the vocabulary used. Queries containing terms seldom, if ever, found in the indexes, for example, biological data and numerical parameters like "LD50", are particularly suited. In some other cases abstract text searching might be of great disadvantage. Especially in those where the topics are already well covered in the indexes, any possible advantages of such an additional search must be weighed against the loss of precision incurred. This would be of special importance if, instead of the small file we used, the *entire* CA text file (all years and all sections) were to be searched.

It would be inaccurate to compare the results we obtained in searching the combination of fields TK + I with those which might be obtained with CA SEARCH (alone and in conjunction with the abstract). We have searched and retrieved hits within the separate fields only and did not retrieve references if terms satisfying the profile were in different fields. CA SEARCH in its current online usage includes all index entry terms, titles, and keywords in the Basic Index, so such references would probably be retrieved.

We have illustrated that a very simple free-text search technique is sufficient to achieve a substantially increased recall, to provide the nonspecialist with a higher retrieval capability, and, possibly, to attract additional user circles to access the CA text file online.

ACKNOWLEDGMENT

We thank CAS staff for their continued technical support and interest in the results of this study and, in particular, Patricia S. Wilson as well as others who, in addition to reviewing and correcting the draft of this paper, have contributed many valuable suggestions. We also thank the management of BASIC for their encouragement and for providing resources and especially H. Kniess from the Ciba-Geigy Information Center for fruitful discussions and valuable assistance in preparing this paper.

REFERENCES AND NOTES

(1) "Information Tools"; Chemical Abstracts Service: Columbus, OH, 1976. CBAC is presently accessible online via the National Library of Medicine (NLM) System as a component of TOXLINE.

(2) Buntrock, R. E. "Searching Chemical Abstracts vs. CA Condensates". *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 174.

(3) Blake, J. E.; Mathias, V. J.; Patton, J. "*CA Selects*—A Specialized Current Awareness Service". *J. Chem. Inf. Comput. Sci.* **1978**, *18*, 187. Blake, J. E.; Ebe, T. "Abstract Text Searching for *CA Selects*". Presented at the 2nd Chemical Congress of the North American Continent, 180th National Meeting of the American Chemical Society, Las Vegas, Nevada, 26 Aug 1980.

(4) "CAS ONLINE & Search Services Catalog"; Chemical Abstracts Service: Columbus, OH, 1984; p 13.

(5) Barker, F. H.; Veal, D. C.; Wyatt, B. K. "Comparative Efficiency of Searching Titles, Abstracts, and Index Terms in a Free-Text Data Base". *J. Doc.* **1972**, *28*, 22.

(6) Wagers, R. "Effective Searching in Database Abstracts". *Online* **1983**, *7* (5), 60.

(7) Durkin, K.; Egeland, J.; Garson, L. R.; Terrant, S. W. "An Experiment to Study the Online Use of a Full-Text Primary Journal Database". Presented at the 4th International Online Information Meeting, 9–11 Dec 1980, London, England.

(8) Cohen, S. M.; Schermer, C. A.; Garson, L. R. "Experimental Program for Online Access to ACS Primary Documents". *J. Chem. Inf. Comput. Sci.* **1980**, *20*, 247.

An Algorithm for Chemical Superstructure Searching

PETER WILLETT

Department of Information Studies, University of Sheffield, Western Bank,
Sheffield S10 2TN, United Kingdom

Received October 29, 1984

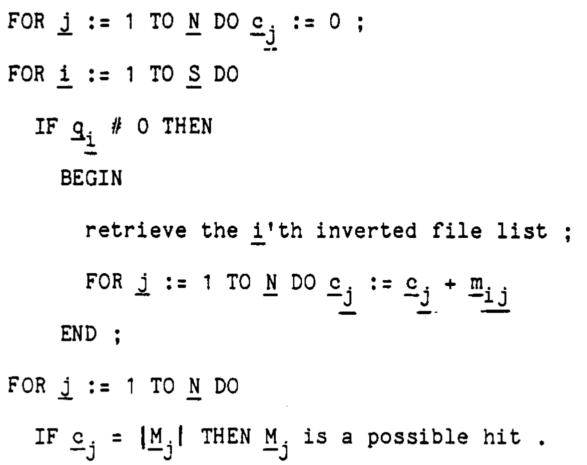
Chemical superstructure searching involves the identification of those molecules that are contained within a given query structure. This paper presents an algorithm for carrying out such searches that may involve fewer accesses to backing storage than a previously described algorithm.

SUBSTRUCTURE AND SUPERSTRUCTURE SEARCH

An important facility in computer-based chemical information systems is the ability to carry out chemical *substructure* searching.[1] Given a query structure, Q, and a set of N molecules $\{M_j\}$ $(1 \leq j \leq N)$, a substructure search results in the identification of those molecules that contain Q as a substructure. Substructure searching is a special case of the more general subgraph isomorphism problem that involves determining whether one graph is a subgraph of another; this problem has been studied extensively and is known to be NP complete.[2] Because of this, substructure search systems[3,4] operate in two stages, with a simple and rapid initial search mechanism being used to eliminate the great majority of the molecules in the file; only those few compounds that pass this initial screening search then undergo the computationally demanding atom-by-atom substructure search.

The screening search is effected by defining a set of features, called screens, that are used to characterize the molecules in the file and the query structure Q; a wide range of substructural features may be used for this purpose, as is illustrated by the screening mechanisms used in the CAS ONLINE system.[5] For some molecule, M_j, to be a *possible hit* in the substructure search, i.e., one that needs to be processed by the atom-by-atom algorithm, it must contain all of the screens that have been assigned to Q: such screens will be referred to here as *query screens*. The screening search may be performed efficiently by setting up an inverted file that contains a series of lists, one for each of the possible screens, with the ith list containing the identifiers of those molecules to which the ith screen has been assigned. The intersection of the lists corresponding to the query screens results in a list containing the identifiers for all of the possible hits that must be processed in the atom-by-atom search.

Wipke and Rogers[6] have recently discussed the inverse of chemical substructure searching, which they call *superstructure* searching. A superstructure search results in the identification of those molecules that are substructures of Q, a search facility that is of importance in computer-aided synthesis design programs. Efficiency of operation is again achieved by the use of an initial screening search based on an inverted file; however, the inverted file is used in a quite different manner from that employed for substructure search. Whereas the latter involves taking the intersection of the query screen lists to identify possible hits, superstructure searching involves taking the union of the lists corresponding to the screens that have not been assigned to Q to identify all of the *definite nonhits*: the atom-by-atom superstructure search is thus restricted to those molecules not eliminated by the



```
FOR j := 1 TO N DO c  := 0 ;
                    j
FOR i := 1 TO S DO
  IF q  # 0 THEN
      i
    BEGIN
      retrieve the i'th inverted file list ;
      FOR j := 1 TO N DO c  := c  + m
                          j     j    ij
    END ;
FOR j := 1 TO N DO
  IF c  = |M | THEN M  is a possible hit .
      j     j        j
```

Figure 1.

screening search. This paper describes an alternative mechanism for carrying out the screening stage of a superstructure search that may have advantages in some circumstances over the Wipke–Rogers algorithm.

THE ALGORITHM

The algorithm to be described here is a simple modification of one that was developed for the calculation of intermolecular similarity coefficients in studies of the automatic classification of chemical structures.[7] It is assumed that the query molecule Q is represented for search by a binary vector

$$(q_1, q_2, ..., q_i, ..., q_S)$$

where q_i is a bit that denotes the presence ($q_i = 1$) or absence ($q_i = 0$) of the ith screen in Q and S is the total number of screens that are available for screening purposes. The notation $|Q|$ will be used to denote the number of non-zero bits in the vector. Each of the N molecules in the file has a comparable vector denoting the screens that have been assigned to it: taken together, these vectors form the inverted file. The ith list in this file ($1 \leq i \leq S$) is a binary vector of the form

$$(m_{i1}, m_{i2}, ..., m_{ij}, ..., m_{iN})$$

where m_{ij} denotes the presence ($m_{ij} = 1$) or absence ($m_{ij} = 0$) of the ith screen in the jth molecule, M_j. Let C be an N-element integer array, the jth element of which, c_j, contains the number of query screens present in M_j: this information may be obtained by *cumulating* the inverted file lists corresponding to the query screens, rather than taking their intersection or union as in conventional Boolean processing. Once the $|Q|$ lists have been added together, the test for superstructure search is trivial: if c_j, the number of query screens that are present in M_j, is not the same as $|M_j|$, the total number of non-zero screens assigned to M_j, then M_j must have been assigned at least one screen that is not a query screen, and thus, Q cannot be a superstructure of M_j; i.e., $c_j = |M_j|$ is a necessary, but not sufficient, condition for Q to be a superstructure of M_j. This algorithm is detailed in Figure 1 in a PASCAL-like notation.

DISCUSSION

Wipke and Rogers enumerate four types of search strategy that may be of importance in synthesis design programs. These are an exact match search for identity, substructure and superstructure search as described above, and a similarity search in which structures are required that are similar, in some sense, to the query molecule. The algorithm in Figure 1 may be used to carry out all of these types of search merely by altering the final IF clause. For an identity search, this clause becomes

$$\text{IF } (c_j = |Q|) \text{ AND } (c_j = |M_j|) \text{ THEN } M_j \text{ is a possible hit}$$

while substructure search involves the use of

$$\text{IF } c_j = |Q| \text{ THEN } M_j \text{ is a possible hit}$$

In both cases, the possible hits may then be passed on for the (sub)graph isomorphism search. It should be noted that these modifications are not necessarily the methods of choice since alternative procedures may be more efficient, specifically the use of hashing for exact match and intersection of the query lists for substructure match; however, they serve to emphasise the general nature of the algorithm. Similarity searching may be carried out by using the c_j values to calculate some measure of intermolecular similarity[8] between Q and each of the N molecules; these similarities may then be sorted so as to identify those molecules that are most closely related to Q. The use of the algorithm in an interactive chemical similarity search system will be reported shortly.

Two other points may be made about the algorithm. First, the number of structures passed on for atom-by-atom searching will be just the same as the number passed on by the Wipke–Rogers algorithm, despite the quite disparate modes of operation; in both cases, only those molecules will be processed in the second-stage search whose associated screens are entirely contained within the set of query screens. Second, the algorithm is best suited to files that are based upon the use of dedicated bit strings in which each position in the binary vector characterizing a molecule is reserved for a single screen. The algorithm will function less efficiently if several screens share a single position, as in the CAS ONLINE[5] or WRAIR[9] screening systems, since it is then possible for a molecule to be accepted as a possible superstructure hit even though it has been assigned a screen that is absent from Q; such a system will tend to lessen the screenout obtained.

The processing of the inverted file lists by the use of additions, rather than by the Boolean operations of intersection or union, might suggest that the algorithm described above would be markedly less efficient than that described by Wipke and Rogers. The advantage of the present approach is that only $|Q|$ inverted file lists need to be processed: since the inverted file for a large collection of compounds will need to be maintained on a backing storage device, this corresponds to a total of ca. $|Q| + 1$ disc accesses if it is assumed that each inverted list, and the N-element table containing the $|M_j|$ values, may be retrieved in a single disc access. The Wipke–Rogers algorithm, conversely, requires the union of all of the $S - |Q|$ lists corresponding to screens that have not been assigned to Q. This may not be too much of a problem if a dedicated processor is available or if only a few tens of screens are used for the characterization of the molecules in a collection, although this must inevitably result in poor screenout.[10] However, most substructure search systems use many hundreds of screens in a time-sharing environment, and thus, the Wipke–Rogers algorithm will necessitate very large numbers of disc accesses, even if some blocking scheme is adopted so that several inverted file lists can be retrieved in a single disc access.

ACKNOWLEDGMENT

My thanks are due to Prof. M. F. Lynch and Dr. S. M. Welford for helpful comments on an earlier draft of the manuscript.

REFERENCES AND NOTES

(1) Ash, J. E.; Chubb, P. A.; Ward, S. E.; Welford, S. M.; Willett, P. "Communication, Storage and Retrieval of Chemical Information"; Ellis Horwood: Chichester, England, 1984.

(2) Tarjan, R. E. "Graphic Algorithms in Chemical Computation". *ACS Symp. Ser.* **1977**, *46*, 1–19.

(3) Sussenguth, E. H. "A Graph-Theoretic Algorithm for Matching Chemical Structures". *J. Chem. Doc.* **1965**, *5*, 36-43.

(4) Figueras, J. "Substructure Search by Set Reduction". *J. Chem. Doc.* **1972**, *12*, 237-244.

(5) Dittmar, P. G.; Farmer, N. A.; Fisanick, W.; Haines, R. C.; Mockus, J. "The CAS ONLINE Search System. 1. General System Design and Selection, Generation and Use of Search Screens". *J. Chem. Inf. Comput. Sci.* **1983**, *13*, 93-102.

(6) Wipke, W. T.; Rogers, D. "Artificial Intelligence in Organic Synthesis. SST: Starting Material Selection Strategies. An Application of Superstructure Search". *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 71-81.

(7) Willett, P. "The Calculation of Inter-Molecular Similarity Coefficients Using an Inverted File Algorithm". *Anal. Chim. Acta* **1982**, *138*, 339-342.

(8) Adamson, G. W.; Bush, J. A. "A Comparison of the Performance of Some Similarity and Dissimilarity Measures in the Automatic Classification of Chemical Structures". *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 55-58.

(9) Feldman, A.; Hodes, L. "An Efficient Design for Chemical Structure Searching. I. The Screens". *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 147-152.

(10) Willett, P. "The Effect of Screen Set Size on Retrieval from Chemical Substructure Search Systems". *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 253-255.

A New System for the Designation of Chemical Compounds. 2. Coding of Cyclic Compounds†

RONALD C. READ

Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Canada N2L 3G1

Received September 1, 1983

In this paper a procedure is given for coding cyclic compounds, that is, deriving a unique designation for any such compound. The method described in part 1 for coding acyclic compounds is first used to detect and remove all side chains from the molecule, their codes being recorded for later use. The remainder of the molecule, which, by definition, is taken to be the ring structure, is then coded by first classifying its atoms and then constructing a special "walk" within the structure. This leads to a unique and concise designation for the ring structure and a standard numbering of its atoms. This numbering enables the locations of the side chains to be precisely specified. The resulting designation for the whole compound has something in common with some existing systems of nomenclature in so far as it specifies first the side chains and then the ring structure to which they are attached. Unlike other systems, however, the coding process does not require the use of lists of ring structures, such as those in the *Parent Compound Handbook*; the designation can be computed in full from the structural formula of the compound or equivalent information. The procedure is very amenable to automatic computation and has already been implemented by a Fortran program of no great length.

(1) INTRODUCTION

Part 1 of this paper[8] described a system for coding acyclic chemical compounds, the main feature of which was an algorithm that computed a unique code or "name" for any acyclic compound by means of operations performed on the structural formula of the molecule being coded (or on anything equivalent to a structural formula, such as a connection table). These operations were purely graph theoretical, making no call on chemical knowledge or intuition, and the resulting code was made up of the customary symbols used in organic chemistry (atom symbols, symbols for bonds, etc.) carrying their usual meanings. Moreover, the code was in a form that was meaningful to a chemist, either immediately or with only a small amount of paperwork.

Typical examples of codes produced by the algorithm are $CH(CH_3)_2(CH_2.OH)$, $C(CH_3)_3((CH_2)_2.CH_3)$, and $C(CH{:}CH_2)(=NH).CH_2.C(CI_3){:}CH.CH(CH_3)_2$

In this paper the more difficult problem of coding cyclic compounds is tackled. The main objectives are as before: to produce a coding algorithm that is automatic, that does not require chemical intuition (and which, therefore, can be easily programmed on a computer), and that produces a code that is at least partly intelligible at sight and is easily decoded in full, even by hand. Naturally, it must meet the basic requirement of any system of nomenclature that is to be used for information retrieval, namely, that each structural formula must give rise to a unique code, no matter in what form the formula is originally presented to the algorithm. These requirements have been met in the present system.

Although the question was discussed in part 1, it would be well to reiterate here the two main ways in which this system is an improvement on existing systems. First, it gives a finite set of rules from which the designation corresponding to *any* structural formula can be derived, without the need to consult any book of reference such as the *Parent Compound Handbook*. This desirable property is, to be sure, shared by a few existing systems—the one due to Morgan[7] is a good example. But these systems are designed for use in a purely computerized environment, and the designations that they produce are not readily intelligible to the chemist. By contrast, designations produced by the present system give a great deal of information about the compound in a readily visible form. The nature of the side chains is clear from the first part of the designation. The tail of the designation, which expresses the ring structure, is more opaque but can easily be decoded, by hand, to yield the ring structure and its canonical numbering.

The general idea just described, that of producing a "name" made up of components representing the ring structure of the molecule and the side chains attached to it, is one that is found in certain other nomenclature systems. In these systems, however, the ring structure is designated by a trivial name, which has to be looked up in, say, the *Parent Compound Handbook*.[2] In the present system, the name for the ring structure can be computed automatically from the structural formula without any lookup process. (Needless to say, the name that this automatic procedure produces is not a name in the sense of something that can be pronounced; it is a string

† The research for this paper was supported by Grant A8142 from the National Science and Engineering Research Council of Canada.

0095-2338/85/1625-0116$01.50/0 © 1985 American Chemical Society