(14) Meyer, E.; Schilling, P.; Sens, E. Experiences with input, translation and search in files containing Markush formulae. In *Computer handling of generic chemical structures*, Proceedings of a conference organized by the Chemical Structure Association, University of Sheffield, England, March 26-29, 1984; Barnard, J. M., Ed.; Gower: Aldershot, 1984; pp 82-95.
(15) Lynch, M. F.; Barnard, J. M.; Welford, S. M. Computer storage and retrieval of generic chemical structures in patents. 1. Introduction and general strategy. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 148-151.
(16) Barnard, J. M.; Lynch, M. F.; Welford, S. M. Computer storage and retrieval of generic chemical structures in patents. 2. GENSAL, a formal language for the description of generic chemical structures. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 151-161.
(17) Welford, S. M.; Lynch, M. F.; Barnard, J. M. Computer Storage and retrieval of generic chemical structures in patents. 3. Chemical grammars and their role in the manipulation of chemical structures. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 161-168.
(18) Barnard, J. M.; Lynch, M. F.; Welford, S. M. Computer storage and retrieval of generic chemical structures in patents. 4. An extended connection table representation for generic structures. *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 160-164.
(19) Welford, S. M.; Lynch, M. F.; Barnard, J. M. Computer storage and retrieval of generic chemical structures in patents. 5. Algorithmic generation of fragment descriptors for generic structure screening. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 57-66.
(20) Barnard, J. M.; Lynch, M. F.; Welford, S. M. Computer storage and retrieval of generic chemical structures in patents. 6. An interpreter program for the generic structure description language GENSAL. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 66-71.
(21) Gillet, V. J.; Welford, S. M.; Lynch, M. F.; Willett, P.; Barnard, J. M.; Downs, G. M.; Manson, G.; Thompson, J. Computer storage and retrieval of generic chemical structures in patents. 7. Parallel simulation of a relaxation algorithm for chemical substructure search. *J. Chem. Inf. Comput. Sci.* **1986**, *26*, 118-126.
(22) Gillet, V. J.; Downs, G. M.; Ling, A. (B.); Lynch, M. F.; Venkataram, P.; Wood, J. V.; Dethlefsen, W. Computer storage and retrieval of generic chemical structures in patents. 8. Reduced chemical graphs, and their application in generic chemical structure retrieval. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 126-137.
(23) Downs, G. M.; Gillet, V. J.; Holliday, J. D.; Lynch, M. F. Computer storage and retrieval of generic chemical structures in patents. 9. An algorithm to find the Extended Set of Smallest Rings (ESSR) in structurally explicit generics. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 207-214.
(24) Downs, G. M.; Gillet, V. J.; Holliday, J. D.; Lynch, M. F. Computer storage and retrieval of generic chemical structures in patents. 10. The assignment and logical bubble-up of ring screens for structurally explicit generics. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 215-224.
(25) Lynch, M. F. Generic chemical structures in patents (Markush structures): the research project at the University of Sheffield. *World Patent Inf.* **1986**, *8*, 85-91.
(26) Barnard, J. M. Online graphical searching of Markush structures in patents. *Database* **1987**, *10* (3), 27-34.
(27) Shenton, K. E.; Norton, P.; Ferns, E. A. Generic searching of patent information. In *Chemical structures: the international language of chemistry*, Proceedings of an international conference at the Leeuwenhorst Congress Center, Noordwijkerhout, The Netherlands, May 31-June 4, 1987; Warr, W. A., Ed.; Springer: Heidelberg, 1988; pp 169-178.
(28) Fisanick, W. The Chemical Abstracts Service generic chemical (Markush) structure storage and retrieval capability. 1. Basic concepts. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 145-154.
(29) Stiegler, G.; Maier, B.; Lenz, H. Automatic translation of GENSAL representations of Markush structures into GREMAS fragment codes at IDC. In *Proceedings of the 2nd International Conference on Chemical Information Systems*, Noordwijkerhout, The Netherlands, June 1990; Warr, W. A., Ed.; Springer: Heidelberg, in press.
(30) Love, R. A. Structure drawing software. In *Chemical structure software for personal computers*; Meyer, D. E., Warr, W. A., Love, R. A., Eds.; American Chemical Society: Washington, DC, 1988; pp 9-36.
(31) Nakayama, T.; Fujiwara, Y. Computer representation of generic chemical structures by an extended block-cutpoint tree. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 80-87.
(32) Tokizane, S.; Monjoh, T.; Chihara, H. Computer storage and retrieval of generic chemical structures using structure attributes. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 177-187.

1990 HERMAN SKOLNIK AWARD PAPER

Computer Representation and Handling of Structures: Retrospect and Prospects†

ERNST MEYER‡
Friedelsheimer Strasse 18, D-6700 Ludwigshafen, FRG

Received October 3, 1990

Topological encoding of structures was a necessary supplement to documentation methods. As a practical approach, it was developed first for substructure retrieval in chemical formulas, but it also proved useful in other areas such as reaction retrieval, synthesis planning, semantical and syntactical concept interrelations, patent claims examination, drug design, and even in other disciplines like electrical and mechanical engineering. A survey of three decades of methodological development is given, and some newer trends are indicated.

INTRODUCTION

It becomes more and more difficult to retain an overview of our growing treasure of knowledge, even in a partial area. Documentation methods were developed long before the advent of computers, but indexes, classifications, card files, and similar tools were soon not effective enough. The computer, invented just in time, was at first able to help with the multidimensional search for words and classes, but as a result of the knowledge explosion even index and full-text searches soon became insufficient means, and, especially in chemistry, classification systems became steadily more effort consuming and could not keep pace with the rapid appearance of new concepts and requirements.

Mankind can increase his ability immensely by protheses, tools, and machines. But if one wants to use the computer as a thinking machine, it has to be able to handle not only numbers and character strings but also structures, because our thinking—especially in organic chemistry—proceeds in structures. This need appeared in chemistry very early due to the size and long-life of its treasure of knowledge. Fortunately, chemical structural formulas were quite suitable models for the development of useful computer methods. The approach was supplied by an old branch of mathematics: topology or—more exactly—graph theory.

TOPOLOGY

Graph theory reduces a structure to a set of nodes and the connecting edges. Numbers can be given to both types of elements, and attributes (consisting of words and/or numbers) can be attached to each node or edge in order to characterize them. In this way it becomes possible to localize a substructure in filed structures. It was an American mathematician from Cambridge, MA, Calvin N. Mooers, who suggested in 1951[1] recording chemical structural formulas in this manner on a computer for structure and substructure searches. However, he never practiced this approach himself.

† This paper was presented at the ACS Symposium on Topological Storage of Structures in Chemistry and Related Fields, Washington, DC, Aug 28, 1990.
‡ Formerly with BASF Aktiengesellschaft.

0095-2338/91/1631-0068$02.50/0

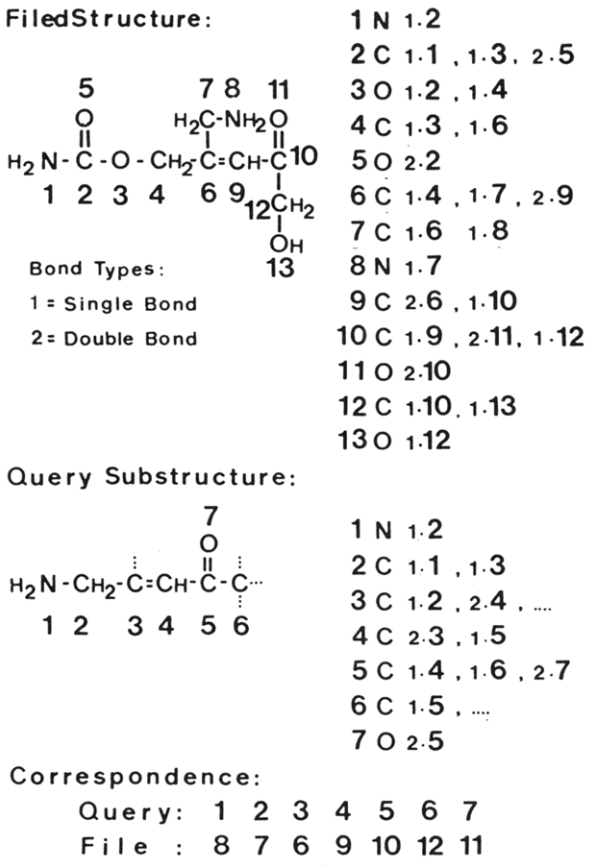


Figure 1. Topological formula encoding.

The principle of topological encoding is very simple: all fragments or atoms, i.e., nodes, are numbered in arbitrary order, and for each of them a line is written in a connection table (Figure 1), consisting for each node of its element symbol and the numbers (i.e., the line addresses) of its neighbors together with the connecting bond types. During a substructure search one does not know the node numbers attached in the file, and therefore arbitrary numbers are used for query encoding. But the computer is able to identify matching nodes by many comparisons according to their element symbols and the attributes of their environments.

STRUCTURAL FORMULAS

In 1958 the emerging reactive dyestuffs had shown that in chemistry new structural features, which had not been encoded before because they did not seem of any interest at input time, can suddenly attain great importance. Incidentally at that time two papers by American authors were published who encoded structural formulas topologically but failed in practical application: The approach of Ray and Kirsch of the National Bureau of Standards[2] was much too time consuming and error prone at the input, and that of Opler and Norton of Dow Chemical Company[3] was not flexible enough for searching, because it used too large fragments as nodes so that not all possible substructures could be searched for.

Ray and Kirsch used non-hydrogen atoms as nodes; Opler and Norton used larger fragments. We now hoped to combine the advantages of the two methods, and we designed a shorthand code system[4] using fragments for input and resolving them into atoms automatically. In this way we encoded about 1000 structural formulas, and our mathematicians wrote substructure search programs. The first successful searches were run early in 1959 with IBM in Paris, because at that time our company did not have an EDP device of its own. From this experiment we learned, however, that the necessary computer time was still too long by several orders of magnitude, and that the encoding was too effort consuming and error prone, too. Our mathematicians lost interest in our problems for several years, but we were not discouraged.

The tasks were now, on the one hand, to automatize fairly well the encoding, and on the other hand to reduce drastically the CPU time needed for substructure searches. For the formula encoding we constructed a formula reading machine[5] (Figure 2) which scanned formulas drawn on transparent grid sheets (Figure 3) and recorded the pulses on punched cards (later on punched tape and finally on magnetic tape). Some of these machines ran for almost two decades with other companies and were replaced by online input later on. They even allowed the encoding of Markush formulas if these did not include more than 27 well-defined substituents. This approach was utilized with great success for the documentation of dyestuff chemistry.[6]

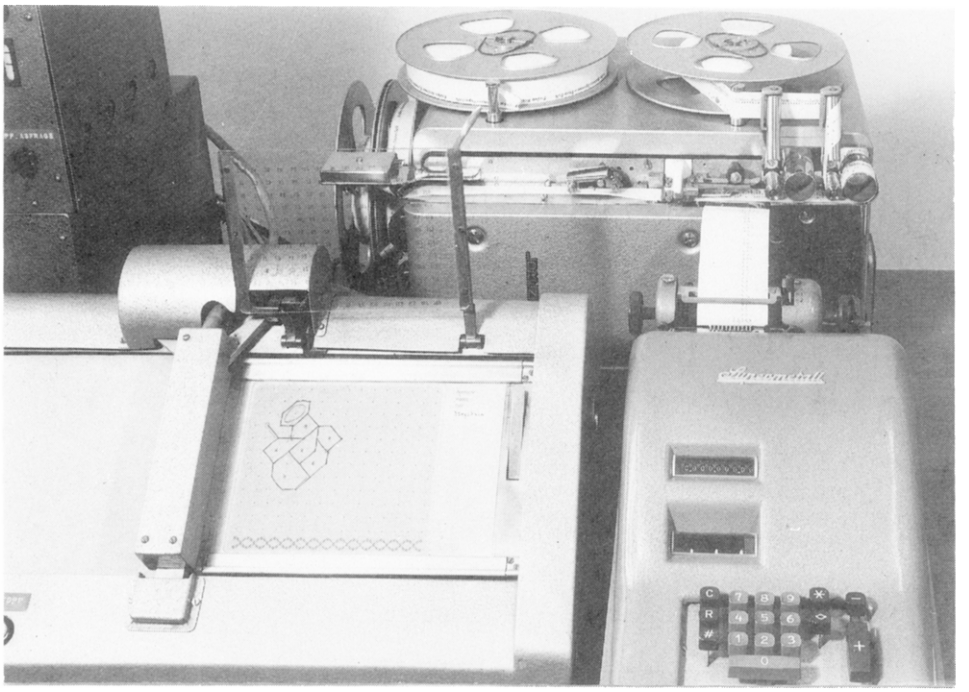


Figure 2. Formula reading machine.

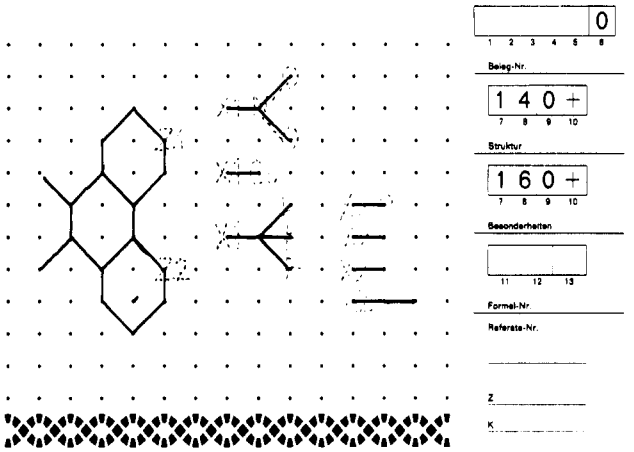


Figure 3. Grid sheet with Markush structure.

In order to perform the substructure search with acceptable computer effort as well, we designed a three-step screening system[7] (Figure 4): Starting from the topological connection table, we first generated a very effective fragment code designed by R. Fugmann of Hoechst AG,[8] i.e., the GREMAS system which is still now in use with IDC (International Documentation in Chemistry). This was a fairly difficult undertaking: for the first time the computer was to carry out tasks which up to then could only be performed by university-trained chemists. A detailed design of flow sheets, however, proved the feasibility of the automatic GREMAS code generation. In $1^1/_2$ years of hard work, E. Sens then wrote the corresponding programs.

The GREMAS search, however, still consumed much too much computer time, and so we added another screening step for preselection[9] which ultimately was based on another ingenious idea of C. N. Mooers,[10] i.e., the superimposed encoding for edge-notched cards (Figure 5), and which, by application to computer comparisons, reduced the search time by at least another order of magnitude. In so doing we also utilized the ability of the computer to compare all bits of a machine word in parallel instead of sequentially bit by bit or character by character. Using this approach we were able to perform substructure searches at a reasonable cost even in very large files such as the CAS Registry File.[11]

GENERATION

With the automatic GREMAS coding, we had demonstrated the general feasibility of the automatic generation of classifying fragment codes, even for Markush formulas if they are sufficiently well-defined conceptually. E. Sens[6] in this way also generated a code for dyestuffs, developed by three German companies because in this field GREMAS was not effective enough. These programs have now been in use for more than two decades. Furthermore, we programmed the translation

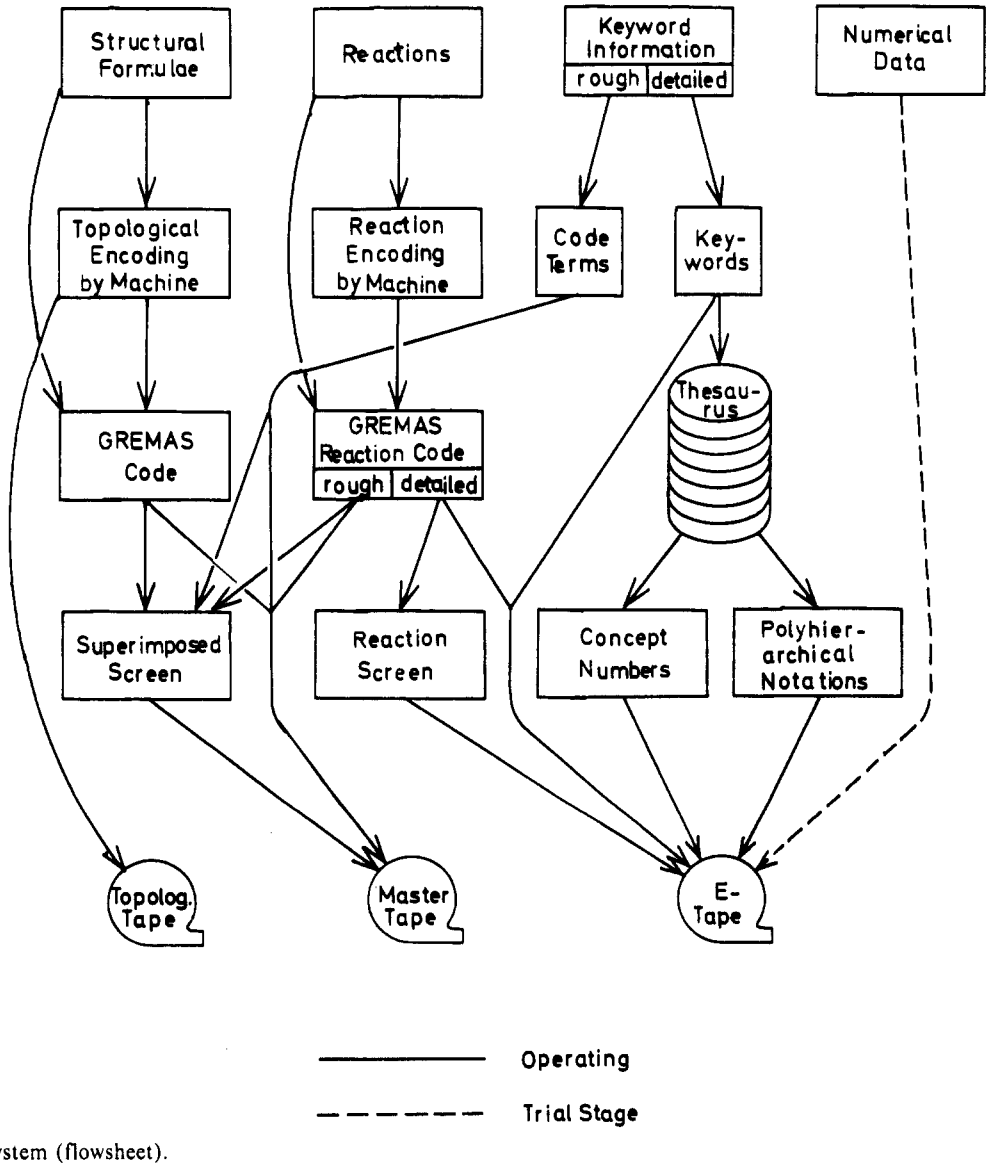
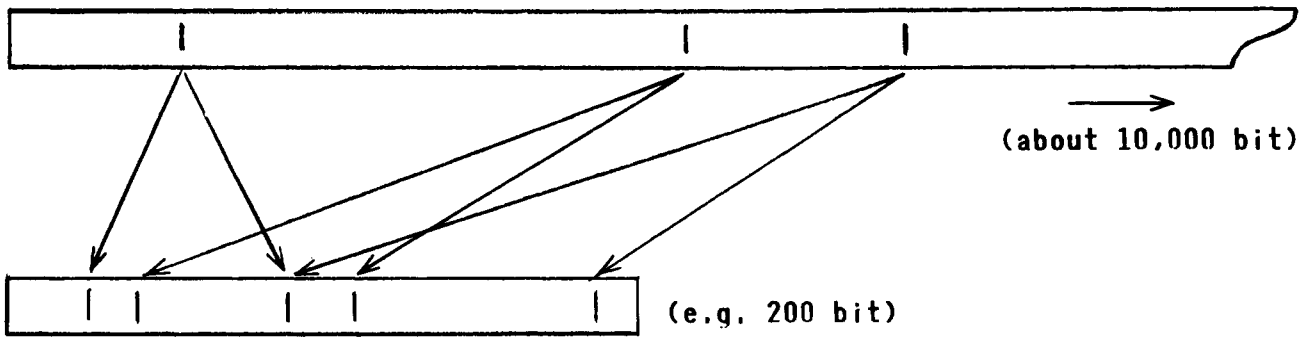


Figure 4. IDC system (flowsheet).



From a sparsely occupied bit string of about 10,000 bits a more occupied shorter string (e.g.of 200 bits) is generated by superimposition of bit combinations for each primary bit. It can be used as a search screen.

Figure 5. Superimposed coding.

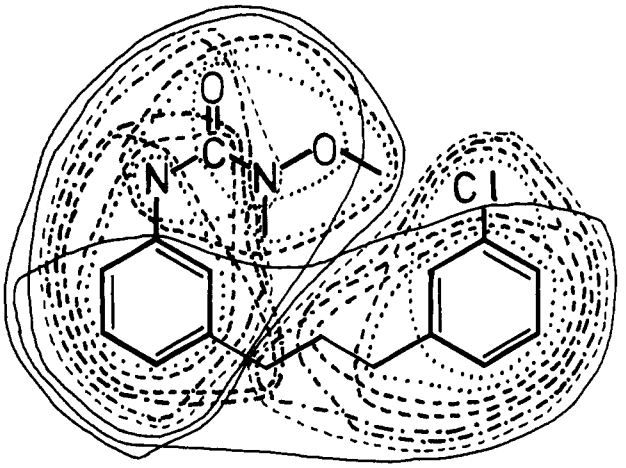


Figure 6. KOWIST: possible pharmacophores.

of the CAS Registry codes into our format, and so we enabled IDC (to which we gave as dowry our approach just like Hoechst did with GREMAS) to conclude an agreement with CAS and save the encoding of journal literature in order to devote all its force to patent documentation input.

The RINGDOC code used by Derwent was also generated later on by U. Klingebiehl and K. Specht.[12] Other notations can be generated from the topological code, too, and here we want to emphasize the achievement of A. J. Lawson, who not only automatized the arrangement of organic compounds into the Beilstein system with this SANDRA program,[13] but even gives very strong assistance in generating the IUPAC nomenclature by the AUTONOM program,[14] a task which overtaxes most chemists.

We could utilize connection tables for another purpose too, namely, for evaluating the compounds tested biologically at BASF, i.e., for the statistical correlation of all their substructures included (as far as they were of suitable sizes for pharmacophores, Figure 6) with their biological effects. We called this approach KOWIST,[15,16] and it allowed us to guide the attention of our colleagues to promising compound classes which had not been realized until then. Using special dialog programs of E. Sens,[15,16] chemists can also construct polyhierarchical trees of substructures (Figure 7) so as to get a better survey of drug classes already realized, in order to identify promising gaps.

All these approaches show that the topological representation of structures provides much more than all previous documentation systems for chemical compounds. A basic jump was performed here for documentation and computer handling of structures, similar to that of changing from the Meisenheimer radical theory to the Kekulé structural formulas a century ago: in spite of a simplification of the mode of thinking and the encoding rules, the new approach was considerably more efficient.

Our system, however, did not yet cater to stereochemistry. For up to four ligands per central atom, this could be done by the order of congeners in the connection table. But this problem becomes much more difficult for inorganic structural chemistry where coordination numbers up to 12 can occur. But even here recently a solution[17] has been implemented and practiced in collaboration between the Gmelin Institute and CHEMPLEX GmbH.

Such systems lead from topology, which describes the relative connection of nodes with each other, i.e., flexible structures, to rigid topography, which defines distances and angles, too. One can attach, of course, lengths and angles to any edges and coordinates to any nodes. So one comes to (partially) rigid structures, i.e., in chemistry from a structural formula to a molecule model which represents the spatial relations more exactly. Because the energy laws and the atom environment determining the bond lengths and angles are fairly well-known, one can calculate the topographical features, i.e., the atom coordinates and molecule surfaces, rather well automatically from the topological code. This is even being tried for very sophisticated and flexible molecules such as polypeptides and proteins. Thus the relatively short connection table bears a lot of useful information regardless of the simplicity of the coding rules.

For a century, chemists thought mainly in Kekulé structures, but nowadays topographic elements and shapes also play a certain role, not only in drug design and crystallography but also for a better understanding of macromolecular substances. Even such structures defined better spatially can be handled, however, adequately by computers.

ADDRESSING

Let us return, however, to the more abstract topological structures. Without being aware of this fact, we had utilized, by jumping through the connection tables, a pointer system, i.e., indirect addressing, a tool familiar to every programmer. The numbers of ligands served us for addressiong other nodes of the graph, thus allowing pursuing pathways even to distant nodes.

Later on we realized (Figure 8) that here we had not only to do with an indirect addressing. Symbolic addressing was a common tool in informatics, and later a contents-addressed search was developed (e.g., in associative memories) where it was not necessary to know either the physical or a logical address of the record searched for, but only a bit string con-

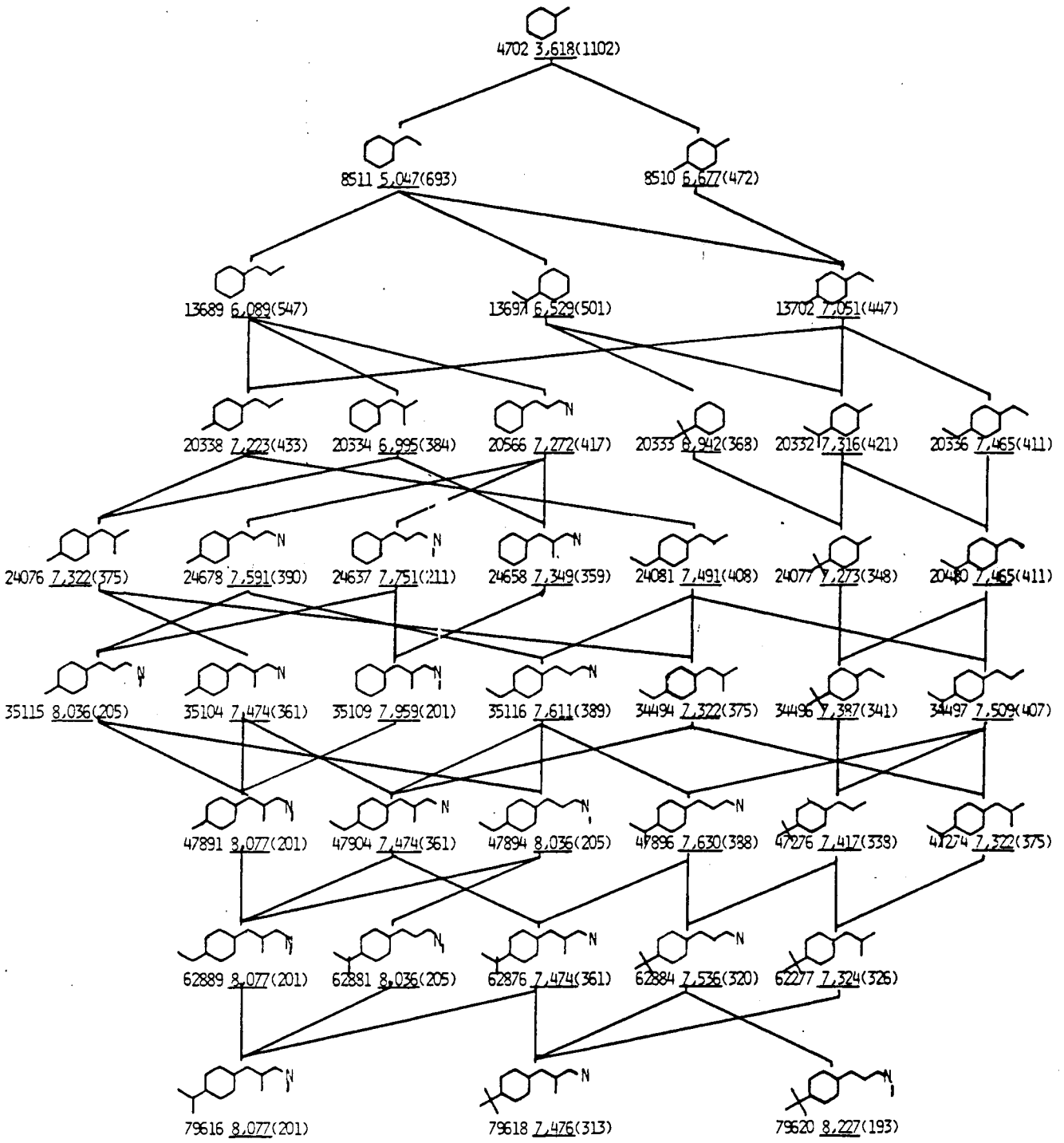


Figure 7. KOWIST: tree of substructures.

tained in it. What was now new in topological search: as an access requirement, one no longer needs to known an exact string but only its conceptual meaning, which could be expressed in memory by many different but equivalent character sequences. Thus, not words but concepts served as addresses or search criteria; not the form was important but the conceptual contents. This was true for structures as well as for thesaurus terms, as we will show now.

THESAURUS

When we thought we had found a satisfactory solution of searching for the "hard" structural formulas, we turned to the next problem, namely to the documentation of "nonstructural facts" and their conceptual interrelations. Here, too, the usual classification systems were not flexible enough for the growing search needs. Each change in the classification system de-

Access Mode	Selection Criterion to be Known
Direct Addressing	Physical Address (e.g. in RAM or Disk Memory)
Indirect (Symbolic) Addressing	Logical Address (e.g. File Name, Record Number, Field Name)
Content Addressing	Formal Contents (e.g. Parts of Well-defined Character Strings)
Semantic Addressing	Meaning of Contents Parts (Conceptual Equivalence of Qery and File Fragment, Locatable e.g. via Thesaurus or Topological Structure Search Algorithm)

Figure 8. Types of addressing.

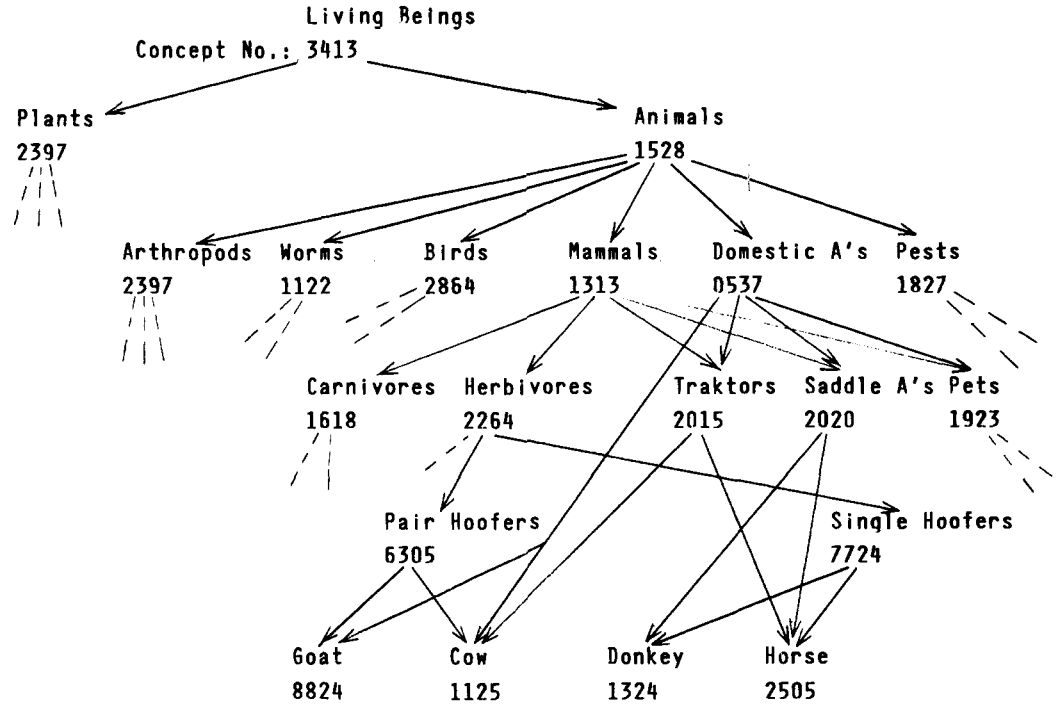


Figure 9. Thesaurus: polyhierarchical tree.

valued the previous files. The full-text search, on the other hand, did not allow the reliable consideration of broader or narrower and related search terms. It was necessary, on the one hand, to define the correspondence between a concept and its denominators, i.e., between our conceptual imaginations and natural languages, and on the other hand to map the structure of our conceptual world so that it could be taken into consideration automatically during search time.

By handling structural formulas we had learned so much about the abilities of computers to map structures that we now dared to tackle the more difficult task of making a sufficiently flexible model of our conceptual world[18] (Figure 9). In doing so, we attached a graph node to every concept, in a way similar to every atom in a big molecule. The "concept record" describing this node could include all of its denominators, even in foreign languages. Correspondingly the semantic interrelations among concepts could be mapped as edges of the graph. Like bond types in a molecule, we also differentiated several types of concept relations in order to describe them more precisely, thus making the search more economical: the abstraction relation (by which the narrower concept includes all features of the broader one), the compound relation (in which the narrower term is part of the broader one), the remaining relations showing a preferred direction (e.g., applications), and the "see also" relation without any direction.

This system enabled us to search retroactively for nonstandardized keywords, like those included in the old GREMAS files, as reliably as if they were classified systematically during input time. And when necessary, this classification could be changed retrospectively by adapting the thesaurus structure to newer realizations and requirements.

Managing the semantic concept interrelations was a step forward, but in some fields the more our files grew the less sufficient it proved to be. Particularly in searching for chemical reactions, syntactic (i.e., document-specific) relations among compounds and/or other concepts gained in importance. In 1963, we published a first model for topological encoding of syntactical relations.[19] We utilized the transparency of the grid sheets of our formula reading machine for copying the educt to product structures (Figure 10). Thus, the machine attached the same numbers to identical atoms, and this could be made a mapping condition for the search. Even reaction centers could be easily realized this way. We only had to input the syntactic structure, i.e., which structures were generated from which other ones and what attribute concepts described the reaction in more detail. Later on, R. Fugmann[20] created a similar but more advanced approach for syntax searches, called "TOSAR", and utilized it mainly for macromolecular chemistry.

Topological encoding of structures was also very suitable for synthesis planning. Particularly the ingenious approach of E. J. Corey and W. T. Wipke[21] was introduced into practice. Initially it was "misused" frequently for reaction retrieval purposes, and its utilization decreased when comfortable programs like ORAC and REACCS became available. Nevertheless, both applications continue to supplement each other.

Topological encoding of hyperstructures will also mean immense progress in another area important for chemistry: for easy input and goalproof retrieval of patent claims. M. F. Lynch[22] conceived an ingenious system and a suitable language (GENSAL) for this purpose. Anybody who has ever encoded or searched for patent claims will appreciate this achievement. Claims became logically so complex that they frequently fill up more than a printed page with a single nested phrase, and an encoder will need hours or days to perceive the contents exactly, and to sort and classify them before starting with the encoding. Using GENSAL it will be possible to write down the logically relevant concepts and relations simply in the order as formulated in the claim without being concious of the complete logical coherence at every moment. Sorting, syntactical analysis, and recording of the logical interrelations can be done by a computer much faster and more reliably than by a human encoder. The search will then be performed in an extended connection table representation (ECTR), i.e., a syntactical hyperstructure of relevant concepts and structural formulas topologically encoded, in a way as flexible as in a substructure search in a single molecule. Even an "up-search" should be possible with this approach. CAS is building up a corresponding patent documentation.

OTHER DISCIPLINES

Recording and retrieval of structures and substructures is of interest not only for chemistry. Other disciplines adopted

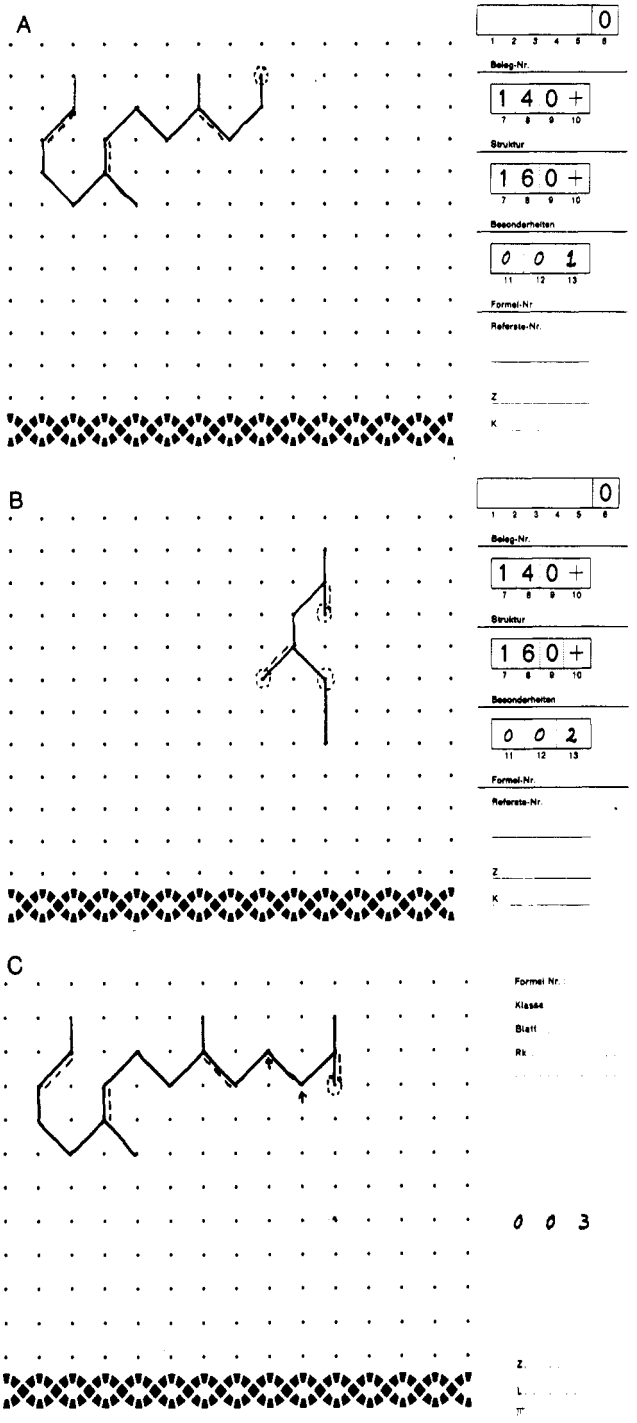


Figure 10. Grid sheets of a reaction.

our methods. Electrical circuits, for instance, are also topological structures which can be encoded and matched exactly by computers. And even in mechanical engineering topological methods have been developed, e.g., for documentation of gear schemes (Figure 11) and rear axle suspensions. Approaches developed in chemistry seem to become popular.

Why, now, was it precisely chemistry which became the forerunner for structure recording and substructure search? There might be several reasons: on the one hand, chemistry is based on an immense number of reproducible facts which are often useful and required even many decades later, e.g., preparation methods for intermediates necessary for further syntheses. Therefore chemical documentation is particularly extensive and worth developing. Secondly, we have to do mostly with "hard" concepts whose meaning and limits are well-defined and not arguable; very few difficulties arose from this side. Mainly, however, it is the physical quantum effect

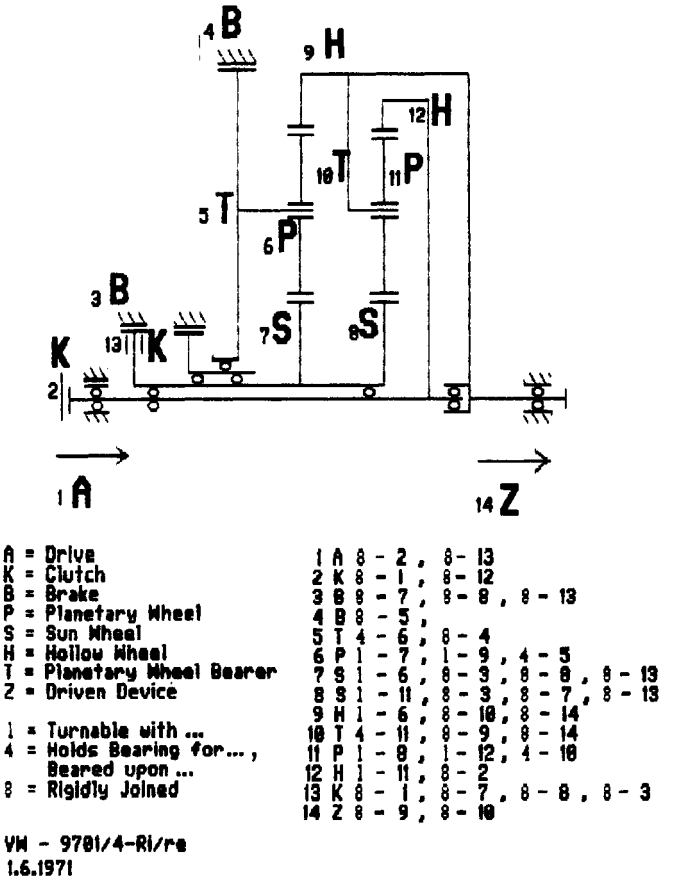


Figure 11. Gear scheme of the Simpson planetary wheel set.

that seems to me to be important: only 100 elements are involved as node values, and (cum grano salis) merely a few types of bonds were sufficient to describe the edges. This simplified the approach considerably.

Thus, we had to handle only a rather simple model case for structure documentation which nevertheless allowed goalproof and useful management of a huge amount and diversity of compounds and facts by relatively simple means. Only those conditions enabled us at that time and stage of computer development to imagine a practicable application, because initially big mainframe computers, although having less capability than the PCs of today, were so expensive that every second of CPU time cost about one dollar. Mass storage, indeed, was very cheap because only magnetic tape was available for this purpose, but reading and writing time was all the more expensive. These obstacles required many tricks in order to keep the costs within affordable limits. Fortunately the computer cost decreased even faster than our files grew. For two decades, however, we had never dared to hope that some day every chemist could have a computer of his own on his desktop, perhaps sufficiently powerful to store, e.g., the whole CAS Registry File, and even to search it for substructures in dialog mode within seconds or minutes.

RECENT DEVELOPMENTS AND TRENDS

Software systems have developed not quite as fast as hardware, but still with impressive speed, concerning their power as well as their user comfort. Considering the past development and its speed, it is difficult to prognosticate the future. We realized that the digit value of information and documentation in chemistry increased considerably with the extent of chemical knowledge. More than ever before, one is also ready to pay for information because savings in research cost will probably justify the expenses, although the true value of a single item of information cannot be measured. Information and know-how have become an important factor in the

production of goods, and thus the requirement for further development is increasing, too. Software engineering has also made great progress so that one can tackle tasks in a much more sophisticated manner than a decade ago. What, then, will be the next steps in managing structures?

We would guess mainly the knowledge-based systems. The present synthesis planning systems are already expert systems in the classic sense, even if computer scientists do not realize or admit this. The CASP system developed from Wipke's SECS program by a group of German and Swiss companies is probably the biggest expert system in the nonmilitary area in the world.

The working style of preparative organic chemists might also strongly be influenced by an integrated system including a comprehensive reaction data base and a synthesis planning program. The synthesis planning systems used to date apply mainly a collection of encoded reaction types to a given end product. They even supply some useful literature citations for the suggested synthesis steps, and they can finish a path branch at wish as soon as they come to an educt which is readily available or included in a catalogue of commercial chemicals or whose preparation is mentioned in *Chemical Abstracts*. Then chemists, if they cannot buy the educt or know well how to prepare it, have to search the literature for suitable preparation methods, evaluate them, and decide which synthesis paths might be the best. In doing so, they get help from Beilstein's and Houben-Weyl's Handbooks, *Chemical Abstracts*, and online data bases. The handbooks are favored, as far as they are sufficiently up to date, because they best collect the relevant preparation methods and briefly summarize and evaluate them.

The Beilstein Online Database would support this work still better if the user could see also the short reaction description on his screen. Computer storage, however, might allow (cum grano salis) even an automatic evaluation and therefore make possible an integrated planning system which improves the selection of promising synthesis path suggestions. There are even ideas how to bridge gaps in pathways which are normally incomplete (in the computer sense) but otherwise viable and reported, by using elements of the Beilstein classification and the automatic analysis of structural formulas[23] in order to include trivial transformations and utilize analogous reactions only if necessary. In this way, synthesis planning and searching reaction data bases will grow together furthermore and will assist the conceptual work of chemists more and more.

Considering the steadily increasing number of useful methods (e.g., spectroscopy, computer simulation, literature search, etc.) that have to be managed by bench chemists, these will much appreciate getting more help from comprehensive, practicable reaction data bases and integrated planning systems. Therefore developments in this directions will be continued and will even be accelerated.

Another field is what is now called *hypertext*. There are already rather efficient programs for this flexible structuring of information giving an integral compound with comfortable flexible user dialog surface even for personal computers, e.g., the CAMS system of C. Edeleanu for PCs, launched early in 1990 as "JUNIPER 1", which is being developed further (in collaboration with the Gmelin Institute) for VAX computers. By means of such systems the user can browse at leisure through the filed knowledge which thus becomes more transparent to him. Transparency, survey, synopsis, and maximal flexibility of query formulation will continue to gain in importance.

Despite all the user friendliness of each single system, the great number of files, surfaces, special features, and query languages impose considerable demands on the documentalist, who is an important partner of the bench chemist at the same time. It is a growing task to equip him, as well as the end user, with tools of increasing power and comfort. Thus, research in this field will attain increasing importance for science, too.

REFERENCES AND NOTES

(1) Mooers, C. N. *Zator Tech. Bull.* **1951**, *59*, 1 ff.
(2) Ray, L. C.; Kirsch, R. A. *Science* **1957**, *126*, 814–819.
(3) Opler, A.; Norton, T. R. *Chem. Eng. News* **1956**, *34*, 2812–2816.
(4) Meyer, E.; Wenke, K. *Nachr. Dok.* **1962**, *13*, 13–19.
(5) Meyer, E. *Nachr. Dok.* **1962**, *13*, 144–146.
(6) Meyer, E.; Schilling, P.; Sens, E. *Computer Handling of Generic Structures*, Proceedings of the CSA Meeting, Sheffield, 1984; Barnard, J. M., Ed.; Gower: Brookfield, VT, 1984; pp 83–95.
(7) Meyer, E. *J. Chem. Doc.* **1969**, *9*, 109–113. Fugmann, R. In *Chemical Information Systems*; Ash, J. E., Hyde, E. Eds.; Ellis Horwood Ltd.: Chichester, 1975; pp 195–226.
(8) Fugmann, R.; et al. *Proc. IUPAC Congr.* **1959**, 331–341. *Ibid. Angew. Chem.* **1961**, *73*, 745–751.
(9) Meyer, E. *Mechanized Information Storage, Retrieval, and Dissemination*, Proceedings of the FID-IFIP Conference, Rome, 1967; Samuelson, K., Ed.; North Holland: Amsterdam, 1968; pp 280–288.
(10) Mooers, C. N. *Am. Doc.* **1951**, *2*, 20–32.
(11) Meyer, E. *Computer Representation and Manipulation of Chemical Information*, Proceedings of the NATO Advanced Study Institute, Noordwijkerhout, 1974; Wipke, W. T., Heller, S. R., Feldmann, R. J., Hyde, E. Eds.; John Wiley & Sons: New York, 1974; pp 105–122.
(12) Klingebiehl, U.; Specht, K. *J. Chem. Inf. Comput. Sci.* **1980**, *20*, 113–116.
(13) Lawson, A. J. In *Graphics for Chemical Structures*, Proceedings of the 192nd ACS Meeting, Anaheim, Sept 1986; Warr, W. A., Ed.; ACS Symposium Series 341; American Chemical Society: Washington, DC, 1987; pp 80–87.
(14) Lawson, A. J.; Goebels, L.; Wisniewski, J. In *Software Development in Chemistry*, 4th ed.; Gasteiger, J., Ed.; Springer Verlag: Berlin, 1990; pp 19–29.
(15) Meyer, E.; Sens, E. *Chemical Structures*, Proceedings of the CSA Conference, Leeuwenhorst, 1987; Warr, W. A., Ed.; Springer Verlag: Berlin, 1988; pp 235–241.
(16) Meyer, E.; Sens, E. *Anal. Chim. Acta* **1988**, *210*, 135–142.
(17) Deplanque, R.; Roth, B. In preparation.
(18) Meyer, E.; Jansen, R. *Nachr. Dok.* **1972**, *23*, 203–210.
(19) Meyer, E. *Proc. 26th Annu. Meet.* **1963**, Part 2, 131. *Angew. Chem. Int. Ed. Engl.* **1965**, *4*, 347–352.
(20) Fugmann, R.; Nickelsen, I.; Winter, J. *Angew. Chem. Int. Ed. Engl.* **1970**, *9*, 589–595.
(21) Corey, E. J.; Wipke, W. T.; *Science* **1969**, *166*, 178–192.
(22) Lynch, M. F.; Barnard, J. M.; Welford, S. M. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 151–161 ff.
(23) Lawson, A. J.; Kallies, H. Multistep Reactions: The RABBIT Approach. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 426–430.