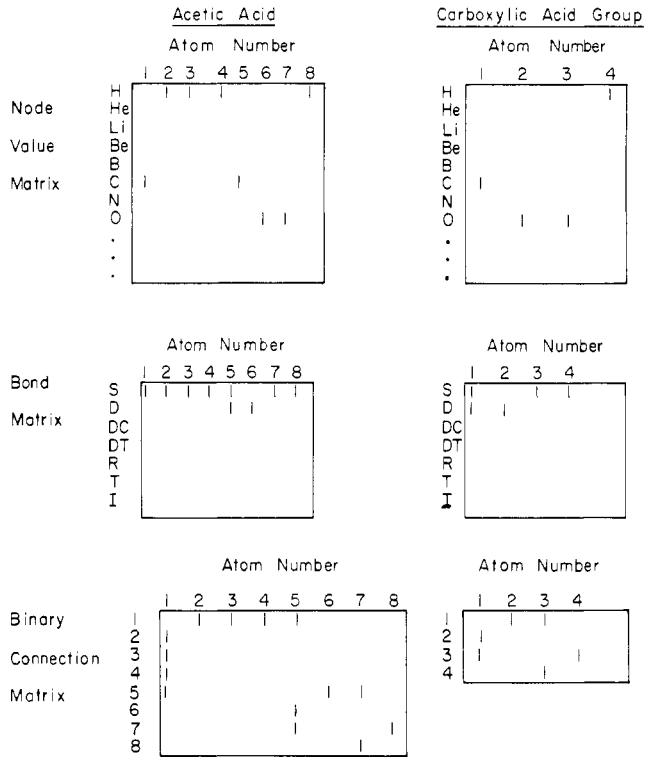



**Figure 3. Node value, bond and connection matrices for acetic acid and carboxylic acid query group**

The atom types of the various nodes are represented in the computer in terms of a node value matrix. Figure 3 shows the node value matrices for acetic acid (as coded in Figure 2) and the carboxylic acid query group. The set of possible node values is the set of chemical elements (H, He, Li, Be, B....). For the first row of the matrix, a binary 1 is marked in the bit positions of those atoms that are hydrogens. Similarly, the second row represents those atoms that are helium atoms, and so on for the rest of the elements in the periodic table.

The types of bonds associated with a given atom are indicated by the bond matrix or branch adjacency matrix. Figure 3 shows an example of the bond matrices for the acetic acid structure, as coded in Figure 2, and the carboxylic acid query group. The different rows of the bond matrix represent, by the bit positions that are 1, those atoms that are involved in one or more single bonds (*S*), double bonds (*D*), double-cis (*DC*), double-trans (*DT*), resonant (*R*), triple (*T*), or ionic (*I*) bonds. For acetic acid, only single and double bonds are significant, and only compound nodes number 5 and 6 are involved in a double bond.

Connectivity of the atoms in the graph is represented by the binary connection matrix, which is illustrated in Figure 3 for acetic acid and the carboxylic acid group. The rows and columns of the connection matrix are numbered according to atom number and the connection of atom *I* to atom *J* is shown by a binary 1 at the intersection of row *I* and column *J* of the matrix, as well as at row *J*, column *I*. For example, row 3 of the matrix shows, by the position of the binary 1's in the row, the atoms connected to atom number 3.

Unfortunately, the use of the bond matrix with only one connection matrix produces a somewhat ambiguous

representation. The connection matrix does not indicate by what type of bond each pair of atoms is connected together, and the bond matrix does not indicate whether a given atom is involved in more than one bond of the type indicated by the relevant row in which it appears (*29*). This ambiguity causes problems in identifying query groups containing conjugated and adjacent double bonds.

The information entered on the RING cards, if any, is saved in a binary ring matrix, where each row represents a different ring and lists the atoms in that ring by the presence of a binary 1 in the relevant bit positions. There are no entries in the ring matrix for acetic acid or the carboxylic acid group. The atoms in a ring could have been determined from the connection matrix, but the ring matrix is much more convenient from a programming standpoint.

The sets generated by the algorithm in attempting to match the carboxylic acid group against acetic acid are shown in Figure 4. Line 1 shows all those atoms in the two structures that are involved in a single bond. Each member of the query set must correspond to one member of the compound set. Line 2 lists all those atoms that are involved in a double bond. Lines 3, 4, and 5 list the atoms that are oxygen, carbon, and hydrogen atoms, respectively.

Line 6 lists the atoms that are connected to one other atom in the query structure (node degree of 1) and to one or more other atoms in the compound structure (node degree of 1 or more). A compound atom could possibly be connected to more atoms than its corresponding query atom, but certainly not to less atoms. Line 7 shows the atoms that have a node degree of 2 for the query structure and a node degree of 2 or more in the compound structure.

Since query node 1 is a carbon atom with degree 2 and is involved in both single and double bonds, its corresponding node in the compound structure is obtained by intersecting the compound sets representing single bond, double bond, carbon atom, and node degree 2 or greater (lines 1, 2, 4, and 7 in Figure 4). The only compound node common to these sets is node 5. The correspondence of query node 1 to compound node 5 is shown on line 8 of Figure 4.

Similarly query node 2 is an oxygen atom of degree 1 and is involved in a double bond. Its corresponding compound node is thus found by intersecting the compound sets representing double bond, oxygen atom, and node degree 1 or greater (lines 2, 3, and 6), leaving only compound node 6. This correspondence is shown on line 9.

The process of intersecting sets to reduce their size is known as partitioning. Lines 10 and 11 show the results of intersecting lines 1, 3, and 7 and lines 1, 5, and 6, respectively. At this stage, all of the query atoms have been uniquely identified, with the exception of the hydrogen atom (query node 4), which has four possible corresponding atoms in the compound structure.

At this point, the connectivity property is applied to the partitioned sets, using the information about atom interconnection. From line 8, the query nodes connected to query node 1 (namely, query nodes 2 and 3) must correspond to those compound nodes connected to compound node 5 (namely, compound nodes 1, 6, and 7). This is shown on line 12. Similarly, lines 13, 14, and 15 show the sets of nodes connected to the nodes shown on lines 9, 10, and 11.

| Line No. | Property | Query Nodes | Compound Nodes |
|---|---|---|---|
| 1 | Single bond | 1,3,4 | 1,2,3,4,5,7,8 |
| 2 | Double bond | 1,2 | 5,6 |
| 3 | Oxygen Atoms | 2,3 | 6,7 |
| 4 | Carbon Atoms | 1 | 1,5 |
| 5 | Hydrogen Atoms | 4 | 2,3,4,8 |
| 6 | Node Degree 1 | 2,4 | 1,2,3,4,5,6,7,8 |
| 7 | Node Degree 2 | 1,3 | 1,5,7 |
| Partition: (lines 1–7) | | | |
| 8 | | 1 | 5 |
| 9 | | 2 | 6 |
| 10 | | 3 | 7 |
| 11 | | 4 | 2,3,4,8 |
| Connectivity: | | | |
| 12 | Line 8 | 2,3 | 1,6,7 |
| 13 | Line 9 | 1 | 5 |
| 14 | Line 10 | 1,4 | 5,8 |
| 15 | Line 11 | 3 | 1,7 |
| Partition: (lines 8–15) | | | |
| 16 | | 1 | 5 |
| 17 | | 2 | 6 |
| 18 | | 3 | 7 |
| 19 | | 4 | 8 |

Figure 4. Sets of query and compound nodes used in matching carboxylic acid group with acetic acid

There are now eight criteria for matching (lines 8 to 15) and, by intersecting the sets on these eight lines, the partitioned sets shown on lines 16 to 19 are obtained. These sets show the one-to-one correspondences between the 4 query atoms and the atoms which constitute the carboxylic acid group in acetic acid, thus proving the presence of the query group in the compound.

## APPLICATION IN A GROUP-COUNTING PROGRAM

To demonstrate the application of the Sussenguth graph-matching technique in a group-counting program for physical property estimation, consider the Lydersen method (*18*) for the estimation of critical properties of organic compounds. The Lydersen method is based on the following equations where $T_c$, $P_c$, and $V_c$ are the critical temperature, pressure, and volume, $T_b$ is the normal boiling point, and $M$ is the molecular weight.

$$\theta = T_b/T_c \tag{1}$$

$$\theta = 0.567 + \Sigma\Delta_T - (\Sigma\Delta_T)^2 \tag{2}$$

$$P_c = M/(\Sigma\Delta_P + 0.34)^2 \tag{3}$$

$$V_c = 40 + \Sigma\Delta_V \tag{4}$$

The $\Delta$ contributions used in the above equations are tabulated elsewhere for the various atomic subgroups which occur in this method—e.g., $-CH_3$ $-CH_2-$, $-CO-$, $=CH_2$, $-NH-$. There is a striking similarity between the Lydersen method and the Rihani-Doraiswamy method (*24*) for estimating the heat capacity of ideal gases ($C_p^0$).

TABLE I

ORDER OF SUB-GROUPS FOR RIHANI-DORAISWAMY/LYDERSEN PROGRAM

| Atom Sub-Class. | Group No. | Group | Atom Sub-Class. | Group No. | Group |
|---|---|---|---|---|---|
| Si | 1 | $-\overset{\vert}{\underset{\vert}{Si}}-$ | C (Adjacent Double) | 36 | $\diagdown C=C=C\overset{H}{\diagup}_{H}$ |
| B | 2 | $-\overset{\vert}{\underset{\vert}{B}}-$ | | | |
| F | 3 | $-F$ | C (Adjacent Double) | 37 | $H\diagdown C=C=C\overset{H}{\diagup}$ |
| Cl | 4 | $-Cl$ | | | |
| Br | 5 | $-Br$ | C (Adjacent Double) | 38 | $=C=$ |
| I | 6 | $-I$ | | | |
| S,O,H | 7 | $-SO_3H$ | C (Double) | 39 | $H\diagdown C=C\overset{H}{\diagup}_{H}$ |
| S,H | 8 | $-SH$ | | | |
| S | 9 | $\leftrightarrow S \leftrightarrow$ | C (Double) | 40 | $\diagdown C=C\overset{H}{\diagup}_{H}$ |
| S | 10 | $-S-$ | C (Double) | 41 | $H\diagdown C=C\overset{H}{\diagup}$ (Cis Double) |
| S | 11 | $=S$ | | | |
| N,O | 12 | $-O-NO_2$ | C (Double) | 42 | $H\diagdown C=C\diagup_{H}$ (Trans Double) |
| N,O | 13 | $-O-N=O$ | | | |
| N,O | 14 | $-NO_2$ | C (Double) | 43 | $\diagdown C=C\overset{H}{\diagup}$ $H$ (Double Unspecified) |
| N,H | 15 | $-NH_2$ | | | |
| N,H | 16 | $-\overset{\vert}{N}-H$ | C (Double) | 44 | $\diagdown C=C\overset{H}{\diagup}$ |
| N | 17 | $-\overset{\vert}{\underset{\vert}{N}}-$ | C (Double) | 45 | $\diagdown C=C\diagup$ |
| N | 18 | $\leftrightarrow N \leftrightarrow$ | | | |
| C,N | 19 | $-C\equiv N$ | C (Double) | 46 | $=CH_2$ |
| C,N | 20 | $-N=C$ | | | |
| C,O | 21 | $-COOH$ | C (Single) | 47 | $-CH_3$ |
| C,O | 22 | $-C(=O)-O-C=O$ | C (Single) | 48 | $-CH_2-$ |
| C,O | 23 | $-C(=O)-O-$ | C (Single) | 49 | $-\overset{\vert}{C}-H$ |
| | | | C (Single) | 50 | $-\overset{\vert}{\underset{\vert}{C}}-$ |
| C,O | 24 | $-CHO$ | | | |
| C,O | 25 | $-\overset{\vert}{C}=O$ | | 51 | 3-membered C or O ring |
| O,H | 26 | $-OH$ | | 52 | 4-membered C or O ring |
| O | 27 | $\leftrightarrow O \leftrightarrow$ | | 53 | 5-membered C-ring, pentane |
| O | 28 | $-O-$ | | 54 | 5-membered C-ring, pentene |
| O | 29 | $=O$ | | | |
| C (Resonant) | 30 | $\leftrightarrow \overset{\vert}{C} \leftrightarrow$ H | | 55 | 6-membered C-ring, hexane |
| C (Resonant) | 31 | $-C\leftrightarrow$ | | 56 | 6-membered C-ring, hexene |
| C (Resonant) | 32 | $\leftrightarrow C \leftrightarrow$ | | | |
| C (Triple) | 33 | $\equiv CH$ | | | |
| C (Triple) | 34 | $\equiv C-$ | | | |
| C (Adjacent Double) | 35 | $H\diagdown C=C=C\overset{H}{\diagup}_{H}$ | | | |

In this latter method, $C_p^\circ$ is expressed as a power series expansion of temperature.

$$C_p^\circ = A + BT + CT^2 + DT^3 \qquad (5)$$

Both methods depend on counting the number of subgroups in the molecule, and the subgroups listed in the table for each method are very similar. Indeed, it is possible to use an identical structure-searching algorithm for both. This algorithm formed the basis of a program which was written to search for the presence of the Rihani-Doraiswamy subgroups, but to generate the structural counts for both the Lydersen and the Rihani-Doraiswamy methods. The methods of García-Bárcena (*11*) for estimating the critical compressibility factor ($Z_c$) and of Franklin (*9, 10*) and Verma-Doraiswamy (*30*) for estimating the standard heat of formation ($\Delta H_f^\circ$) also have many common structural groups with the Rihani-Doraiswamy method, as well as some extra groups. These extra groups were also included in the standard 50 query structural groups used for searching the compound structure, to provide for inclusion of these methods within the system at some future time.

The 50 query subgroups are stored in a particular order on a query tape in the form of the Sussenguth binary matrices shown in Figure 3. They are shown in Table I together with the six extra groups for the over-all structure of ring compounds, as used by the Rihani-Doraiswamy method (group Nos. 51 through 56 in Table I). The structure of the compound, whose properties are to be estimated, is read in from cards and translated into the binary matrices described above before being stored on a "compound tape." When more than one compound is studied, the structure of each compound is stored sequentially on this tape. To count the subgroups in a compound, its structure is read into core storage from the compound tape. The query groups are then read in under control of the main structure-counting program for matching against the compound structure by the Sussenguth graph-matching subprograms. After each compound has been analyzed, the next compound structure is read in from the compound tape until all the compounds have been analyzed.

Before attempting to match a query subgroup, a screening test is performed by the program to decide whether all of the atoms in this query are present in the compound graph. If so, the Sussenguth matching program is called and the query structure is read into core from the query tape. If all the required atoms are not present, the remaining query subgroups in the same atom class are skipped over on the query tape, since they cannot possibly be present, and the next atom class is tested. The atom classes used were, in order:

Si; B; F; Cl; Br; I; S, O, H; S, H; S; N,
O; N, H; N; C, N; C, O; O, H; O; C

The atom class for each of the 50 groups is also shown in Table I. Groups 30 to 50 contain only C and H. They were further subdivided into subclasses based on the bonds present in the query structure:

Resonant; Triple; Adjacent Double (=C=); Double; Single

When the graph-matching programs determine that a query subgroup is not present as a subgraph on the compound graph, the program returns to test whether the next query group could possibly be present. If a subgroup is found to be present, then it is removed from the compound structure so as not to be identified again.

Since the Lydersen method differentiates between ring and non-ring contributions when a group which could possibly be in a ring is identified, a subroutine is called to determine whether any of the atoms which were removed were present in a ring.

After the relevant group count has been incremented, the residual compound structure is tested to determine whether it is empty. If it is not empty, the graph-matching algorithm is used again to match the same query structure (already resident in core storage) as a subgraph of the compound graph.

If the compound structure is found to be empty, all of its constituent subgraphs have been identified, and the count vectors are printed. Before generating the Lydersen count vector, the program tests whether any "illegal" Lydersen groups, which could not be handled by the Lydersen method, are present. If any of these are present, no critical properties can be estimated, and the program continues on to test for "illegal" Rihani-Doraiswamy groups.

The contributions are finally summed and property cards for $P_c$, $V_c$, and $\theta$ are printed and punched for input to the AIChE System. If none of the illegal Rihani-Doraiswamy groups are present, the Rihani-Doraiswamy count vector is then generated and the sums of the contributions to the coefficients $A$, $B$, $C$, and $D$ in the $C_p^0$ power series expansion (Equation 5) are calculated.

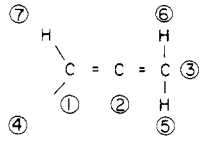
After completing the calculations for the first compound, the program then returns to read in the next compound structure from tape.

## RESULTS AND DISCUSSION

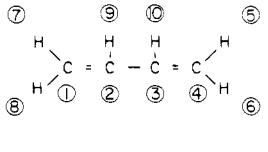
The program described above to obtain Rihani-Doraiswamy and Lydersen group counts was tested on 48 compounds to determine the execution time of the various parts of the program and to test the logic of the program. In almost all cases, the compound was correctly analyzed, and the program gave the same results as were computed by hand. The exceptions are discussed below.

The program does not correctly analyze compounds containing conjugated or adjacent double bonds as in 1,3-butadiene. This is due to the loss of information in translating the input structural data into the bond matrix and connection matrix representation. For example, consider the matching of query group 35 against 1,3-butadiene. Both structures are shown below.

Query Group 35



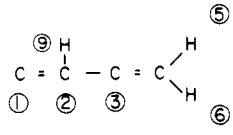
1,3-butadiene



The bond matrix does not uniquely record the number of double bonds in which a given atom is involved, nor the atoms to which it is attached by these double bonds.

Thus, the double bond set does not differentiate between those nodes that are involved in two double bonds (query node 2 and no compound nodes) and those involved in only one double bond. The connection matrix representation cannot detect that compound node number 3 is not connected to compound node number 2 by a double bond, as it should be for query group 35 to be present.

Thus, the algorithm eventually determines that the subgraph of butadiene shown below matches query group number 35. The match is successful even though node 3 is not involved in two double bonds, as required by group number 35, and node 3 is attached to a hydrogen atom (compound node 10) in butadiene, but its corresponding node in group 35 (query node 2) is not attached to a hydrogen atom.




After removing the identified group, the remaining atoms did not match any other query group and the compound could not be analyzed completely.

To analyze correctly conjugated and adjacent double bond structures for the Rihani-Doraiswamy method, it will be necessary to use a separate connection matrix for each bond type, indicating which atoms are joined together by single bonds, by resonant bonds, etc. These separate connection matrices would be used instead of the general connection matrix, when generating the connected sets in the Sussenguth algorithm.

The execution times of the various parts of the program were timed to the nearest sixtieth of a second, using the interval timer attached to the MIT IBM 7094.

The time to search a compound structure for the presence of the 56 subgroups listed in Table I ranged from 64/60 seconds to 620/60 seconds, depending on the complexity of the molecule. The average was around 195/60 seconds. Large groups were identified relatively quickly, because of the many criteria which are applied for matching the query structure. The Sussenguth algorithm is not very efficient when there are a number of identical groups in the molecule, or even when there are a number of similar groups present.

The average total time per compound, including reading and translating the input structural data, searching for the constituent groups, and generating the Lydersen and Rihani-Doraiswamy counts in core storage, was 264/60 seconds. To estimate the time required for handling one compound in a job, a minimum of 297/60 seconds should be added to this total for writing and detecting end-of-file marks, rewinding tapes, and linking between "chain links." This is a considerable amount of overhead, and most of it is involved in chain linking.

The execution times reported above do not compare too favorably with those given by Brasie and Liou (*4*) for a program which decodes linear Wiswesser notations (*31, 32*) to generate the group counts for the Lydersen method (*18*). Their decoding program was written in the language Algol 60 for the Burroughs B-5000 computer, and they report execution times of 3.2 minutes for 825 compounds. This time included compilation of their program, which took an average of 40 to 45 seconds. Thus, searching of a compound structure for the Lydersen group counts, using their program, certainly takes less than 11/60 second per compound on the Burroughs B-5000, which appears to be comparable to the IBM 7094 in memory access time.

However, the principal disadvantage of the Wiswesser code for input of structural data in the computer estimation of physical properties is the complexity of the code and the time required for an engineer to learn the system so as to be able to encode even simple compounds. Despite Wiswesser's claims (*20*), from the point of view of the engineer, the Wiswesser notation system does not follow Zipf's Principle of Least Effort (*33*). In addition, the Wiswesser input format is not very adaptable to a versatile structure-counting program, which must be able to identify rapidly all of the types of query subgroups used in group-contribution methods.

Two recent bachelor's theses at MIT by Ross (*25*) and Dillon (*5*) successfully automated two other group contribution methods. Both programs accept structural information in the input format of Sussenguth illustrated in Figure 3. Ross used the Sussenguth algorithm in estimating critical pressures and temperatures by the methods of Forman and Thodos (*7, 8*). The search times he reports are in the same range as those reported above for the Rihani-Doraiswamy/Lydersen program (*16*). The program developed by Dillon (*5*) for the estimation of the standard heat of formation ($\Delta H_f^\circ$) of ideal gases by the Anderson-Beyer-Watson method (*2, 14*) used a completely different type of logic. The atoms in the compound are inspected one at a time to determine the type of atomic subgroup which they constitute. This information is used to "synthesize" the molecule by building it up from a base group related to the homologous series of the compound. Dillon made extensive use of push-down stacks to keep track of the alternative chains as the algorithm proceeds along the molecule. However, his program is specific for the Anderson-Beyer-Watson method and cannot be adapted to other group contribution methods. He reports execution times in the range 1/30 second to 1/5 second for the total time for each compound, including searching the structure and estimation of the enthalpy of formation.

Further study is needed to determine whether it is possible to adapt the push-down stack concept to speed up the counting of structural groups for a general-purpose structure-counting system so as to improve on the execution times obtained using the Sussenguth programs.

## GENERAL-PURPOSE STRUCTURE-COUNTING SYSTEM

Figure 5 shows a suggested skeleton flow diagram for an over-all structure-counting system designed to run separately from the AIChE System as a precursor.

The structural data for each compound would be read in and translated into Sussenguth-type binary matrices and written on the compound library tape. Physical property data, such as the normal boiling point and the temperature at which $C_p^\circ$ is desired would also be read in and saved on a property tape for later use by the program. Disk or drum storage, instead of tape, would improve the efficiency of the program considerably.

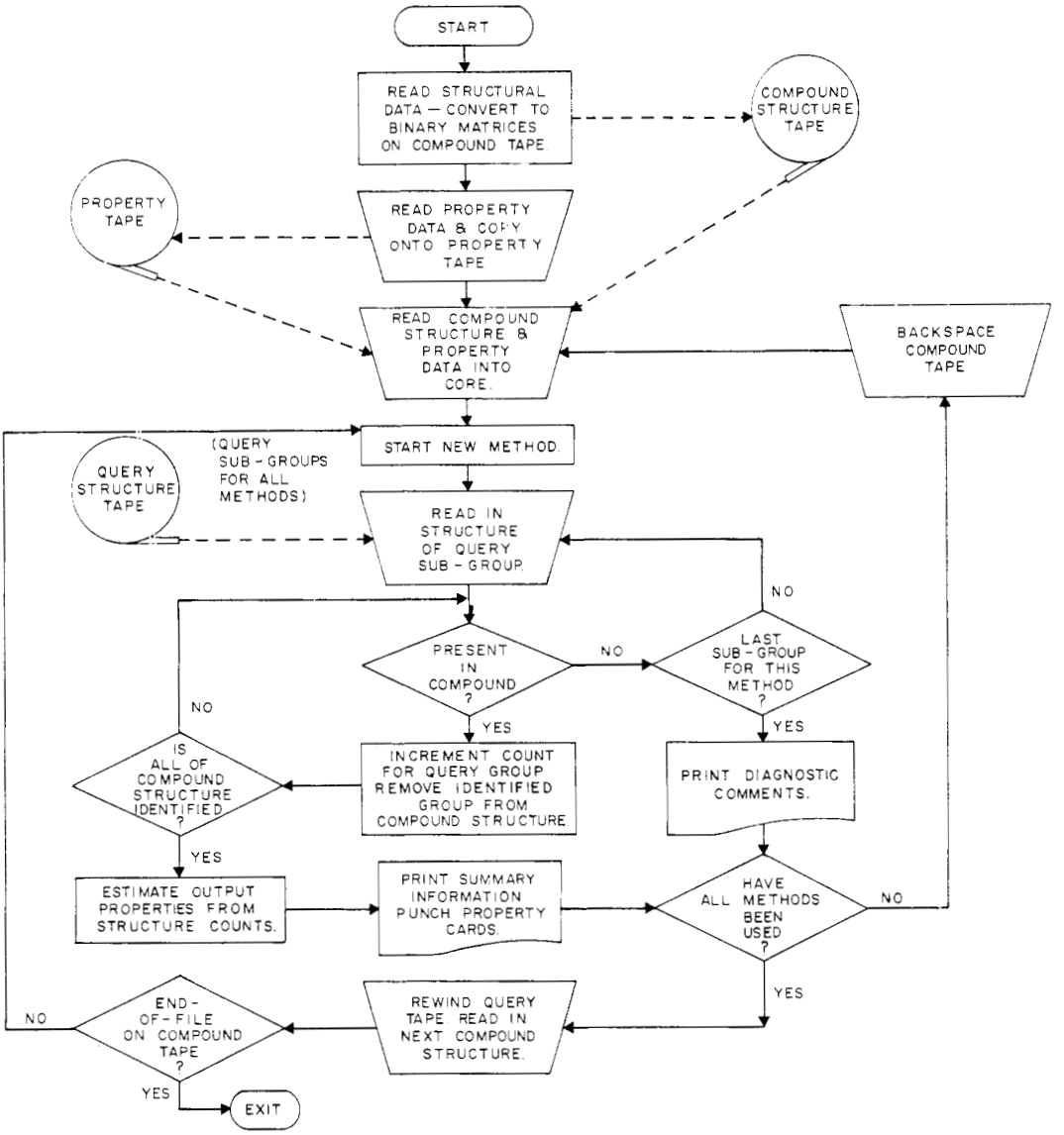



Figure 5. Flow diagram of structure-counting system

After reading all the input structural data, the program would be directed to start the calculation of the properties of the first (or only) compound by reading its structure into core from the compound library tape and the known properties from the property tape. The structures of the query subgroups used by each estimation method would have been previously saved in the form of Sussenguth-type binary matrices on the query tape, an end-of-file separating the query structures for each method.

The search of the compound structure would be directed by a different subprogram for each method, but each of these "driving" programs would use the same utility subroutines for reading in query structures, for matching the query (using the Sussenguth algorithm), and for removing identified atoms. The details of the screening process would vary from method to method. When each compound structure has been identified, the group counts could either be punched as property cards for input to the AIChE System, or the output properties could be estimated from the group counts and then punched as property cards.

To start the next method, the compound tape would be backspaced one record to read in the compound structure again, since it will have been destroyed in core by removing identified groups.

When all the methods have been used on the first compound, the query tape would be rewound and the next compound (if any) read in from the compound tape. An end-of-file on the compound tape terminates the job, since all the compounds would have been processed.

If it is desired to reduce substantially the execution times obtained using the Sussenguth search-and-match type of algorithm, a completely different approach would have to be adopted. Further study should show whether it is possible to use a sequential process of inspecting the atoms of the compound one at a time and counting the groups until all of the atoms have been identified in a similar fashion to the logic used by Dillon (5). Push-

down stacks would be used to keep track of the last branch point at which the program is directed to move along one of the alternative branch chains.

## ACKNOWLEDGMENT

The invaluable help and advice of Edward H. Sussenguth, Jr. of IBM Corp., and of James Murphy and George Harris of Arthur D. Little, Inc., is gratefully acknowledged. All of the computation work reported in this paper was performed on the IBM 7094 in the MIT Computation Center.

## LITERATURE CITED

(1) American Institute of Chemical Engineers Physical Property Estimation System User's Meeting, Arthur D. Little, Inc., Cambridge, Mass., Nov. 14, 1966.
(2) Anderson, J. W., G. H. Beyer, and K. M. Watson, *Natl. Petrol. News*. Tech. Sect. **36**, R 476 (July 5, 1944).
(3) Arthur D. Little, Inc. "AIChE Physical Property Estimation System," Vol. **I** Rept., Vol. **II** User's Manual, Vol. **III** System Manual.
(4) Brasie, W. C., and D. W. Liou, *Chem. Eng. Progr.* **61** (5), 102 (1965).
(5) Dillon, R. S., "Computer Estimation of the Enthalpy of Formation of Chemical Compounds by the Method of Anderson, Beyer and Watson," S. B. thesis in Chemical Engineering, MIT, Cambridge, Mass., 1967.
(6) Dyson, G. M., "A New Notation and Enumeration System for Organic Compounds," Longmans, Green & Co., London and N. Y., 1949.
(7) Forman, J. C., and G. Thodos, *AIChE J.* **4**, 356 (1958).
(8) Forman, J. C., and G. Thodos, *AIChE J.* **6**, 206 (1960).
(9) Franklin, J. L., *Ind. Eng. Chem.* **41**, 1070 (1949).
(10) Franklin, J. L., *J. Chem. Phys.* **21**, 2029 (1963).
(11) García-Bárcena, G. J., "P-V-T Relations at the Critical Point," S. B. thesis in Chemical Engineering, MIT, Cambridge, Mass., 1958.
(12) Gluck, D. J., J. CHEM. DOC. **5**, 43 (1965).
(13) Heitman, R. E., and G. H. Harris, *Ind. Eng. Chem.* **60**, 50 (1968).
(14) Hougen, O. A., K. M. Watson, and R. A. Ragatz, "Chemical Process Principles," Part II, 2nd. ed., pp. 1004–1013, Wiley, New York, 1959.
(15) International Union of Pure and Applied Chemistry, "Rules for IUPAC Notation for Organic Chemistry," Wiley, New York, 1961.
(16) Jochelson, N., "Computer Estimation of Physical Properties of Chemical Compounds by Group Contribution Methods," Sc.D. thesis, MIT, Cambridge, Mass., 1967.
(17) Leiter, D. P., and H. L. Morgan, J. CHEM. DOC. **6**, 226 (1966).
(18) Lydersen, A. L., "Estimation of Critical Properties of Organic Compounds by the Method of Group Contributions," *Univ. of Wisconsin, Eng. Expt. Sta. Rept. No.* **3**, Madison, Wisconsin, April 1955.
(19) Meadows, E. L., *Chem. Eng. Progr.* **61**, (5), 93 (1965).
(20) National Academy of Sciences–National Research Council, "Survey of Chemical Notation Systems," Bull. **1150**, Washington, D. C., 1964.
(21) Norris, R. C., *Chem. Eng. Progr.* **61**, (5), 96 (1965).
(22) Ray, L. C., and R. A. Kirsch, *Science* **126**, 814 (1957).
(23) Reid, R. C., and T. K. Sherwood, "The Properties of Gases and Liquids," 2nd. ed., McGraw-Hill, New York, 1966.
(24) Rihani, D. N., and L. K. Doraiswamy, *Ind. Eng. Chem. Fundamentals* **4**, 17 (1965).
(25) Ross, R., "Computer Estimation of Critical Properties by the Methods of Forman and Thodos," S. B. thesis in Chemical Engineering, MIT, Cambridge, Mass., 1967.
(26) Salton, G., and E. H. Sussenguth, Jr., "A New Efficient Structure Matching Procedure and its Application to Automatic Retrieval Systems," Presented at 26th Annual Meeting, American Documentation Institute, pp. 143–146, Chicago, Ill., October 1963.
(27) Sussenguth, E. H., Jr., J. CHEM. DOC. **5**, 36 (1965).
(28) Sussenguth, E. H., Jr., "Structure Matching in Information Processing," Doctoral Dissertation, Harvard University, Cambridge, Mass., 1964.
(29) Sussenguth, E. H., Jr., personal communication, IBM Corp., San Jose, Calif. (June 1966).
(30) Verma, K. K., and L. K. Doraiswamy, *Ind. Eng. Chem. Fundamentals* **4**, 389 (1965).
(31) Wiswesser, W. J., "A Line-Formula Chemical Notation," T. Y. Crowell Co., New York, 1954.
(32) Wiswesser, W. J., E. G. Smith, and H. T. Bonnett, "A Line-Formula Chemical Notation," mimeographed revision (March 1962).
(33) Zipf, G. K., "Human Nature and the Principle of Least Effort," Addison-Wesley Press, Inc., Cambridge, Mass., 1949.