norbornane. However, the use of rings other than in the SSSR would not be applicable to all cases; as indicated above, the nine-membered cyclononane ring could not be used for this purpose when analyzing hydrindane.
(6) Gasteiger, J.; Jochum, C. *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 43-48.
(7) Roos-Kozel, B. L.; Jorgensen, W. L. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 101-111.
(8) One important point to note is that a molecule may have more than one SSSR. For example, cubane contains six four-membered rings, yet there are only five in the SSSR. In the algorithm described here, only one SSSR is determined; to date this has proved sufficient for our purposes. For some molecules it might be preferable to select one specific SSSR rather than use an arbitrary one. It would then be necessary to either extend the algorithm or use some other method.
(9) Wipke, W. T.; Dyott, T. M. *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 140-147.
(10) The concept of valence is used in COBRA to mean the sum of the bond orders for a given atom type. COBRA also distinguishes between charged and neutral atom types-hence sp^3-hybridized oxygen has a valence of 2; whereas, negatively charged sp^3 oxygen has a valence of 1.
(11) Jochum, C.; Gasteiger, J. *J. Chem. Inf. Comput. Sci.* **1977**, *17*, 113-116.
(12) See, for example: (a) Sussenguth, E. H. *J. Chem. Doc.* **1965**, *5*, 3643. (b) Randić, M. *J. Chem. Inf. Comput. Sci.* **1978**, *18*, 101-107. (c) Wipke, W. T.; Rogers, D. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 255-262. (d) von Scholly, A. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 235-241.
(13) Leach, A. R. D. Phil. thesis, University of Oxford, 1989 (Chapter 8).
(14) For example, the formation of an acyclic C-N bond from a C-C unit provides a structure very close to the minimum energy conformation for this fragment, but the formation of a cyclic double bond from a single bond leads to some distortion. The algorithms can perform more than one adjustment per unit (e.g., in the formation of morpholine from cyclohexane in which two ring carbons are substituted).

AUTONOM: System for Computer Translation of Structural Diagrams into IUPAC-Compatible Names. 1. General Design

J. L. WISNIEWSKI

Beilstein Institute, Varrentrappstrasse 40-42, D-6000 Frankfurt/Main 90, West Germany

Received May 9, 1990

The rules for assigning the systematic name to a structure are complex and frequently lead to ambiguous names. It is this difficulty in assigning names that can be overcome by a program which uniquely translates graphic structures into IUPAC-compatible text names and is readily available as a personal computer tool. The algorithm developed for AUTONOM analyzes the compound's structural diagram, input via a graphic interface, and generates the name purely on the basis of the resulting molecular connection table. This paper describes the design of AUTONOM, presents an analysis of important software and chemical nomenclature solutions adopted during the work on the system, and discusses the system's current accuracy and reliability.

INTRODUCTION

The structural information on a chemical compound can be represented and communicated by a variety of methods. The three most important categories are

chemical nomenclature used to name compounds
formulas and line notations used as shorthand representations of the content and orientation of compounds
structural diagrams used to represent complete graphic information on composition and topology of compounds

The structural diagram conventions are established as an international standard and transcend language barriers among chemists. Chemists are trained to communicate chemical information by using graphical images; however, to nonchemists they are only interesting shapes and strange configurations which convey little understandable information. Various line notations[1] focus on facilitating computer input and structure manipulation. With their human-unfriendly encrypted names and complex systems of rules and conventions (different for each different line notation), which have to be memorized, they create no alternative to either structural diagrams or chemical names. Names which can accurately describe the composition and format of the structure are still vital for a wider audience. In situations where chemical information on a compound needs to be communicated by the spoken or written word, structural diagrams are inappropriate and names are the only alternative. Names are also vital for institutions producing chemical information, such as the Beilstein Institute or Chemical Abstracts Service (CAS), as indexing tools and are important search-key fields for their databases.

Contrary to well-established, standardized, and internationally acknowledged structural diagram conventions, a complete comprehensive grammar for systematic chemical nomenclature does not exist to date. The system of recommendations[2] which have been developed by the Commision on Nomenclature of Organic Chemistry of the IUPAC has not become a universal standard, mainly because of the complexity of the recommended rules, frequent ambiguity in name assignment, and associated continuing use of much quasi-systematic and trivial nomenclature. There is also a reluctance by the chemical industry[3] to perceive the need for fully systematic nomenclature.

For the purpose of obtaining consistency in the selection of preferred names, both CAS and Beilstein devised nondocumented ad hoc subrules which only amplified the difficulty of uniquely naming organic compounds. These subrules were necessary since IUPAC rules frequently allow more than one name for a given chemical. As a result, both institutions revised the IUPAC system and created their own "systematic" IUPAC-compatible rather than IUPAC-sanctioned nomenclatures. In addition, trivial or trade names, being shorter and more concise, have successfully replaced systematic names for

0095-2338/90/1630-0324$02.50/0

a number of chemicals which are of commercial importance or are the subject of public concern, e.g., pharmaceuticals, insecticides, or pollutants.

The main driving force for this divergence was the desire to create a truly unambiguous computer-based system which would translate unique graphic structure representations into unique names and vice versa. As long as such a system does not exist, the practicing chemist will find himself alienated from systematic chemical nomenclature. This was the general conclusion of the latest 1987 conference "Chemical Nomenclature into the Next Millenium—Has It a Role?" that addressed the issue.

Computer translation from and to systematic nomenclature has a long history dating from Garfield's pioneering algorithm for the generation of molecular formulas from names[4] over 27 years ago. Since then several organizations have reported the successful implementation of computer systems that convert chemical names into structural diagrams.[5-9] To date, however, there are no reports of a general program for a conversion in the reverse direction, i.e., from structures to names. The earliest papers on research in this area (Conrow[10] and Van Binnendyk and MacKay[11]) reported naming programs for selected classes of ring systems. The first program for a complete structure, but for a very limited class of compounds restricted only to alloys, copolymers, mixtures, and addition compounds, was reported by Vander Stouw et al. in 1976.[12] Then, in 1981, Mockus et al. presented a paper[13] on CAS plans for complete automatic generation of organic compound names. The paper discussed in detail the design of the algorithm; however, there has been no subsequent report of the implementation of such an algorithm. Recently, Meyer and Gould[14] have confirmed the feasibility of creating a microcomputer-based program that would accept structure input and generate IUPAC-like systematic names.

The AUTONOM (AUTomatic NOMenclature) computer-based system presented in this paper is the latest attempt by the Beilstein Institute Research Division to provide the chemical community with a fully automated "structure-to-name" translator.

The IUPAC rules have been adopted as the basis of rational naming. There are, however, some "respectable" nomenclature practices being used which, although yielding succinct and intelligible names, have not yet been sanctioned or codified by IUPAC. These usages are also incorporated in AUTONOM.

Substitutive nomenclature has generally been preferred except where other styles, i.e., subtractive, radicofunctional, or additive, are widely recognized and/or unavoidable.

AUTONOM as an expert system is not limited to specific classes of compounds and has been designated as a general-purpose system for the unambiguous translation of structural diagrams of organic compounds into IUPAC-compatible names.

GENERAL DESIGN

AUTONOM analyzes structure diagrams of organic compounds, entered via a structure-drawing package, and generates names purely on the basis of the molecular connection tables of the input structures. The name generation part of AUTONOM is isolated from the graphic interface of the system and operates on connection tables designed and formatted for optimal processing, in terms of speed and computer storage requirements. Thus the system is fully independent of the graphic package used for structure input. The support of a specific structure-drawing software package[15] (Molkick, Molmouse, ChemText, ChemSmart, ChemDraw, Wimp, Egg, Spellbinder Scientific, Molstruc, PsiGen, etc.) requires only a readjustment of setup parameters in the graphic interface part of AUTONOM with no need to change the kernel of the system. The output character string that represents the compiled chemical name can be directly displayed on a monitor, stored as a database query term, or rerouted to another application for further processing.

The algorithm designed for AUTONOM operates on abstract objects represented in the program by appropriate data structures. These objects fall into three distinct classes, namely functional groups, chains, and ring systems. The objects are created in the course of the naming cycle, and corresponding data structures are initialized and accordingly filled with real data derived purely from the connection table of the processed structure. The abstract objects and their data structures represent skeletal units, i.e., catenated or cyclically connected atoms forming a discrete, nameable structural entity. This object-oriented approach accurately conforms to the rationales of both chemical nomenclature and effective modern programming techniques.

AUTONOM names a structure, just as a nomenclature specialist would, by identifying candidate parent structural fragments. The appropriate IUPAC nomenclature principles are then successively applied to eliminate less-preferred candidates. These then become substituents on the selected parent fragment. Chemists can often apply nomenclature principles almost unconciously by simply "seeing" the structure. AUTONOM does not have this ability and must therefore, during the processing cycle, systematically collect, step-by-step, numerous items of information about the structure. The system has been designed for implementation on a personal computer with much attention paid to storage and CPU limitations normally occurring in personal computing. The items of information about the structure collected in the process of naming are stored in dynamically allocated computer memory segments and kept, only for as long as they are needed, in a currently executed decision-making task and then immediately discarded. The memory segments are then deallocated and returned for further use.

The complete naming cycle consists of the following logical phases:

1. initialization and ring system perception
2. functional group recognition
3. ring system identification
4. parent structure selection
5. name tree creation
6. name assembly

and these are discussed in greater detail in this paper.

During the first five phases, the algorithm creates and identifies objects as well as establishes mutual relations among them. The text name entities are generated, appropriately ordered, and compiled into the complete name only during the last phase of the naming cycle.

INITIALIZATION AND RING PERCEPTION

Once the complete structure has been entered, using a drawing package, the graphic interface of AUTONOM fills the input buffer with data. The communication between the name generation part and the graphic interface of AUTONOM is maintained only via the common buffer. The internal construction of the buffer is static, and the graphic interface is responsible for handling the differences in coding of the input structure by various drawing packages. The data in the buffer are then used by the naming algorithm for constructing a two-dimensional Boolean atom connection matrix and an atom vector. Each component of the vector corresponds to a single atom of the input structure and contains a data record with complete information on the atom type (atomic number, electronic charge, etc.) and its bonding to other atoms in the structure.

```
SSSR:= ([1,2,3,4,5,6,7];[2,3,11,12,13,14,15];[12,13,16,17,18];
        [21,22,23,24,25,26];[27,28,29,30,31,32];[33,34,35,36,37,38,];
        [39,40,41,42,43,50];[42,44,49];[44,45,46,47,48,49])

Ring Systems:= ([1,2,3,4,5,6,7,11,12,13,14,15,16,17,18];
        [21,22,23,24,25,26];[27,28,29,30,31,32];
        [33,34,35,36,37,38];[39,40,41,42,43,44,45,46,47,48,49,50])
```

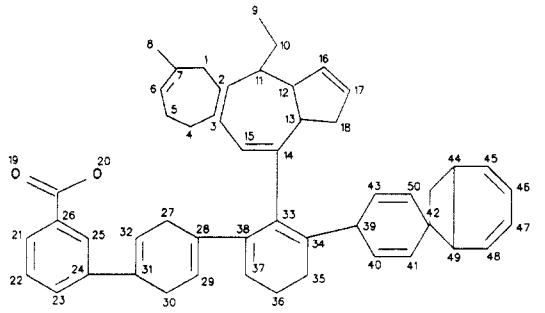


Figure 1. Transition from SSSR into ring systems.

On completion of the initialization step, the first part of the ring system analysis, namely ring perception, is carried out. The other part, i.e., ring system identification, is postponed until the third phase of the naming cycle. The reliability of the ring perception routine is the crux of the naming algorithm.

The ring perception model adopted for AUTONOM can be considered as representative for the "smallest set of smallest rings" (SSSR) category of algorithms. From point of view of nomenclature, it was concluded that the perception of such a set of rings is the most suitable one for successfully naming the cyclic entities occurring in the structure. Once the SSSR for the structure is generated the algorithm finds all the rings that share at least one atom with one or more other rings and builds the list of nameable ring systems. On completion of the process all the atoms involved in cyclic closures are marked and distributed among the ring systems. The transition from perceived SSSR into ring systems for a given structure is illustrated in Figure 1.

Once all the ring systems have been localized and their atoms marked, the algorithm, using the Boolean connection matrix and atom vector components of the marked atoms, generates for each ring system a corresponding so called hash code. The hash code, which is nothing more than a short string of numbers, represents, in a very compact form, information on atom characteristics and atom interconnections of the ring system. The simple nonreversible hashing scheme used for the coding is unique in the sense that the same ring system is always tagged with the same hash string, but it does not mean that each system has a unique code (such a noncollision hashing algorithm does not exist so far). The main purpose of the hashing, as will be explained in greater detail later in this paper, was to relate all the rings in the input structure with classes or pots of rings having the same hash codes.

FUNCTIONAL GROUP RECOGNITION

A dictionary look-up approach with an atom-by-atom connectivity search mechanism was chosen for functional group recognition. Acyclic portions of the input structure are browsed in order to localize skeletal units bearing functional group characteristics such as hetero atom arrangements with unsaturated bonds. Each localized candidate functional group skeletal unit is compared, in an atom-by-atom manner, with entries of the predefined ordered dictionary of functional groups. On completion, the units which were identical with entries in the dictionary are sorted in priority order, guided by nomenclature considerations, and a ranked list of all group present in the structure is composed. The highest ranking groups are then used in the parent structure selection phase to generate a set of candidate parent fragments.

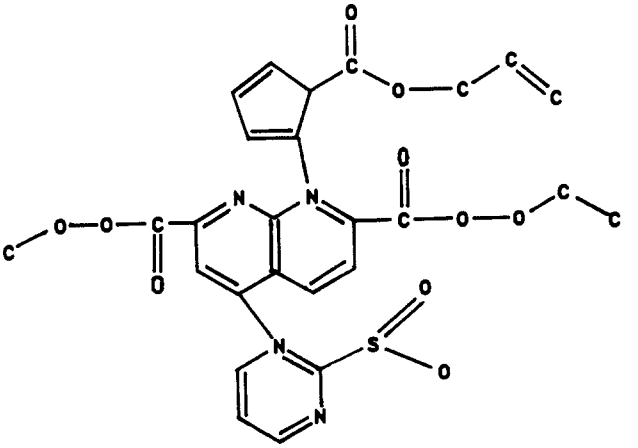


Figure 2. Rank and multiplicity order during functional group recognition.

The term highest ranking is derived from the IUPAC recommendations[2] and is supposed to set clear conditions on the selection of the so-called principal functional group, which later (in the completed name), is cited as the suffix or is the most decisive criterion in forming the name. Unfortunately, the formulation of the recommendation (p 86 in ref 2) in its present form is ambiguous, and both CAS and Beilstein have broadened this recommendation with their own guidelines, particularly in the area of acids and their derivatives.

AUTONOM implements so-called "rank and multiplicity" order during selection. It can be briefly illustrated by the following scenario (Figure 2): two ester derivatives of a carboperoxoic acid, an ester of a carboxylic acid, and a free sulfinic acid group have been recognized during the search cycle. The IUPAC recommendations rank free acids highest in the list. This would then mean choosing the sulfinic acid as the principal group and the term "sulfinic acid" as a suffix in the name. Moreover, the pyrmidine ring system would automatically become the parent structure, while the other rings would become substituents on pyrimidine. It would also mean transforming both ester derivatives of the carboperoxoic acid and the ester of the carboxylic acid into prefixes (ester part name fragments + hydroperoxycarbonyl and ester part name + carboxy). AUTONOM follows the IUPAC recommendations as far as the rank order of acids is concerned, but additionally, mainly in order to simplify and shorten the names, crisscrosses the rank principle with the "maximum multiplicity" principle. This means, for the above example, choosing the carboperoxoic acid group as the principle group, multiplying it into the "dicarboperoxoic acid" suffix attached to a [1,8]-naphthyridine parent structure, and leaving the prefix form of the esterified carboxylic acid unchanged, and also changing suffix form sulfinic acid into the prefix sulfino.

The task of functional group recognition, although here described as the sovereign phase of the naming cycle, is not usually accomplished in a single run. A list of functional groups is delivered for further processing. The entries on the list as well as the data characterizing the entries normally do not stay in the initial form throughout the whole naming cycle. The initial form is, however, extremely important for selecting the principal group(s) which must be fully identified before the algorithm enters the parent selection phase. Later, after the parent fragment of the structure has been selected, the list is usually corrected by running it, at least once more, through dedicated functional group updating routines.

One of the update operations, most frequently encountered in the nomenclature of acids and their derivatives, is the so-called group splitting illustrated in Figure 3. The initial list of functional groups, before entering parent selection phase, has two entries, namely an esterified (by methyl) carbothioic

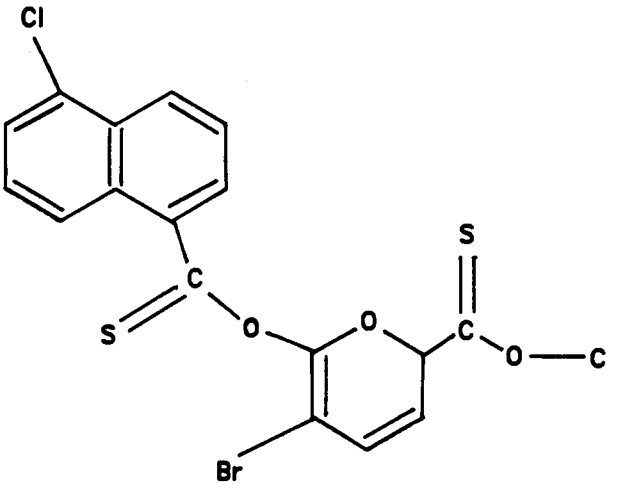


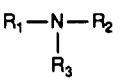
Figure 3. Group splitting during functional group recognition.

acid attached to pyran and an esterified (by pyran) carbothioic acid attached to naphthalene. The both groups are, and must be, initially localized as "based" at R_0 and not at R_1

$$\text{expected parent unit} \leftarrow R_0\text{—}C(=S)\text{—}O\text{—}R_1 \leftarrow$$

This is the initial preferred assignment of "direction" of an atom arrangement in the groups with reference to the expected parent structural unit. Both groups compete for selection as the principal functional group and as the suffix in the final name. The decision as to which group is chosen as the proper suffix (the rank + multiplicity principle cannot help in this case as both groups are identically ranked and multiplied) must be postponed until the parent structure selection phase. According to IUPAC recommendation C.14.11a (p 101 in ref 22) formulated as "all heterocycles are senior to all carbocycles", the pyran ring is selected as the parent structure and accordingly the carbothioic acid attached to it must be selected as the suffix group. In the contrast to the example from Figure 2, the simple transition from the suffix form of the other carbothioic acid into its prefix form (thiocarboxy) does not apply in this case. The group must be split into two other groups, namely (-O) with the prefix oxy descriptor and (=S) with the prefix thioxo descriptor. The splitting takes place sometime later, during the name-tree creation phase, after the parent fragment has been localized and the position of the group, in relation to the parent, has been established.

The preferred initial "direction" of acid groups from Figure 3 (and also from Figure 2) is determined by the established natural topology of atoms in acids. Moreover, since the skeleton C(=S)—O is not symmetrical, it is relatively easy to design a routine that finds the "base" atom of such groups. This is unfortunately not always the case. For highly symmetrical amines, for example



the assignment of direction and the finding of the base atom is impossible during the functional group recognition phase of the naming cycle. In order to avoid a circularity problem (a structural unit is the parent structure if it contains the base atom, and the atom is determined as the base atom if it belongs to the parent structure), AUTONOM ranks all the $R_{1,2,3}$ atoms as potential base atoms. As a consequence of such an approach, in many practical cases, the parent structure selection phase delivers not one, but several (maximum three, in case of amines) candidate parent structures and the decision as to which of them is the proper parent must be postponed until the creation of the complete name tree. This means unavoidable redundancy in processing time and storage demands,

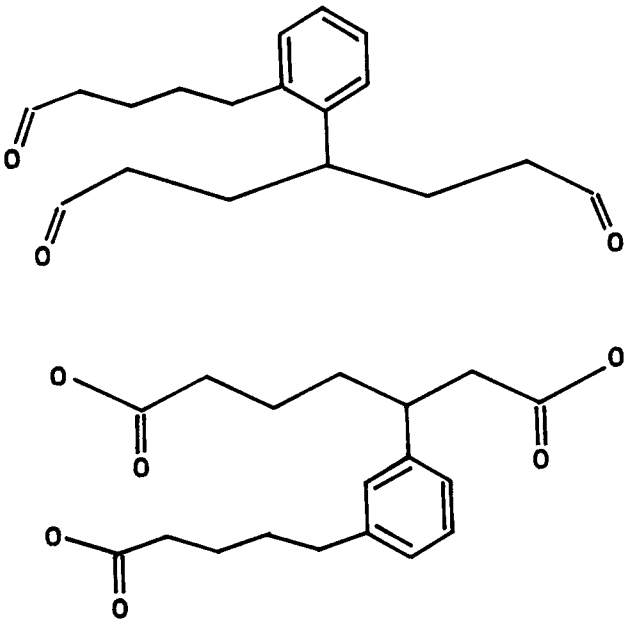


Figure 4. Concept of the "moving base" atom.

but eliminates the problem of circularity without the need to implement recursive programming techniques.

Another example of complex problems, already considered in the course of the functional group recognition phase, is the nomenclature of hydrocarbon chains whose terminal CH_3 units are replaced by functional groups which contain carbon as the central atom. This is the case for aliphatic acids, acyl halides, aliphatic nitriles, and (thio)(seleno)(telluro)aldehydes. In all such cases the central carbon atom of the corresponding group, depending on the status of the chain (parent chain or substitute chain) to which it is attached, may stay as a part of the group or must be included into the set of atoms constituting the chain. This example is described here in order to illustrate that the method adopted for AUTONOM of dividing the structure skeletal units into three separable classes of objects (functional groups, chains, and rings) does not always conform to reality and must be occasionally corrected. In this particular case, AUTONOM identifies the groups with a "moving base" atom rather than with a "fixed base" atom. The concept is illustrated in Figure 4.

For the first structure all three carboxyl groups R_0-COOH are initially recognized as being based at R_0 atoms. Later after the pentane chain has been chosen as the parent structure and its two terminal carboxyl groups identified as suffix principal groups (dioic acid), the central carbons of the groups are returned to the parent chain—it will be later renamed as heptane—and become new R_0 base atoms. On the other hand, the central atom of the carboxyl group attached to the substituent butyl chain is supposed to stay as a part of the group and no further manipulations with the base atoms are necessary. When carboxyl groups are replaced by R_0—C=O carbaldehyde groups, as it happens in case of the second structure from Figure 4, the situation becomes even more complicated.

The treatment of the parent, a 7-C (derived from a 5-C) chain, is identical except for a different suffix (dial). The approach to the carbaldehyde attached to the substituent butyl chain is different however. Its central carbon atom is included in the carbon chain—which will later be renamed as pentyl—and becomes a new base atom. In effect, the carbaldehyde skeleton R_0—C=O is transformed into an R_0=O ketone skeleton, which can no longer be named using a formyl prefix but requires an oxo prefix.

No matter how trivial and obvious the above discussion may seem to a nomenclature expert, the computer implementation

of all the tasks presented is by no means trivial.

RING SYSTEM RECOGNITION

During the preliminary studies on the feasibility of creating a computer system for automatic name generation, it was concluded that the most complex and difficult problems would lie in the area of ring-system naming and ring-atom numbering (assigning ring locants). The programming cycle of the AUTONOM project fully confirmed these expectations.

The preliminary ring perception phase discussed earlier in this paper delivers a complete list of all localized ring systems. Each ring system in the list is described by the set of atoms constituting the system, the array of pointers pointing to positions of its subrings in the previously generated SSSR, and by the hash code. Each entry from such a composed list is then processed by the ring recognition routines.

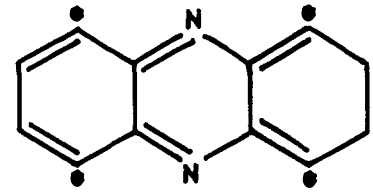
In the first step, obligatory for each perceived ring system, a look-up dictionary access to trivial name ring systems via an atom-by-atom connectivity search mechanism is implemented. The atom-by-atom matching is conducted in a previously prepared dictionary containing trivial name ring connection tables together with prescribed numbering and names. Much attention has been paid to optimizing the organization and construction of the dictionary in terms of efficient processing demands.

Each ring system chosen for inclusion in the dictionary is coded with exactly the same hashing algorithm used for localizing the ring systems in the input structure during the ring perception phase. Thus, the complete set of trivial name ring systems is grouped into small pots with a common hash code as an address to the pots. At the time of writing, the measured average rings/pot ratio was approximately 7. Although the statistical distribution of rings among pots is far from being uniform, the maximum registered number of rings in a single pot is less than 25. This means, in practice, that in the worst case each localized ring system must be matched, atom-by-atom, against a maximum of 25 dictionary entries. To speed up searching even more, only inter-ring atom connections—with no information on bonding—are stored in addition to atom characteristics. In this way no bond restoration is necessary in this phase of the naming cycle, thus saving the time normally spent on bond denormalization and unscrambling during searching.

It is obvious that before the AUTONOM project is completed the current dictionary storage space of ca. 1.5 Mb will increase as will the access time to its entries. Storage requirements will be significantly reduced in future by extracting textual ring descriptors from the dictionary entries, storing them in a separate file, and then applying effective text compression techniques[16] to them.

The search in the dictionary of rings differs only slightly from well-known substructure searching methods.[17] Instead of looking for a single substructure in the set of prescreened structures, a set of perceived cyclic substructures is sought in a single structure. The whole process might, to coin a phrase, be described as an "infrastructure search". The query patterns (ring systems from the list) are compared with all members from the dictionary ring–pot until an exact match is obtained. Having matched the ring system with a dictionary entry, the algorithm continued with the complex task of ring-atom numbering.

For nonsymmetrical systems the situation is relatively simple. Each atom of a ring requires only one fixed locant which means that for each dictionary entry only one collection of fixed locants needs to be stored with the entry. For symmetrical systems this is unfortunately not the case. Depending on the number of symmetry axes, a single atom can be numbered with many locants. It would mean, in practice, that not one but all possible combinations of fixed locants should be stored with every symmetrical ring entry. In the following example of phenazine, each of the marked atoms can, in



principle, be given the number "one". Which of the marked atoms is finally assigned "one" is decided in accordance with the IUPAC recommendation C.15.1 (p 105 in ref 2), and normally only this atom is enumerated as "one", thus guaranteeing that the rest of the atoms will get the lowest possible locants. When deciding on enumeration the following criteria are applied, successively, in the order given:

a. principal groups
b. "indicated" hydrogens
c. multiple bonds in compounds whose names indicate partial hydrogenation (cycloalkenes, pyrazolines, and the like)
d. number of substituents
e. lowest locants for substituents named as prefixes
f. lowest locants for substituents named as prefixes in alphabetical order of citation

However, in order to apply these criteria, the applicable combinations of fixed locants (only 4 in the case of phenazine, rising to 24 for the extreme case of the highly symmetrical cubane) must be known beforehand. The idea of storing all these combinations as arrays of locants, even in the most compressed form, in the dictionary was rejected as nonfeasible from the point of view of both storage requirements and the expected dramatic degradation in system performance. Instead, a relatively simply routine based on look-ahead techniques has been designed and included in the atom-by-atom search procedure. The routine is capable of generating all permutations of locant compositions strictly on the basis of the one composition stored in the dictionary entry and the two-dimensional connection matrix of the ring system that was matched with the entry.

For a wide spectrum of ring system classes dictionary access is obsolete and has been replaced with purely algorithmic ring identification. The dictionary does not contain any entry that falls into one of these classes, instead AUTONOM generates the names of such ring systems exclusively on the basis of the analysis of their internal composition. The policy adopted during work on the AUTONOM project was to cover the greatest possible number of ring system classes with the algorithmic identification and refer to dictionary access only if the latter proved to be distinctly more efficient from a system performance point of view. The classes of ring systems which are recognized purely by an algorithm include:

monocyclic hydrocarbons
bicyclic alkanes
monospirocyclic alkanes
dispirocyclic alkanes
tricyclic alkanes
heteromonocyclics named by the Hantzsch-Widman method[18]
replacement ("a" terms) nomenclature heteromonocyclics
replacement ("a" terms) nomenclature heterobicyclics
replacement ("a" terms) nomenclature monospiroheterocyclics
replacement ("a" terms) nomenclature dispiroheterocyclics

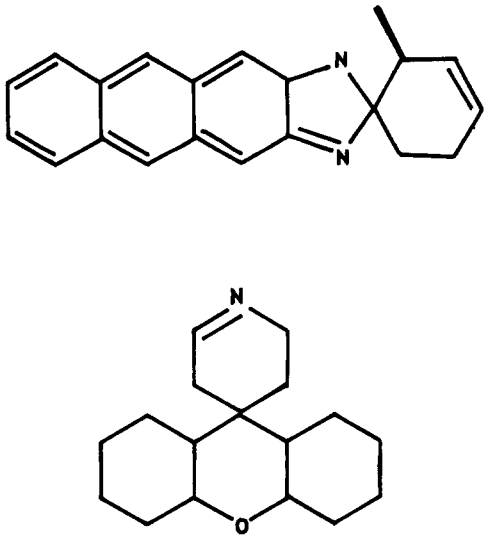


Figure 5. Combined dictionary and algorithmic ring recognition.

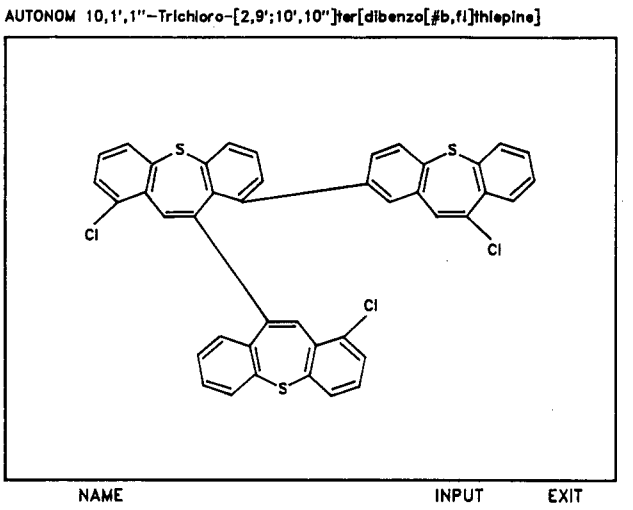


Figure 6. Example of ring assembly (original name and structure output screen from AUTONOM).

replacement ("a" terms) nomenclature heterotricyclics

There is a relatively large group of ring systems, for which a combined dictionary and algorithmic scheme of identification is desirable. This combined approach is applied for all fused polycyclic hydrocarbons and heterocarbons with monospiro or dispiro connections to the ring systems classified in the above list. Two such systems are illustrated in Figure 5.

Implementation of this mixed identification scheme required an amendment to the routines of both the ring perception phase and the ring recognition phase. During the ring perception phase both ring systems are classified as being composed of two conceptually different parts: a fused polycyclic part which should be identified by a dictionary access (anthra[2,3-*d*]-imidazole for the first structure and xanthene for the second structure), and an algorithmically recognizable monocyclic part (cycloalkene for the first structure and Hantzch-Widman system for the other). The atoms of both parts are separated from one another. The spiro atom gets a special "shared atom" priority status, and two separate entries are generated and inserted in the ring system list. Later, after both parts are fully recognized by using the appropriate methods, the results are combined into a common ring descriptor. In the case of the structures from Figure 5 these are 1,11a-dihydro-spiro[anthra[2,3-*d*]imidazole-2,1′-cyclohex-3′-ene] and tetradecahydro-spiro[pyridine-4,9′-xanthene], respectively.

Although the above description sounds simple, the routines to realize this type of processing are very sophisticated and had to be specially designed. It must always be remembered that both parts, although originally isolated from each other for the purpose of simplifying processing, constitute one ring system. Any change applied to one of the parts may result in quite a different name for the whole system. In the case of the second structure (Figure 5), for example, the loss of the double bond in the "pyridine" part leads to a different name, namely dodecahydro-spiro[piperidine-1,9′-xanthene]. The algorithm must "know" that fully saturated "pyridine" becomes "piperidine" and that the term piperidine implies a full saturation of one of the parts of the whole ring system. The unsaturation degree of the other part must be in this circumstance recalculated in order to generate a correct hydro prefix (tetradecahydro → dodecahydro) for the whole ring system.

Preliminary recognition of ring assemblies is the final step in the ring system recognition phase. In IUPAC nomenclature, a ring assembly is a linear composition of two or more identical ring systems joined by acyclic single or double bonds, not necessarily at equivalent positions (Figure 6).

The part of ring assembly recognition conducted in this phase consists of finding all ring systems involved in assembly connections and identifying these systems not as isolated rings but as parts of an assembly. Usually, it means the inclusion of specific nomenclature criteria such as special priority status for the point of attachment atoms in the process of numbering or assigning nonprimed, primed, or multiprimed locants to atoms constituting particular ring systems of the assembly.

PARENT STRUCTURE SELECTION

Once the functional groups and ring systems have been recognized, the algorithm proceeds with the identification of parent chain(s). These are either chain(s) bearing functional groups or chain(s) (not bearing functional groups) which would be acceptable as parents only when a function expressible as a suffix is not present in the structure. The basic design approach to the chain identification (identical for parent chain here and substituent chain later) is built around a single recursive routine, which uses a well-known graph transition technique.[19]

The selected potential candidate structural fragments [ring(s), chain(s), or functional parent(s) if localized] are then ranked according to the relevant nomenclature principles. Seniority of ring systems is decided by applying 12 criteria which are, in general, in agreement with the corresponding IUPAC recommendation C.0.14 (p 101 in ref 2). The principal chain, i.e., the chain upon which the nomenclature and numbering of the whole structure will be based, is chosen by applying successively 10 criteria formulated by the IUPAC recommendation C.0.13. (p 97 in ref 2).

If more than one candidate structural unit of the same rank competes for selection as a parent, the following sequence of principles, in the order given, is applied until a decision is reached:

a. greatest number of the principal functional groups cited as a suffix
b. preferred hetero atom content
c. rings preferred to chains
d. seniority of rings (if ring candidates left)
e. seniority of chains (if chain candidates left)
d. the greatest number of substituents
e. lowest locants for the substituents
f. lowest locants for the substituent cited first as prefixes in alphabetical order

The other structural fragments, which cannot be assigned the status of parent, automatically become substituents on the selected parent structural unit.

It is worth noting here that except for the atoms belonging to substituent chains (and only those chains not being con-

sidered as potential parent candidates) all other atoms of the input structure, on leaving this phase of naming cycle, are hierarchically distributed among nameable structural units.

NAME TREE CREATION

Preliminary investigations prior to the development of the AUTONOM computer system had concluded that the hierarchic principle underlying the approach to a chemical name construction (parent, substituent, substituent-on-substituent, etc.) should be followed as faithfully as possible while designing the appropriate data format for name generation analysis. It was decided to implement the format based on an ordered binary tree concept[19] as fulfilling the majority of both nomenclature and system-performance requirements. The data structure corresponding to this format will be hereafter referred to as the name tree.

Each node of the name tree corresponds to one nameable unit of the input structure and contains a record of data characterizing the unit. The parent fragment selected in the preceding phase is established as the root of the name tree. Starting at the root and traversing in an upward direction, mutual relationships among the nodes (e.g., type of bonding, locants of connections, indicated hydrogen locants for ring systems, locants of multiple bonds in chains, etc.) are sought, and data concerning these relationships are added to the existing node records. Concurrently, if necessary, existing nodes are eliminated from the tree or new nodes are formed and added to the tree.

The formation of new nodes is encountered frequently in the case of newly generated functional groups created during the process of group splitting. One of the groups which arise from the splitting replaces the original node, while the other forms a brand new node of the tree. New nodes must be also created for all not-yet-identified substituent chains. These could not be localized and identified before their starting atoms were known. Now, when the starting atom is determined (atom of attachment to the currently visited node), the same routine that was used during principal chain identification is called here. In case of several "best" substituent chains starting at the same atom, the standard seqeunce of IUPAC principles is applied to choose the senior chain.

The need to eliminate nodes from the tree is encountered in the case of rings involved in assembly junctions. Once a node that corresponds to a ring system has been visited and the involvement of the ring in a ring assembly confirmed (by checking the state of special binary flag stored with the ring data record), the normal tree traversal is interrupted. A sophisticated routine finds all other rings involved in the same assembly junction, then concatenates atoms from all such rings together and generates a single common assembly data record. When the tree traversal is resumed, the nodes of rings involved in an assembly junction are discarded and replaced with a single assembly node.

The mapping of the input structure onto the name tree is complete once the tree has been fully traversed. The input structure is now completely divided into substructural nameable units. The units have been hierarchically ordered and related to the computer data records containing the full information on the units and their mutual relationships.

For optimization reasons the upward traversal routine has been designed as a single-run process with no recursive calls. The ordered binary tree upon which the routine operates has been implemented as a double-linked list of pointers to dynamically allocated and deallocated variant records.

NAME ASSEMBLY

The term "assembly" is reserved, in the theory of computer programming, for describing the task executed by a special

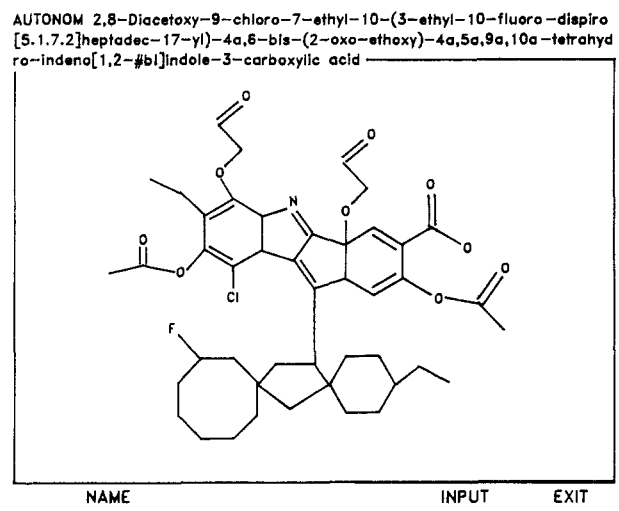


Figure 7. Incorporation of sanctioned trivial nomenclature by AUTONOM.

program that translates the source language of mnemonics and symbols into an object language of binary instructions which the computer hardware "understands". This translation is associated with two scans, or passes, over the source code (so called two-pass assemblers). On the first pass a symbol table (usually implemented as a stack) is constructed which is used in the second pass for substituting the symbols encountered in the source language of mnemonics and to complete the translation. The example of a two-pass assembler is not cited here by accident. The design of routines for the assembly phase of the naming cycle have been very strongly influenced by the solutions adopted in modern assemblers.

Starting at the highest node and traversing the name tree downward, the data records of visited nodes are processed and the resulting textual name fragments are stored in a so-called name fragment table. In order to keep track of the path and the sequential order of the nodes visited while traveling from a given node (substituent) to the root of the tree (parent structure), each node is given a unique label. The label has a form of variable length character string whose length measures the distance of the node from the root (all nodes from the same tree level have labels of the same length) and whose particular characters represent the sequence of nodes that should be visited in order to get to the root of the tree. Each label thus generated is immediately stored on a dedicated, reversed, "first-in–first-out" stack. On completion of the name tree traversal, the name fragments are readily available in the name fragment table while the indexes to their entries in the order that should be followed are stored on the stack. The first pass of the name assembly has been completed.

During the second pass, starting at the top element of the label stack, the corresponding name fragment entries, pointed at by the labels, are successively taken from the name fragment table and combined, by applying the proper chemical nomenclature semantics and syntax, into longer fragments. The whole process repeats itself until the stack is empty and the first version of a complete name is obtained. The task of combining name fragments into longer units and finally into a complete name is realized by several complex routines that handle such sophisticated operations as alphabetization, multiplication, punctuation, vowel deletion, suppression of unnecessary locants, superscript and italic string placement, etc. It should also be mentioned here that the combining process is, from the very beginning, uninterruptedly monitored by intelligent so-called "triviality" controller routines. The controller is responsible for tracking and replacing, when necessary, two systematic and thus not intelligible nomenclature with well-established IUPAC-sanctioned, nonsyste-

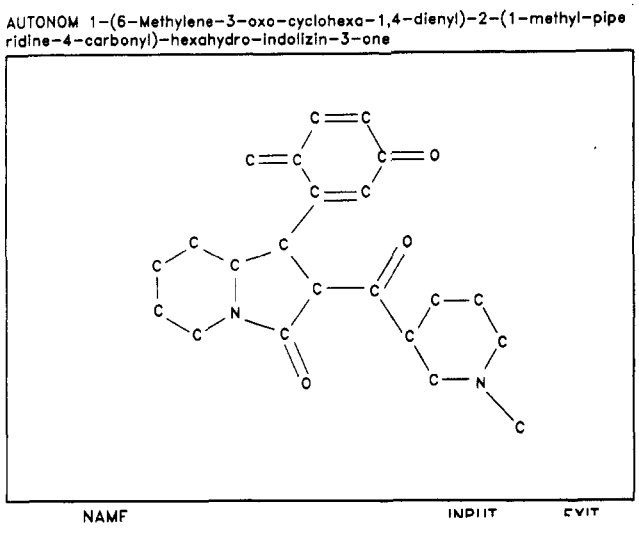


Figure 8. Name and structure output screens from AUTONOM.

matic, traditional nomenclature. The analysis of one relatively simple example (Figure 7) proves that incorporating trivial nomenclature into a consistently systematic computer algorithm is by no means trivial.

In case of the structure from Figure 7 (original name and structure output screen from AUTONOM), the transient prefix name fragment (1-oxo-ethoxy) has been replaced with acetoxy, which is preferred by both CAS and Beilstein. On the other hand, the prefix (2-oxo-ethoxy), and the "triviality" controller should "know" this, does not need any further changes. The simple string replacement in the first case has important consequences as far as the syntax of the global name is concerned. The name fragment (1-oxo-ethoxy) would be multiplied with bis-, tris-, tetrakis, etc., multiplication affixes, while the resulting acetoxy must be multiplied with di-, tri-, tetra-, etc., affixes. Deciding on this multiplication scheme means that during alphabetical ordering of prefixes diacetoxy, according to IUPAC recommendation C.16.31. (p 109 in ref 2) as starting with "a" (the multiplying affix di does not count), is cited as first. On the other hand, the (2-oxo-ethoxy) prefix should be multiplied with a bis affix, stays enclosed in parentheses, and starts with the initial "o" of the oxo subprefix. This is why, in the complete name, it is the last prefix.

The text string replacement described above should not be confused with another string exchange which can (if required) be implemented as a much simpler single-run process in a user-defined shell. This may be done at the very last stage of name assembly, where language and usage considerations should be taken into account [propanoic → propionic, indole → indol (German), etc.].

SUMMARY

Tests conducted on random samples from the Beilstein database indicate that the program in the current phase of the AUTONOM project achieves expert status in ca. 61% tested so far, where expert status is defined as output identical with that of the Nomenclature Department of the Beilstein Institute. At the moment AUTONOM cannot handle stereochemistry which accounts for the bulk of the "missing" 39% of truly expert status.

Figure 8 illustrates the use of the program in a PC-based implementation. The given structures were drawn with a mouse, and the AUTONOM names were returned (as shown) in 8, 17, and 6 s, respectively.

Programming AUTONOM to its current state was a substantial task resulting in 19 500 Pascal program lines and 97 routines and functions. It runs on an IBM-AT (or compatible, also 80386-based) with a minimum of 512 kB of RAM and a hard disk with at least 1.5 MB free for the storage of the dictionary of trivial name ring systems. On average, depending on the complexity of the structures, AUTONOM names up to 17 structures per minute. Input structures are limited to 255 non-hydrogen atoms.

ACKNOWLEDGMENT

The development of AUTONOM is generously supported by the Bundesministerium für Forschung und Technologie (BMFT). The author and the Beilstein Institute gratefully acknowledge this support. Contributions from Dr. L. Goebels, who is responsible for the nomenclature aspect of the AUTONOM project and Prof. A. Lawson, who leads the project, are also gratefully acknowledged.

REFERENCES AND NOTES

(1) Davis, C. H.; Rush, J. E. *Information Retrieval and Documentation in Chemistry*; Greenwood Press: London, 1988; p 143.
(2) International Union of Pure and Applied Chemistry. *Nomenclature of Organic Chemistry*; Pergamon: Oxford, 1979; Sections A-F and H.
(3) Silk, J. A. Realistic vs. Systematic Nomenclature. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 146–148.
(4) Garfield, E. An Algorithm for Translating Chemical Names to Molecular Formulas. In *The Awards of Science and Other Essays*; Garfield, E., Ed.; ISI Press: Philadelphia, 1985; p 453.
(5) Vander Stouw, G. G.; Elliot, P. M.; Isenberg, A. C. Automated Conversion of Chemical Substance Names to Atom Bond Connection Tables. *J. Chem. Doc.* **1974**, *14*, 187–193.
(6) Jochum, C. 192nd National Meeting of the American Chemical Society, Anaheim, CA, 1986; CINF Abstract 28.
(7) Cooke-Fox, D. I.; Kirby, G. H.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Nomenclature. 1. Introduction and Background to a Grammar-Based Approach. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 101–106.
(8) Cooke-Fox, D. I.; Kirby, G. H.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Nomenclature. 2. Development of a

Formal Grammar. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 106-112.
(9) Cooke-Fox, D. I.; Kirby, G. H.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Nomenclature. 3. Syntax Analysis and Semantic Processing. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 112-118.
(10) Conrow, K. Computer Generation of Baeyer System Names of Saturated, Bridged, Bicyclic, Tricyclic, and Tetracyclic Hydrocarbons. *J. Chem. Doc.* **1966**, *6*, 206-212.
(11) Van Binnendyk, D.; MacKay, A. C. Computer-Assisted Generation of IUPAC Names of Polycyclic Bridged Ring Systems. *Can. J. Chem.* **1973**, *51*, 718-723.
(12) Vander Stouw, G. G.; Gustafson, C.; Rule, J. D.; Watson, C. E. The Chemical Abstracts Service Chemical Registry System. IV. Use of the Registry System to Support the Preparation of Index Nomenclature. *J. Chem. Inf. Comput. Sci.* **1976**, *16*, 213-218.
(13) Mockus, J.; Isenberg, A. C.; Vander Stouw, G. G. Algorithmic Generation of Chemical Abstracts Index Names. 1. General Design. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 183-195.
(14) Meyer, D. E.; Gould, S. R. Microcomputer Generation of Chemical Nomenclature from Graphic Structure Input. *Am. Lab.* **1988**, *20* (11), 92-96.
(15) Meyer, D. E.; Warr, W. A.; Love, R. A. Chemical Structure Software for Personal Computers; ACS Professional Reference Book; American Chemical Society: Washington, DC, 1988.
(16) Wisniewski, J. L. Effective Text Compression with Simultaneous Digram and Trigram Encoding. *J. Inf. Sci.* **1987**, *13*, 159-164.
(17) Willet, P. A Review of Chemical Structure Retrieval Systems. *J. Chemom.* **1987**, *1*, 139-155.
(18) International Union of Pure and Applied Chemistry. Revision of the Extended Hantzsch-Widman System of Nomenclature for Heteromonocycles. *Pure Appl. Chem.* **1983**, *55* (2), 409-416.
(19) Tenenbaum, A. M.; Augenstein, M. J. *Data Structures Using Pascal.* Prentice-Hall: Englewood Cliffs, NJ, 1981; pp 252 and 318.

Topological Statistics on a Large Structural File

MICHEL PETITJEAN and JACQUES-ÉMILE DUBOIS*

Institut de Topologie et de Dynamique des Systèmes (ITODYS), associé au CNRS, Université de Paris VII, 1 rue Guy de la Brosse, 75005 Paris, France

Received March 7, 1990

Statistics based upon connection tables have been determined for a large structural file. Many distributions have unexpected local maxima and minima. Parity phenomena are observed in the distribution of hydrogen and carbon. An interpretation of even and odd distributions is proposed. Some of the compounds which represent topological extremes are shown.

INTRODUCTION

Many statistics in chemistry can be developed from chemical compounds that are, with their associated data, registered in databases, covering for example, spectroscopy, biological activity, thermodynamic properties, and so on. These parameters represent a source for numerous statistical investigations and research into correlations. When such investigations focus on the structural data for fully characterized compounds, very few analyses have been reported and they all depend upon Chemical Abstracts Service (CAS) for their source data.[1,2] When large files are involved, some aspects of our basic knowledge of chemical data depend largely upon these statistics. In this paper, we present original results derived from a CAS file[3] containing 3 424 428 compounds registered through July 1978.

Although statistics on cyclic and heterocyclic systems have been reported,[1,2] some complementary topological information in the structural data are reported here and provide valuable information for the chemist investigating large chemical datasets. Such information is useful for optimization of algorithms which require a statistical knowledge of topological data, such as atomic excentricities or concentric layers around a focus which can be used, for example, when applying the Cahn-Ingold-Prelog rules in computation of configuration. Some unexpected statistical results were obtained and these cannot be interpreted without the use of graph theory. Most of the statistical variables, therefore, will be derived from graph theory rather than from chemical considerations.

REPRESENTATION OF CHEMICAL COMPOUNDS

The expanded formulas of the compounds in the database are coded by means of a DARC-like[4] colored graph, but the statistical study is carried out without any preconceived idea concerning coding rules. The study is aimed essentially at extracting fundamental topological information from the file. The following colored graph terminology is used in the presentation of the results.

In a compound formula:

the graph nodes are the atoms

the graph edges are the chemical bonds

The atoms and bonds are both colored. Each atom assumes one of the 103 colors defined by the Mendeleev Table (the 103 atomic symbols), and each bond takes one of the following values: SI (simple), TA (tautomer), AR (aromatic), DO (double), or TR (triple).

In addition, the atoms are labeled with secondary chromatic information, and in this way, the complete description provided for each molecular structure includes

"Unusual" valency: arithmetic positive value between 0 and 99, a value of zero meaning the usual valency.

Charge: algebraic signed value between −9 and +9 associated with the delocalization flag; localized charge or not.

Isotope: arithmetic positive value giving the integer mass of the isotope. Zero means natural abundance.

Stereochemistry: may be included in the code, but is not registered in this part of the file.

0095-2338/90/1630-0332$02.50/0