Table V. Statistics on Use of Search Operators

search operator	search level	total no. of uses	uses per session
AND	document	1397	3.25
SAME	field or paragraph	437	1.02
WITH	sentence	654	1.52
ADJ	word	746	1.73
OR		538	1.25
NOT		65	0.15
\$	truncation	777	1.81

did not seem to do a great deal of online browsing through the retrieved documents.

Analysis of User–System Dialog Tapes. Table V provides an indication of the usage of the various search operators. To effectively search a full-text file, it is normally necessary to use a search operator that restricts search terms to some unit of text smaller than the total article. In the BRS language, the SAME, WITH, and ADJ search operators restrict terms to the paragraph, sentence, and word level, respectively. The AND operator requires search terms to be only within the same document. While there are certain specific situations in which the AND operator must be used, most searches against a full-text file should utilize one of the more restrictive search operators to avoid large numbers of false drops.

The total usage of the three more restrictive search operators combined (SAME, WITH, and ADJ) is greater than the usage of the AND operator, as would be expected in a full-text file. However, use of the AND operator an average of 3.25 times per sessions is surprising and is probably due to unfamiliarity with full-text searching. Use of the document-level search operator is probably the result of habits formed in searching the much more common bibliographic files in which potential penalites in terms of false hits are not as great as in the case of full-text files.

CONCLUSIONS

The major conclusions of the study are summarized as follows. (1) Online full-text searching is a powerful method. The extreme depth of indexing, which is essentially total indexing, is particularly valuable in that it allows searchers to locate items of information embedded deeply within the text of primary documents that may not be the main point of the article.[7] (2) Full-text searching is a valuable addition to the information retrievel methods available to the scientific community. Utility of the file discussed here would be greatly enhanced by addition of non-ACS journals. (3) A cost of \$100 per hour is too high a connect hour charge for many users, in particular users from academic institutions. (4) There was a tendency on the part of some users to apply the more familiar bibliographic searching techniques to the full-text file. To obtain good precision in full-text files, search strategies quite unlike those used in bibliographic files are usually needed. User education in full-text searching will be both useful and/or necessary.

ACKNOWLEDGMENT

We gratefully acknowledge the assistance of John T. Keys (ACS Washington) for his programming efforts in processing the data from the online questionnaire.

REFERENCES AND NOTES

(1) Schermer, C. "The Primary Journal System: A Case Study". *Graphic Commun. Comput. Assoc. J.* **1978**, *2*, 19–24.

(2) Cohen, S. M.; Schermer, C. A.; Garson, L. R. "Experimental Program for Online Access for ACS Primary Documents". *J. Chem. Inf. Comput. Sci.* **1980**, *20*, 247–252.

(3) Standard Distribution Format is described in "Chemical Abstracts Service Specifications Manual for Computer-Readable Files in Standard Distribution Format"; Chemical Abstracts Service: Columbus, OH.

(4) "BRS System Reference Manual"; Bibliographic Retrieval Services, Inc.: Latham, NY.

(5) Garson, L. R.; Cohen, S. M. "Users' Manual Primary Journal Database ACS Full-Text File"; American Chemical Society: Washington, DC, 1983.

(6) In the illustration of displayed text, there are certain words delimited by plus signs (e.g., +GAMMA+). These are spelled-out expansions of special characters that are not represented in the character set used in the online system. A list of these words is given in footnote 5.

(7) Indexing in this context refers to making all words in the text searchable, except for trivial stop words (such as of, the, and, etc.). It does not refer to the intellectual effort of indexing such as performed by abstracting and indexing services. Of course, all the ACS journals are covered by *Chemical Abstracts* as well as other services. Whether the lack of controlled indexing and standardized nomenclature within the full-text file itself is a deficiency remains to be determined.

A Relaxation Algorithm for Generic Chemical Structure Screening

ANNETTE VON SCHOLLEY[†]

Department of Information Studies, University of Sheffield, Sheffield S10 2TN, U.K.

Received November 7, 1983

A safe screening method for structure search within a database of generic chemical structures is described, and some results are shown. The search algorithm is based on a relaxation technique. The generic structures are represented by the Extended Connectivity Table Representation from which a data structure is set up that enables the rapid execution of the search algorithm. This data structure enables generic expressions with alternative substituents, variable positions of substituents, multipliers for singly or doubly connected substituents, and generic expressions like alkyl to be handled without need for enumeration.

INTRODUCTION

In the past a lot of quite sophisticated structure storage and retrieval systems for databases of specific chemical structures have been developed (e.g., CAS ONLINE, DARC, GREMAS). COUSIN, an online system implemented at the Upjohn Co. in Michigan, also works with a specific database, but for the query input it is possible to specify variable substituents by using the Rk notation.[1,2]

Systems dealing with generic databases, however, lag far behind those for specific structures. The Central Patent Index (CPI) of Derwent Publications,[3] Gremas of IDC (International Documentation of Chemistry),[4] and the IFI/Plenum database[5] all rely on manually assigned fragment codes.

At the University of Sheffield, a system for storage and retrieval of generic chemical structures is being developed. Generic structures are encoded in an unambiguous formal language called GENSAL,[6] which is intelligible for a chemist

[†] Address correspondence to the author at Beilstein-Institut, Varrentrappstr. 40-42, 6000 Frankfurt am Main 90, Federal Republic of Germany.

0095-2338/84/1624-0235\$01.50/0 © 1984 American Chemical Society

and compact to store in a computer. An Interpreter[7] automatically converts these expressions into an Extended Connectivity Table Representation (ECTR).[8] This ECTR can be used for different methods of structure search.

GENERAL STRATEGIES FOR THE SEARCH SYSTEM

A search system for generic structures should be able to answer the following types of question:[9] Is a specific molecule contained in a generic expression? Is a generic substructure query contained in a given generic structure (regardless of whether the query is partly or totally in variable parts of the structure)? Do two generic structures have a specific molecule in common?

Exact structure matching belongs to the class of NP-complete problems.[10] Because no efficient algorithms are known for these problems, many screening systems have been developed for substructure search with specific chemical structures. By eliminating structures for which a match with a given query is certainly impossible, these screening methods rapidly reduce the number of structures in a database for which an atom-by-atom match must be used.

In considering generic structures as opposed to specific structures, a bitscreen search becomes less accurate, and an atom-by-atom match is much more complicated and time consuming. Therefore, a three-step search is suggested: (1) A bitscreen search quickly eliminates structures that cannot match the query. (2) A relaxation algorithm is performed on the structures found by step 1. (3) An exact atom-by-atom match is necessary only for those structures that have passed step 2, if at all.

Step 1 is under development by Welford.[11] For each structure, a bitstring representing limited-environment fragments is stored. These fragments are automatically generated from the ECTR[12] and include fragments from within constant or variable parts and those that bridge the connections between these.

Step 2 is the topic of this paper. A relaxation technique is used to examine if atoms of a query are locally consistent with atoms of a given structure. Locally consistent assignments are a necessary but not a sufficient condition for structure match. Therefore, relaxation is a safe screening method in that it identifies all structures that match a given query, in addition to some number of structures for which the match is less exact. The algorithm described in this paper works only with the connectivity between atoms and does not take account of bond orders.

Step 3 is still under consideration.

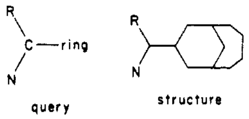
Although the aim of this work was to deal with generic structures, an interesting question is whether the relaxation algorithm could be used as an intermediate step in a specific database to improve the speed of structure searching. Tables I and II show results of substructure search for a specific query in a file of specific structures.

WHAT IS RELAXATION?

Among other definitions of the term relaxation, this term is used to describe a class of iterative methods for classification or constraint analysis. A value of some kind for each node of a structure graph is calculated by a two-step algorithm: (1) Each node is assigned an initial value, usually some characteristic of the node itself. (2) The value of each node is improved by examining the corresponding values of the neighbors of this node. By use of the newly calculated values of each node, this process is repeated until further iterations do not improve the result.

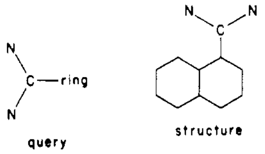
In the past, relaxation methods were mainly studied for image segmentation.[13-16] Other applications are breaking of substitution ciphers.[17,18] Ugi and Schubert[19] use relaxation for canonicalization of chemical structures. The calculation of extended connectivity values, as used, for example, in the Morgan algorithm,[20,21] the automatic reaction site detection procedure of Lynch and Willett,[21] the Augmented Connectivity Formula,[22] and a stereochemically unique naming algorithm by Wipke and Dyott[23] provide other examples of the use of relaxation techniques in chemical documentation. Kitchen and Krishnamurthy[24] and Kitchen and Rosenfeld[25] suggest relaxation as a screening method for chemical structures but have not described its practical application. However, the term relaxation has only recently been used in the literature, and thus, the word relaxation is not mentioned in all of the references.

Table I



ring size	match	iter	
3	no	3	0.04
4	no	4	0.06
5	no	4	0.07
6	yes	6	0.11
7	no	6	0.13
8	no	7	0.17
9	yes	9	0.23
10	no	9	0.24

Table II



ring size	match	iter	CPU (s)
3	no	3	0.06
4	no	4	0.07
5	no	5	0.09
6	yes	8	0.15
7	no	6	0.15
8	yes[a]	7	0.20
9	no	7	0.19
10	yes	8	0.26

[a] Incorrect match.

The above definition of the term relaxation excludes the algorithms for matching chemical structures of Figueras[26] and Sussenguth.[27] Sussenguth works with pairs of corresponding sets. Each pair consists of a set of query atoms and a set of structure atoms that have the same atomic symbol or the same bond type. Possible mappings of query atoms and structure atoms are reduced by intersection of two pairs of sets. Further eliminations are achieved by intersection of pairs of neighboring atoms of query and structure. The algorithm terminates when further intersections fail to improve the result. This iteration is analogous to step 2 of the above definition.

Figueras suggests a set-reduction algorithm related to Sussenguth's; bond orders are considered. After the first-order search fails, higher order connection tables are explicitly set up for query and database structure by using subsequently more remote neighbors. This algorithm terminates when higher order connection tables do not improve the result.

PROGRAM RELAX

In the following text, the term "structure" is used for the items of a database and the term "query" for search expressions. The term "label" always refers to query atoms. These query atom labels are represented in the examples by integers, while lower case alphabetics are used for structure atoms.

Chart I

```
CONST
MAXSET        = maximal number of nodes in the query
MAXCONGENERS = maximal number of neighbours for a node

TYPE
SETTYPE     = SET of 0..MAXSET;
PSTRUCTURE = ^TSTRUCTURE;
PNEIGHBOUR = ^TNEIGHBOUR;

TNEIGHBOUR = RECORD     (* list of alternative neighbours to one position *)
   ATOMPOINTER  :  PSTRUCTURE; (* pointer to the neighbour in TSTRUCTURE *)
   NEXT         : PNEIGHBOUR
END;

TSTRUCTURE = RECORD            (* list of nodes of the structure *)
   MATCH        : SETTYPE;    (* all labels of the query, that can be matched *)
   NEIGHBOUR    : ARRAY[1..MAXCONGENERS] OF PNEIGHBOUR;
   MAXFREQUENCY : INTEGER;
   NEXT         : PSTRUCTURE
END;
```

Here, query is to be understood as describing a substructure, in that the query data structure described below contains sufficient information only to satisfy the criterion for substructure search, namely, embedment.

Program RELAX is written in the computer language Pascal, and it is implemented on the Sheffield University PR1ME 750 minicomputer. It enables generic structures and queries with alternative substituents, variable positions of substituents, multipliers for singly or doubly connected substituents, and generic expressions like alkyl to be handled without need for enumeration.

Some generic expressions, like alkyl, describe an unlimited number of specific structures. Others cover a limited number of specific radical structures. In this case, a Chemical Grammar[28] theoretically makes possible the enumeration of all possibilities. But even such limited generic structures easily contain several thousands of specific structures. Therefore, total enumeration is not a practicable way for dealing with a generic database.

DATA STRUCTURES

It is possible to represent specific data structures by a graph in which each node represents exactly one atom. Generic structures may describe a large or even an unlimited number of atoms. Therefore, a data structure is chosen in which each node may represent a group of atoms (Chart I). This data structure has similarities with the Composite Augmented Atom (CAA).[11] The difference is that it contains pointers to the neighbors instead of explicitly recording the atomic symbols of those neighbors.

This data structure allows the representation of an unlimited number of atoms by a limited number of nodes. Atoms of substituents with multipliers occur only once in this representation. The neighbors of each node contain all essential or optional neighbors of all atoms in this group. If there exists a chain like C(n), the node contains itself as a possible neighbor.

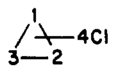
The data structure PSTRUCTURE enables the rapid execution of the relaxation algorithm, but it contains less information than the ECTR and is no longer unambiguous. It does not use any of the logical relationships or exclusions that may be present in the definition of generic structures. In the case of multipliers, it may indicate too many alternatives for a neighbor. The errors produced by this simplification are only of the kind that result in too many structures being found to match a given query, so that recall is never compromised. If a substituent has variable positions of attachment, all of these are described in the data structure, but after the relaxation, additional procedures check whether the MAXFREQUENCY of each node is greater than or equal to the number of nodes that are required by the query.

This data structure is also suitable for dealing with generic expressions like alkyl. In the ECTR, these expressions are stored as lists of parameters.[6] Parameter identifiers include carbon count, ternary branches, quaternary branches, ring counts, etc. Each parameter is defined by a range of integers. The information stored in these parameters is used to calculate nodes in PSTRUCTURE with the appropriate neighbors.

Further investigations need to be made to determine whether it is more economical to store PSTRUCTURE explicitly in the database than to generate the ECTR each time from the GENSAL notation, from which PSTRUCTURE is then derived.

REPRESENTATION OF THE QUERY

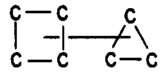
The nodes of the query are defined similarly to those of the structure. For each node, its minimal frequency and the alternative neighbors for each position are stored. Only essential neighbors are considered. If there exists alternatives for one neighbor but this neighbor has a fixed position, then all these neighbors are stored in one set. If, like in



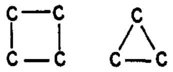
node	neighbors
1	[2] [3]
2	[1] [3]
3	[1] [2]
4	[1, 2, 3]

a connection has a fixed connection point to one component and a variable connection to the other compound, then the resulting data structure is nonsymmetric.

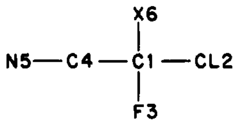
The above example shows how variable connections between two partial structures in a query expression are treated. Alternative neighbors are enclosed in square brackets. The bond between the cyclopropane ring and the chlorine atom is considered only within the neighbors of the chlorine. RELAX is unable to distinguish between a query consisting of two components both with variable points of connections and a query with the same unconnected components. For example



is treated as



For the query, a data structure, in the form of a list, is created in which for each atom type encountered in the query the labels of the nodes of that type are stored as a set. For example, from the query below, the following list is first produced:



atom type	labels
C	[1,4]
CL	[2]
F	[3]
N	[5]
X	[6]

This list is then enlarged to a form that enables easy handling of generic atom types like HALOGEN ("X"), HETEROATOM ("HT"), or ANYATOM ("A"). A chlorine atom can be matched to a halogen and a halogen to a fluorine, but of course, a bromine should not be matched to a chlorine:

atom type	labels list
C	[1, 4]
CL	[2, 6]
F	[3, 6]
N	[5]
X	[2, 3, 6]
HT	[2, 3, 5, 6]
A	[1, 2, 3, 4, 5, 6]

SEARCH ALGORITHM

Each node of the database structure must now be associated with those nodes of the query structure that it may match. This is also a set of the same atomic type. The use of the

Chart II

```
procedure RELAXATION
begin
   fcr each ncde l of the structure dc
   fcr each label K in MATCH cf I do
      if none cf the alternatives of each neighbcur cf K
      can be matched tc one of the alternatives cf a different
      neighbcur cf I
      then
      eliminate K frcm MATCH cf 1
end
```

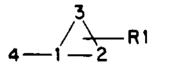
enlarged list produced from the query allows this assignment to be made rapidly.

Procedure RELAXATION (Chart II) operates on the query label sets that have been assigned to the database structure. The elimination of labels is made by using only the number and identity of the neighbors of each query and each structure node in such a way that for each neighbor of the query node that may correspond to this structure node in question there is at least one matching neighbor of the structure node.

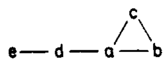
RELAXATION is repeated until the last iteration fails to bring any improvement. Each iteration uses information about nodes further away from the node under consideration without the time-consuming process of explicitly looking at subsequent neighbors of the neighbors of this node.

The implementation of procedure RELAX is based on set operations. All alternatives for one query neighbor are stored in one set. The matching labels of alternative neighbors of a structure node are combined in one set before each iteration of the relaxation. To determine whether a structure neighbor bears the labels required by a query neighbor, it is necessary only to determine whether the intersection of these two sets is not empty. That means, the alternatives for the neighbor of the query atom and the alternatives for one neighbor of the structure atom have at least one label in common.

The following describes the operation of the search algorithm on a simple example:



query
R1 = 5F or 6CL



structure
R1 = fCl or gBr

Examining whether the above query is a substrate of the structure proceeds by the following steps. The initial matching of labels to the structure, assigned on atom type only is

node	MATCH
a	(1, 2, 3, 4)
b	(1, 2, 3, 4)
c	(1, 2, 3, 4)
d	(1, 2, 3, 4)
e	(1, 2, 3, 4)
f	(6)
g	( )

For each label of each node of the database structure, the first iteration of relaxation is as follows.

Node a. No elimination is possible because all labels are supported by neighbors with the required labels. In order to bear label 1, a node must have three neighbors with labels 2, 3, and 4; the neighbors b, c, and d fulfill this condition. Label 2 requires two neighbors with labels 1 and 3; these labels are found in the label set of the neighbors b–d. Label 3 can only be assigned to a node that has two neighbors with the labels 1 and 2; these labels are assigned to atoms b, c, and d. Label 4 needs to be supported by a neighbor with label 1; at this state of the relaxation algorithm, all three neighbors of atom a bear label 1.

Node b. In order to bear label 1, a node must have three different neighbors with labels 2, 3, and 4. Node b has three neighbors, a, c, and f or g, but they do not fulfill this condition. Node a bears the labels 1, 2, 3, and 4, node c contains 1, 2, 3, and 4, and the combination of node f and g is 6. Therefore, label 1 is removed from MATCH. Label 2 demands two neighbors with labels 1 and 3. Label 3 has to be supported by two neighbors with the labels 1 and 2. Label 4 can only be matched to atoms that have a neighbor with label 1. All these labels can be found in the labels of the neighbors of atom b; therefore, the labels 2, 3, and 4 remain in the label set:

node	MATCH
a	(1, 2, 3, 4)
b	(2, 3, 4)
c	(1, 2, 3, 4)
d	(1, 2, 3, 4)
e	(1, 2, 3, 4)
f	(6)
g	( )

Node c. Equivalent to atom b, label 1 is removed, and the labels 2, 3, and 4 remain:

node	MATCH
a	(1, 2, 3, 4)
b	(2, 3, 4)
c	(2, 3, 4)
d	(1, 2, 3, 4)
e	(1, 2, 3, 4)
f	(6)
g	( )

Node d. Label 1 is eliminated because it requires three neighbors with labels 2, 3, and 4:

node	MATCH
a	(1, 2, 3, 4)
b	(2, 3, 4)
c	(2, 3, 4)
d	(2, 3, 4)
e	(1, 2, 3, 4)
f	(6)
g	( )

Node e. All labels are removed from this atom:

node	MATCH
a	(1, 2, 3, 4)
b	(2, 3, 4)
c	(2, 3, 4)
d	(2, 3, 4)
e	
f	(6)
g	( )

Node f. No eliminations are possible.

Node g. No eliminations are possible.

Between each iteration of procedure RELAXATION, a check is made to see if every label can be assigned to a node of the structure. Otherwise, no match is possible and further search would be useless. Frequencies of query and structure nodes are considered during this step.

A generalized heuristic described by Cheng and Huang[29] is used to improve speed and results of the relaxation. If a label can be assigned to a number of nodes in the structure that equals its minimal frequency of occurrences, then every structure node that is assigned this label can bear only this label.

In the above example, label 1 can be assigned only to node a. Therefore, node a is not allowed to contain any other label, and the label set that results at the end of the first iteration is

node	MATCH
a	(1)
b	(2, 3, 4)
c	(2, 3, 4)
d	(2, 3, 4)
e	( )
f	(6)
g	( )

During the second iteration of the relaxation algorithm, eliminations are possible for node d only. Label 2 is removed from node d, because the required neighbors of label 2, i.e., labels 1 and 3, are not supported by the label sets of nodes a and e. Label 3 can only be assigned to a node with neighbors bearing the labels 1 and 2. The neighbors a and e do not fulfill this condition; therefore, label 3 is eliminated:

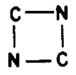
node	MATCH
a	(1)
b	(2, 3, 4)
c	(2, 3, 4)
d	(4)
e	( )
f	(6)
g	( )

No further eliminations are possible; the algorithm terminates. All labels of the query can still be matched to atoms of the structure. Therefore, the query is found as a substructure of the structure.

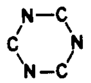
Additional procedures examine the frequencies of query and structure nodes. Although R1 can occupy variable positions and contains alternatives, queries that contain both a chlorine and a bromine atom at fixed positions or in two different substituent groups, or queries with more than one chlorine, cannot be matched to this structure. These queries would be found only if R1 in the structure had an appropriate multiplier.

SUBSTRUCTURE SEARCH

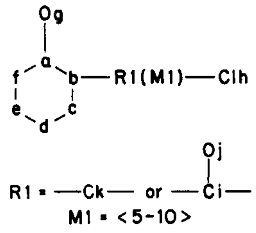
In most cases relaxation gives quite good results. If the query consists only of a ring or an unbranched chain comprising atoms of identical atom type, then this query is found in every structure that contains a ring of this atom type, regardless of the ringsize; these are not common queries, however. RELAX also finds, for example



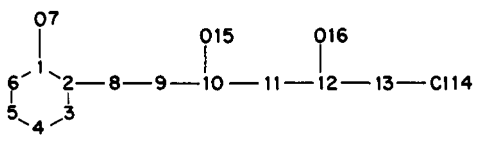
in



However, if the query comprises one or more substituted rings, the results improve (see Tables I and II):



The above generic expression contains 2016 specific structures. They can all be found quite rapidly by using the PSTRUCTURE data structure described above. For example, for the query

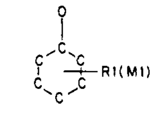


the search time was 0.210 s.

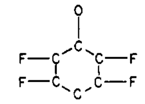
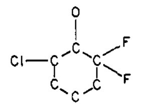
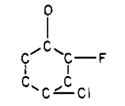
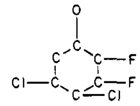
SPECIFIC MOLECULES IN COMMON

If two structures have at least one specific molecule in common, the conditions for local consistency between query and structure nodes are all essential neighbors of the query label must be matched to either essential or optional neighbors of the structure node and all essential neighbors of the structure node must be matched to either essential or optional neighbors of the query label.

Table III

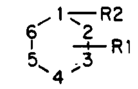


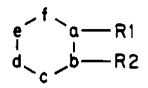
structure
R1 = F/Cl
M1 = <2 - 3>

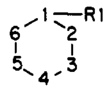
query	found	iter	CPU (s)
	no	3	0.11
	yes	3	0.10
	yes	3	0.10
	yes[a]	3	0.11

[a] Incorrect match.

One possibility for investigating if structures A and B have





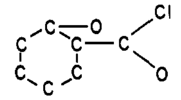


A
R1 = O8/F7
R2 = alkyl SB Cl13 and O12

B
R1 = Oj/Nk
R2 = Og-Ch-Cli

C
R1 = alkyl SB Cl and OH

a specific molecule in common is first to find out if A is a substructure of B; this examines the first condition. If so, then the second condition is examined by investigating whether B is a substructure of A. With this method, it is possible to detect that structure A has a specific molecule in common with structure B but that neither structures A and C nor structures B and C have any specific molecule in common. The common molecule of structures A and B is



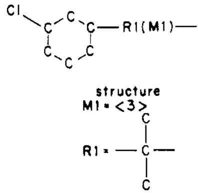
But the algorithm described here does not identify this common molecule, nor does it look for any common substructure of two structures.

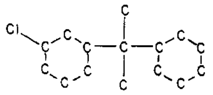
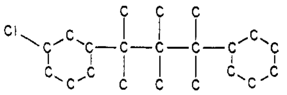
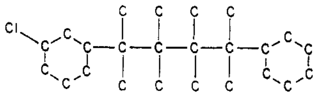
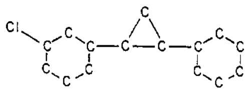
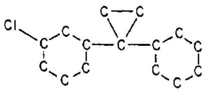
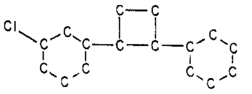
RESULTS

So far it has only been possible to test the program with a moderate database. Further investigations are necessary to obtain accurate screenout and precision figures and to determine whether the efficiency of the algorithm is sufficient enough to cope with the screenout obtained from an initial bitstring screening. However, the results so far seem to be quite promising.

Tables I–VI show some of the results. The examples are more choosen to show the possibilities of the algorithm than to represent typical queries. Table III shows the treatment of singly connected substituents with multipliers and variable positions. Tables IV and V deal with doubly connected sub-

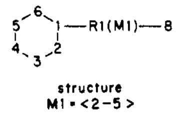
Table IV

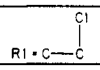
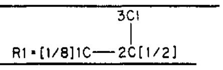
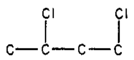
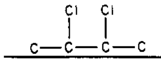


query	found	iter	CPU (s)
	yes[a]	5	0.66
	yes	5	0.79
	no	2	0.54
	no	2	0.54
	no	2	0.59
	no		0.31

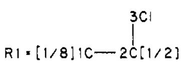
[a] Incorrect match.

Table V[a]



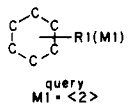
						
query	found	iter	CPU (s)	found	iter	CPU (s)
	yes	3	0.11	yes	3	0.11
	yes	3	0.12	no	2	0.11

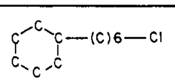
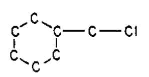
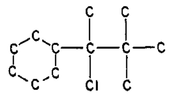
[a] For the second structure in this table, a position set is used during input of the GENSAL notation. Position sets are enclosed in square brackets. The expression before the substituent defines possible connection points at the constant structure; the set after the definition of the substituent defines possible connection points within the substituent. The below query describes a directed polymer. This expression



means that atom 1 of the parental structure has to be connected to atom 1 of the substituent and atom 8 of the parental structure has to be connected to atom 2 of the substituent, and it requires as the substituent repeats that they are connected head to tail.

Table VI[a]



M1 = ⟨2⟩	structure	found	iter	CPU (s)
R1 = alkyl ⟨2-5⟩ SB Cl		yes[b]	7	0.44
R1 = alkyl SB Cl		yes	2	0.13
R1 = alkyl ⟨1-6⟩ Q ⟨1⟩ SB Cl		no	1	0.10
		yes	3	0.24

[a] The queries show an example for input in GENSAL notation[6] for the parameter lists of generic expressions. The values between the first angular brackets define the range for the carbon count; the value in the angular bracket after the symbol Q defines the numbers of quaternary atoms. SB is the abbrevation for substituted. [b] Incorrect match.

stituents, and Table VI deals with homologous series. The CPU times are those needed for the searching algorithm and do not include the time spent in establishing data structures and making the initial label assignments. The CPU times are only approximate values because they depend on how many users are working at the computer. The last iteration that does not improve the result is also counted in each case.

ACKNOWLEDGMENT

This work was made possible by support from a studentship of the science committee of NATO from the agency of the DAAD. Thanks are due to the University of Sheffield Department of Information Studies for providing facilities for this research and to Prof. M. F. Lynch, Dr. J. M. Barnard, Dr. S. M. Welford, and Dr. P. Willett for provision of software and valuable discussions.

REFERENCES AND NOTES

(1) Hagadone, T. R.; Howe, W. J. "Molecular Substructure Searching: Minicomputer-Based Query Execution". *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 182–186.

(2) Howe, W. J.; Hagadone, T. R. "Molecular Substructure Searching: Computer Graphics and Query Entry Methology". *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 8–15.

(3) Kaback, S. M. "Chemical Structure Searching in Derwent's WPI". *J. Chem. Inf. Comput. Sci.* **1980**, *20*, 1–6.

(4) Fugmann, R. "The IDC System". In "Chemical Information Systems"; Ash, J. E.; Hyde, E., Eds.; Chichester: 1975; pp 195–225.

(5) Balent, M. Z.; Emberger, J. M. "A Uniqure Chemical Fragmentation System for Indexing Patent Literature". *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 100–104.

(6) Barnard, J. M.; Lynch, M. E.; 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.

(7) 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.

(8) Barnard, J. M.; Lynch, M. F.; Welford,f S. M. "Computer Storage and Retrieval of Generic Chemical Structures in Patents. 4. An Extended Connection Table Representation (ECTR) for Generic Structures". *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 160–164.

(9) 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*, 148–150.

(10) O'Korn, L. J. "Algorithms for Chemical Computations". *ACS Symp. Ser.* **1977**, *No. 46*, 122–148.

(11) Welford, S. M. "Topological Grammars and the Generation of Limited-Environment Fragments for Generic Chemical Structures". Ph.D. Thesis, University of Sheffield, 1982.

(12) 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.

(13) Kuschel, S. A.; Page, C. V. "Augmented Relaxation Labeling and Dynamic Relaxation Labeling". *IEEE Trans. Pat. Anal. Mach. Intell.* **1982**, *PAMI-4* (6), 676–682.

(14) Nagin, P. A.; Hanson, A. R.; Riseman, E. M. "Studies in Global and Local Histogram-Guided Relaxation Algorithms". *IEEE Trans. Pat. Anal. Mach. Intell.* **1982**, *PAMI-4* (3), 263–276.

(15) Zucker, S. W.; Krishnamurthy, E. V.; Haar, R. L. "Relaxation Processes for Scene Labelling: Convergence, Speed and Stability". *IEEE Trans. Syst. Man. Cybern.* **1978**, *SMC-8*, 41–48.

(16) Shapiro, L. G.; Haralik, R. M. "Structural Descriptions and Inexact Matching". *IEEE Trans. Syst. Man. Cybern.* **1981**, *PAMI-3* (5), 504–519.

(17) Peleg, S.; Rosenfeld, A. "Breaking Substitution Ciphers Using a Relaxation Algorithm". *Commun. ACM* **1979**, *22* (11), 598.

(18) Hunter, D. G. N.; McKenzie, H. R.; "Experiments with Relaxation Algorithms for Breaking Simple Substitution Ciphers". *Comput. J.* **1983**, *26* (1).

(19) Schubert, W.; Ugi, I. "Constitutional Chemistry and Unique Descriptors of Molecules". *J. Am. Chem. Soc.* **1978**, *100*, 37–41.

(20) Morgan, H. L. "The Generation of a Unique Machine Description for Chemical Structures—A Technique Developed at Chemical Abstracts Service". *J. Chem. Doc.* **1965**, 107–113.

(21) Lynch, M. F.; Willett, P. "The Automatic Detection of Chemical Reaction Sites". *J. Chem. Inf. Comput. Sci.* **1978**, *18*, 154–159.

(22) Freeland, R. G.; Funk, S. A.; O'Korn, L. J. Wilson, G. A. "The Chemical Abstracts Service Chemical Registry System. 2. Augmented Connectivity Molecular Formulae". *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 94–98.

(23) Wipke, W. T.; Dyott, T. M. "Stereochemically Unique Naming Algorithm". *J. Am. Chem. Soc.* **1974**, *96*, 4834–4842.

(24) Kitchen, L.; Krishnamurthy, E. V. "Fast, Parallel Relaxation Screening for Chemical Patent Data-Base Search". *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 44–48.

(25) Kitchen, L.; Rosenfeld, A. "Discrete Relaxation for Matching Relational Structures". *IEEE Trans. Syst. Man. Cybern.* **1979**, *SMC-9*, 869–874.

(26) Figueras, J. "Substructure Search by Set reduction". *J. Chem. Doc.* **1972**, *12* (4), 237–244.

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

(28) Welford, S. M.; Lynch, M. E.; 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.* **1982**, *21*, 161–168.

(29) Cheng, J. K.; Huang, T. S. "A Subgraph Isomorphism Algorithm Using Resolution". *Pattern Recog.* **1981**, *13* (5), 371–379.

DARC-SYNOPSYS. Designing Specific Reaction Data Banks: Application to KETO-REACT

R. PICCHIOTTINO,* G. GEORGOULIS, G. SICOURI, A. PANAYE, and J. E. DUBOIS*

Association pour la Recherche et le Developpement en Informatique Chimique, 25 rue Jussieu, 75005 Paris, France, and Institut de Topologie et de Dynamique des Systemes, associé au CNRS, Université Paris 7, 75005 Paris, France

Received February 3, 1984

On the basis of the Entity/Relationship approach, specific reaction data banks have been designed by modeling data in a logical scheme and by proposing a compatible physical scheme that takes access optimizations into account. This architecture determines the general organization of the IGRES-RECRE acquisition and retrieval software incorporating original computer validation procedures that improve the quality, exactitude, and coherence of reaction data and thereby ensure bank reliability. The RECRE software interactivity assists retrieval by letting the user break down a question into formalized elementary requests (DARC structure and substructure searching, nonstructural searching, logical operations, output) in a quasi-natural language. This methodology is illustrated on the KETO-REACT data bank in the framework of the DARC-SYNOPSYS expert system.

INTRODUCTION

Tools for computer-aided design in chemistry, developed within the last 10 years,[1,2] contribute to the progress of expert systems in artifical intelligence.[3-6] Many systems for the computer-aided organic synthesis design of a given molecule have been built over this period.[7]

According to Gund,[8] the design of a synthesis involves an overall approach, which is the planning stage, and a local approach by which the experimental description of a reaction becomes accessible. Today's computer-aided synthesis systems[7] use both approaches simultaneously, at the expense of the retrieval of detailed reaction data. Actually, though such systems are built from an in-depth literature analysis,[9] there is currently a real need[8] for tried and proven procedures to (*i*) *constitute specific reaction data banks and* (*ii*) *access their data*. The specific reaction data banks implemented in the DARC-SYNOPSYS expert system[6,10-14] have been designed to meet this need.

Unlike chemical compounds,[15] which are generally described by "hard data",[16] reactions are mostly described by "soft data",[16] so they are more difficult to define and organize. Both types of data serve to define the field of a reaction that can be organized conveniently on the basis of its structural data.[14]

Our approach to reaction modeling consists in identifying this set of data and in conducting a logical analysis of these data and of the relationships between them in order to attain a clear description of the various functions (input, validation, search, output) of a specific reaction data bank. This results in the proposal of a data model irrespective of its subsequent use, e.g., documentary search or computer-aided synthesis. The logical and physical schemes resulting from this data model express the architecture of the specific reaction data banks that is intended for straightforward integration into a chemical Data Base Management System (DBMS). This methodology is illustrated hereafter on the KETO-REACT data bank in the framework of the DARC-SYNOPSYS expert system.[6]

ORIGIN AND NATURE OF DATA

Bank Coverage. The coverage of a reaction data bank is expressed by its *scope* and *exhaustivity*, which should be defined precisely. The DARC-SYNOPSYS data banks are compiled from existing organic chemistry periodicals or compendiums,[17,18] whereas other systems rely mostly on compendiums. Although the exhaustivity of a compilation of reactions of general interest has been recognized as an illusory goal,[19] this remains conceivable in a limited and strictly defined area.

The methodology we proposed in 1969 for selecting reaction data for the preparation of families of compounds deals with the two above-mentioned factors.[20] In applying this methodology to the synthesis of ketones, a bibliography made it possible in a first stage to select from the literature those articles mentioning at least one method for preparing aliphatic and acyclic ketones having a first or immediate environment limited to B, designated hereafter as *first EB ketones* (Figure 1). Such an approach provides an exhaustivity criterion that

0095-2338/84/1624-0241$01.50/0 © 1984 American Chemical Society