in chemical information, both academic and nonacademic, I also feel that the training should not be just in on-line methods. A solid grounding in the use of all chemical information sources is needed, and on-line instruction should be integrated into the remainder of the course work. The ACS Division of Chemical Information has a new Education Committee, chaired by Arleen Somerville, and its mission deals with a wide range of chemical information instruction and awareness topics, both academic and nonacademic.

There are occasions when on-line services can be priced too low. Several years ago the MEDLINE/TOXLINE users community would ask NLM for various file improvements, improvements that were already featured in the ORBIT systems. Although sympathetic, the answer would often be that the on-line charge was so low the user would be able to afford the noted inefficiencies.

Probably because of a long history of cost recovery for novel services, providers of information services in the private sector have long been concerned with cost effectiveness of, and productivity in the use of, those novel services. Although not addressed previously in this paper, database quality is also of prime importance. Well-indexed and -abstracted material makes for cost-effective searching because intellectual effort spent in creating the file facilitates productive use of the file. After all, the total cost of the use of a file or service is the sum of all charges for system use plus the "people" costs associated with that use. To facilitate more productive end user or customer use of information, more work needs to be done on effective search recording, constructive sort, merge, and edit programs, and also training of end users in the use of information services.

REFERENCES AND NOTES

(1) Almond, J. R.; Nelson, C. H. "Improvements in Cost Effectiveness in On-Line Searching. I. Predictive Model Based on Search Cost Analysis". *J. Chem. Inf. Comput. Sci.* **1978**, *18*, 13-15.

(2) Almond, J. R.; Nelson, C. H. "Improvements in Cost-Effectiveness in On-Line Searching. II. File Structure, Searchable Fields, and Software Contributions to Cost-Effectiveness in Searching Commercial Data Bases for U.S. Patents". *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 222-227.

(3) Magson, M. S. "Modelling On-Line Cost-Effectiveness". *Aslib Proc.* **1980**, *32*, 35-41.

(4) Lancaster, F. W. "Some Considerations Relating to the Cost-Effectiveness of Online Services in Libraries". *Aslib Proc.* **1981**, *33*, 10-14.

(5) (a) Buntrock, R. E. "Searching *Chemical Abstracts* vs. *CA Condensates*". *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 174-176. (b) Buntrock, R. E.; Mulvihill, J. G. "The American Petroleum Institute (API) Data Bases: Comparison of On-Line and Batch Searching"; ASIS Mid-Year Meeting, 3rd, Johnstown, PA, May 17, 1974.

(6) Kaminecki, R. M.; Llewellen, P. A.; Schipma, P. B. "Searching *Chemical Abstracts Condensates*, On-Line and Batch". *J. Chem. Inf. Comput. Sci.* **1976**, *16*, 125-127.

(7) Prewitt, B. G. "Searching the *Chemical Abstracts Condensates* Data Base via Two On-Line Systems". *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 177-183.

(8) Pemberton, J. K. "The Inverted File". *Online* (*Weston, Conn.*) **1979**, *3*, 6-7.

(9) Hoover, R. E. "A Comparison of Three Commercial ONLINE Vendors". *Online* (*Weston, Conn.*) **1979**, *3*, 15-21.

(10) Buntrock, R. E. "The Effect of the Searching Environment on Search Performance". *Online* (*Weston, Conn.*) **1979**, *3*, 10-13.

(11) Boyce, B. R.; Gillen, E. J. "Is It More Cost-Effective to Print On- or Offline?". *Ref. Q.* **1981**, *21*, 117-120.

(12) Stewart, A. K. "The 1200 Baud Experience". *Online* (*Weston, Conn.*) **1978**, *2*, 13-18.

(13) Fortune, J.; Horwich, J.; Schwartz, R. "Use of a Word Processor Interfaced to a Mini Computer to Facilitate On-Line Searching". "Abstracts of Papers", 175th National Meeting of the American Chemical Society, Honolulu, HI, Apr 1-6, 1979; ACS/CSJ Chemical Congress; CHIF 54.

(14) Dedert, P. L. "Electronic Editing of Online Search Results—Choices and Experiences". Tri-Society Symposium (ACS-CINF, ASIS SIG-BC, SLA Chem. Div.), Columbus, OH, Oct 17, 1982; Abstr.

(15) Stewart, A. K. "Selection and Use of Equipment to Manipulate Search Output". Proceedings of the ASIS Annual Meeting, 44th, Washington, DC, Oct 25-30, 1981, pp 251-252.

(16) Buntrock, R. E. "Chemcorner". *Database* **1981**, *5*, 79-81.

Computer Storage and Retrieval of Generic Chemical Structures in Patents. 5. Algorithmic Generation of Fragment Descriptors for Generic Structure Screening

STEPHEN M. WELFORD, MICHAEL F. LYNCH,* and JOHN M. BARNARD

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

Received December 8, 1983

Considerations for the use of limited-environment screens for screening generic chemical structures are discussed. The general strategy and detailed procedures for the automatic generation of screens from the extended connection table representation (ECTR) of generic chemical structures are described. A bitscreen record for generic database structures and specific, generic, and substructure queries is described, and a number of screening algorithms are proposed.

INTRODUCTION

Previous papers in this series have introduced a novel approach to the computer representation and searching of generic chemical structures in patents.[1-5] This approach employs a systematic language GENSAL[2] for the interactive input of structure information in both graphic and textual form, during which process a machine-based representation of the generic structure is automatically created.[6] This representation is called the extended connection table representation (ECTR) and has been described previously.[4]

A topological structure grammar TOPOGRAM has been proposed,[3] and its associated generative and recognitive algorithms have since been developed. The potential uses of TOPOGRAM in the context of a generic structure storage and retrieval system have been described under three areas of application. These are, first, as a means of providing for computer storage a compact description of generic nomenclatural expressions, and particularly of homologous series terms, second, as a device for generating structural fragments characteristic of these generic expressions for use in screening and subsequent structure-matching procedures, and, third, as a generalized mechanism for determining the inclusion of specific substructures within the radical classes defined by generic nomenclatural expressions.

The systematic nature of the GENSAL language enables a complete topological representation of many types of generic structural descriptions to be created automatically in the form of an ECTR. The organization of the ECTR, whose com-

0095-2338/84/1624-0057$01.50/0

plexity of form reflects the complex topology typical of these structures, necessitates novel algorithms both for the generation of fragment descriptors for screening and for more sophisticated structure-matching strategies capable of greater precision. The need for several levels of screening and substructure search was suggested previously,[1] the lower levels to be implemented inexpensively and executed rapidly by means of degenerate bitscreen records and subsequent levels to use more informative representations, each successive level including additional relational information derived from the ECTR.

Section 1 of this paper considers some general aspects of the screening of generic chemical structures, and section 2 outlines the strategy for screen generation from the ECTR. A familiarity with the nature of generic structures in patents and an understanding of the organization of the ECTR is assumed from earlier publications.[1,4] Section 3 describes the ECTR tracing algorithm *TREETRACE* and the generation from *Specific* partial structure records of the *composite augmented atom* fragments from which screens are subsequently derived, while section 4 describes the use of TOPOGRAM to generate directly screens from *Generic* partial structure records of the ECTR. A more complete description of TOPOGRAM, and particularly of its treatment of classes of monocyclic and fused ring radicals, is to be reported in a subsequent paper in this series.[7] Section 5 describes the generation of screens from the *composite augmented atoms* generated during the operation of *TREETRACE* and their integration with screens generated by TOPOGRAM, and section 6 describes the bitscreen record and generic structure screening algorithm implemented at Sheffield.

(1) CONSIDERATIONS FOR GENERIC STRUCTURE SCREENING

Traditionally, files of generic chemical structures have been maintained and searched by a manually assigned fragmentation code. As a consequence of the lack of a complete structure record in any form other than its "paper description", mechanized structure searches of generic structures are limited to the matching of the assigned fragment terms and can only be improved upon by scrutiny of the search output. In contrast, substructure search systems that search files of specific chemical substances provide a variety of automated search capabilities. These are made possible by the capture in machine-readable form of a complete and unambiguous description of each structure, in the form of either a chemical name, a line notation or a connection table. With a *topological* connection table record, it is possible to achieve by entirely automatic means a search performance for specific substances that shows both 100% recall and, in principle, 100% retrieval precision.[8]

In order to achieve the latter in specific structure searching, some form of "atom-by-atom" matching is likely to be required to make the fine distinction between closely related structures. However, it is impractical to perform this type of search on every structure in a large file due to the high computational costs associated with this type of algorithm.[9] Accordingly, a prior screen search is invariably conducted with the purpose of minimizing the number of candidate structures for which the atom-by-atom search is necessary.[10] Specifically, the screen search rejects from the structure file those structures that bear little or no resemblance to the query structure and retains for further consideration only those that satisfy certain minimal requirements of the search query. It is not easy to draw the distinction between screen searches and searches of specific and generic structure files that proceed on the basis of an applied fragmentation code. In both cases substructural fragments are used as the basis on which structure comparisons are made. However, *screening* is usually reserved for the case of specific structure search systems, in which the fragment screens are generated algorithmically from the complete structure record, which is itself available for subsequent atom-by-atom searching. Practical and theoretical considerations of the design and implementation of screening systems have been discussed previously.[10-12]

This description of a screening system is entirely appropriate to the present context of searching generic chemical structures, for which the ECTR is capable of supporting algorithmic screen generation, and is amenable, in principle, to a variety of substructure search algorithms. A two-stage screening search followed by an atom-by-atom search of the ECTR was proposed in an earlier paper,[1] although little consideration was given at that time as to how the latter, in particular, might be effected. In the light of practical experience, it has become clear that searches of generic structures at the atom-by-atom level will incur enormous computational expense. Accordingly, a greater emphasis is likely to be placed on the use of a series of screening searches, with the expectation that the *screenout* obtained at the most refined level will be such that the additional expense of an atom-by-atom search will not be justified. To this end, a number of screening algorithms is being investigated, of which that reported here is the first of the screening levels to be implemented in the Sheffield generic chemical structure search system. The description of a more discriminating, higher level screening search based on a graph relaxation algorithm has been submitted for publication[13] and is expected to be implemented and tested shortly within this system.

Our initial implementation of a generic structure screening system is based upon the types of screen used in the CAS ONLINE system for searching the specific substance records in the CAS Registry file.[14,15] These in turn are based on the search screens used at the Basel Information Centre (BASIC) in their substructure search system.[16,17] BASIC developed this system by modifying and extending an earlier CAS substructure search system, utilizing the results of research into the design of screening systems carried out previously at Sheffield in the early 1970s.[11] Unlike the fragment terms of a fragmentation code, which typically represent chemically significant functional groups and ring systems, screens generated algorithmically from a connection table are generally less explicit and smaller in size than the chemically significant groups present in the molecular structure. Consequently, the screens are relatively inexpensive to generate exhaustively but are required to be assigned in sufficient numbers to provide an adequate discrimination between structures. In order to facilitate substructure searches with queries described at various levels of specificity[18] and to reduce the disparate distribution of even these small screens in chemical structure files such that they accord more fully with the theoretical desirability for equifrequent assignment,[10] screens are themselves generated at various levels of specificity and size. For these purposes, the CAS ONLINE screen set[19] includes augmented atom screens, atom sequence, bond sequence, and connectivity sequence screens of various lengths and specificity, and atom and ring counts and simple ring descriptors.

We have selected for investigation an exact subset, both in terms of content and organization, of the CAS ONLINE screen set, specifically augmented atom (AA), atom sequences (AS) of lengths four (AS4), five (AS5), and six (AS6) atoms, and bond sequences (BS) of lengths three (BS3), four (BS4), and five (BS5) bonds. Extension of this set to include atom connectivity sequences (CS) of lengths three (CS3), four (CS4), five (CS5), and six (CS6) atoms is being considered, although the utility of CS screens for screening structures of variable topology may be limited.[15] This selection is not

Table I. Content of Screen Dictionary[a]

species	tot	screen nos.	distribution 1	2	3	4	5	6	7	8	9	≥10
AA	833	445	390	21	8	6	3	5	0	0	0	12
AS4	408	135	14	38	58	1	20	0	0	0	2	2
AS5	572	205	24	63	91	0	23	2	0	0	1	1
AS6	526	176	15	46	82	2	28	0	1	0	2	0
BS3	51	33	17	15	0	1	0	0	0	0	0	0
BS4	134	62	15	29	14	1	3	0	0	0	0	0
BS5	179	67	16	23	15	5	2	4	1	0	0	1

[a] Total number of screens 2703; screen numbers in use 1123.

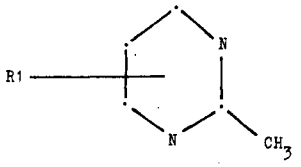


Figure 1. Generic chemical structure. Substituted 2-methylpyrimidines in which the pyrimidine is mono- or disubstituted ortho to a ring nitrogen by C(=O)R_2 or NHC(=O)R_3, in which R_2 is either OCH_3, NH_2, or an alkenyl group of between two and four atoms and R_3 is a 1–6C cycloalkyl group.

suggested as either optimal or complete for the present purpose but provides descriptors that may be generated exhaustively at various levels of specificity and size. The content of the present screen dictionary is summarized in Table I. This specifies for each type of screen in the dictionary the total number of screens (sequence inversions are not counted), the number of screen numbers to which those screens are allocated, and the distribution of those screens among the available screen numbers. This distribution is reported in terms of the number of sets of shared screens that contain a specified number of screens, specifically singleton sets, sets that contain two to nine screens, and sets that contain 10 or more screens.

The reason for choosing in this context screens of this type is not only because of the accumulated experience gained in their use in (albeit specific) structure search systems but also because of the characteristics of generic chemical structures themselves. A screen generation algorithm must perforce take account of the options for substitution represented in the generic structural description, particularly of the variable chemical nature of substituent groups and the often variable substitution patterns in which these groups may occur. A simple generic structure that illustrates these options is shown in Figure 1, for which a corresponding GENSAL description is shown in Figure 2. These characteristics are such that considerable, and most likely insupportable, costs would be incurred in the exhaustive generation and use of screens of structural fragments that are significantly larger than those proposed here. These costs result not only from the computational complexity of generating exhaustively larger screens over extended regions of variable topology and chemical identity but also from the consequent increase in the number of screens generated and the low utility of a great number of these in the screening system. Nevertheless, the use of larger fragments generated from the ECTR by a graph segmentation algorithm to be stored and searched in a relational data structure that mirrors the organization of the ECTR is also being studied at Sheffield. The utility of this type of substructure search algorithm for generic structure searching will be reported.

As illustrated in Figure 1, a generic chemical structure typically encompasses a large or infinite number of distinct specific structures. The screens generated exhaustively from the ECTR by the procedures described here comprise the logical union of the screens for each of these specific structures, although these structures are not enumerated explicitly.

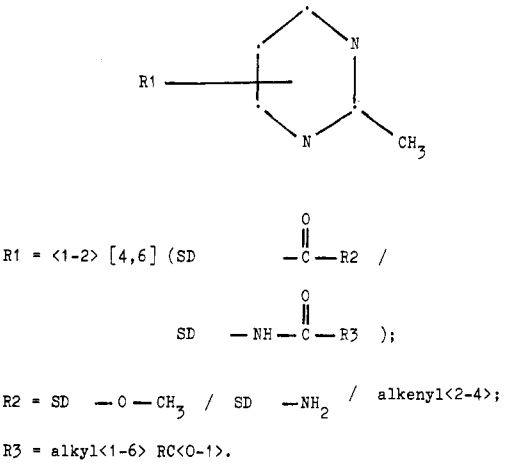


Figure 2. GENSAL description of generic chemical structure.

Screens that are common to each of these notional structures can be considered as representing *essential* fragments of that generic structure, and we refer to these as MUST screens. Screens that are not essential to the generic structure are described as MAY screens. A similar distinction is drawn between MUST and POSSIBLE functional group fragments in the IFI/Plenum system.[20] MUST screens are derived from the invariant part(s) of the generic structure but may also include atoms and bonds from without the invariant part where these are common to each of the alternatives for a variable substituent group attached at fixed position to the invariant part. MAY screens are derived exclusively from the variable parts of the generic structure and include those screens that span the connections between the invariant and the variable parts and that are not otherwise considered as MUST screens.

The MUST screens of a generic chemical structure stand in a logical AND relationship to one another, in the same way as the screens of a specific structure. However, the degeneracy of a bitscreen record in which screens are unrelated by their structural context requires that all MAY screens stand in an OR relationship to one another and to each MUST screen. At higher levels of representation, it becomes possible to describe more explicitly the logical relationships and exclusions that pertain between subsets of the screens and to apply these during search. Reflection of this information in the simplest bitscreen record is necessarily limited, and the consequences of this for generic structure screening are addressed in section 6.

(2) GENERAL STRATEGY FOR SCREEN GENERATION

Algorithmic generation of topological screens is facile in the case of specific chemical structures, involving no more than path tracing within a single connection table. In the case of generic structures, the situation is more complicated, and requires not only the generation of screens from within each partial structure (PS) of the ECTR but also the generation of those screens that span the permitted connections between PSs. Screens are generated exhaustively from the ECTR by a depth-first search algorithm *TREETRACE*. This algorithm accesses every PS in the ECTR and calls the screen generation routines that are appropriate to that PS. Figure 3 is a schematic description of the ECTR generated by the GENSAL Interpreter[6] from the GENSAL structure description shown in Figure 2. For clarity in the following discussion, a unique alphabetic identifier is associated with each PS, and each explicit atom is numbered.

Partial structures are stored in the ECTR in one of several different forms.[4] For the purpose of screen generation, only *Specific* and *Generic* PSs are considered. *Unknown* and *Other*

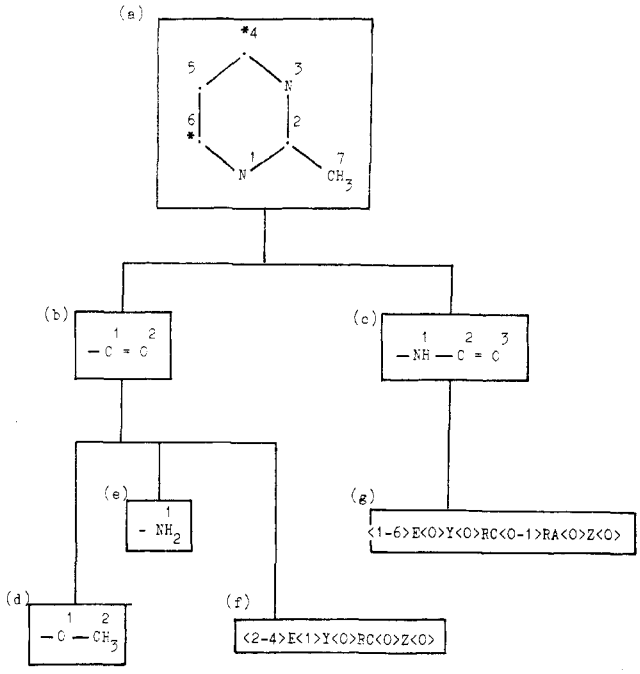


Figure 3. Schematic description of ECTR.

PSs represent substituent groups about which no structural information is given or can be inferred and are not amenable to screen generation. *Specific* PSs are present in the ECTR in the form of a partial connection table and represent the structure diagrams, linear formulas, and specific nomenclatural terms of the GENSAL input. *Generic* PSs provide a less explicit representation of radical class substituents that are expressed in GENSAL as generic nomenclatural terms and are present in the ECTR in the form of a set of structural parameters,[3] which collectively describe the radical class for the purpose of processing by TOPOGRAM. An atom in any *Specific* PS(p) is described as being *externally* connected if any one or more of the atoms to which it is or may be connected (*congeners*) belongs to another PS(q) ($p \neq q$). These atoms are described as *external* atoms with respect to PS(p). Connections to atoms within a PS are described as *internal*. In Figure 3, atoms [a4] and [a6] are externally connected (indicated here by "*"), and atoms [b1] and [c1] are *external* atoms with respect to PS(a). Partial structures are logically connected in the ECTR by means of *gates*. These are represented schematically in Figure 3 by the links between PSs. For a given PS, each *external* connection is recorded, together with supplementary information relating to positions of attachment and multiplicity of occurrence, in the appropriate *gate*. Thus, the single connection between PS(a) and each of its alternative substituents PS(b) and PS(c) is recorded in the *childgate* associated with PS(a). This same connection is redundantly recorded in the *parentgate* associated with each of the respective *child* PSs.[4]

Specific PSs are processed for screen generation in a different manner from *Generic* PSs, in accordance with the form in which they are present in the ECTR. For each *Specific* PS accessed by *TREETRACE*, a set of *composite augmented atom* (*caa*) fragments is generated. Each *caa* describes an atom of the PS and the identity of each of its possible *congeners*, including those that are *external* to that PS. The set of *caa*'s constitutes for each *Specific* PS an intermediate representation between the ECTR and the search screens, from which the latter are derived at a later stage in the processing. In the case of a *Generic* PS, *TREETRACE* passes control to TOPOGRAM, which generates directly the AA, AS, and BS screens that are characteristic of the radical class represented by that PS.

Table II. Bond Codes and Specificity Levels for Supplementary Screens

specificity level	bond code	bond description
	0	not specified
general	1	any bond
intermediate	2	any chain bond
	3	any ring bond
	4	any single bond
	5	any double bond
	6	any triple bond
specific	7	chain single bond
	8	chain double bond
	9	chain triple bond
	10	chain tautomeric bond
	11	ring single bond
	12	ring double bond
	13	ring triple bond
	14	ring delocalized bond
	15	ring tautomeric bond

After processing of the ECTR by *TREETRACE* is complete, AA screens are derived from the sets of *caa*'s generated from *Specific* PSs of the ECTR. Where a *caa* contains a bond between an externally connected atom and an *external* atom, AA screens that are derived from that *caa* and contain that bond describe the immediate environment around the point of connection between the respective PSs. AS sequence screens are generated by concatenating pairs of *caa*'s across a common bond and enumerating from each *concatenated pair* the AS4 screens to which it gives rise. AS5 and AS6 screens are generated from each AS4 sequence by further concatenation of the appropriate *caa* with the *terminus* atom at each end of the sequence. AS screens generated in this way describe the extended environments around the points of connection between two or more *Specific* PSs if the *caa*'s from which the sequences are enumerated contain bonds that span these connections. AA and AS screens that span the connections between a *Specific* PS and a *Generic* PS are generated by interfacing the *caa* from the *Specific* PS with the screens generated by TOPOGRAM for the *Generic* PS.

Screens are generated at the level of atom and bond specificity at which they are represented in the ECTR. As noted in the previous section, it is desirable to include in the search record screens at lower levels of specificity. Accordingly, the screens generated from the ECTR are enriched by the addition of *supplementary* screens, each of which is derived directly from a screen of higher specificity. Supplementary screens are generated where appropriate by reducing the level of bond specificity and by generalizing atom types. In the first case, use is made of the hierarchy of bond types employed in the present implementation, as shown in Table II. This hierarchy subdivides into several groups. For the purpose of generating supplementary screens, the groups designated *Specific*, *Intermediate*, and *General* are used. Thus, a single chain bond (7) at the *Specific* level of description may be generalized to a chain bond (2) at the *Intermediate* level and to any bond (1) at the *General* level of description. A condition is imposed on the generation of supplementary screens such that those generated by reduction of bond specificity should contain only bonds from a single level of description. Bond sequence BS3, BS4, and BS5 screens are generated from the corresponding AS screens (AS4, AS5, and AS6, respectively) by generalizing each atom to the reserved atom-type "A" (any atom), with the result that only bonding information remains in each screen.

(3) ECTR TRACING ALGORITHM: TREETRACE

TREETRACE is a recursive, depth-first search algorithm that systematically accesses each PS of the ECTR with the

```
begin
repeat
  if CHILD.recursive
    then begin
           ALTCHILD <---- CHILD.FIRST.ALTERNATIVE
           repeat
             TREETRACE(ALTCHILD.COMBINATION)
             ALTCHILD <---- ALTCHILD.next
           until ALTCHILD.end
         end
    else if not_already_processed
           then PSPROCESS(CHILD.PS);
    CHILD <---- CHILD.next
until CHILD.end
end.
```

Figure 4. ECTR tracing algorithm, *TREETRACE*. The variable CHILD is passed as a value parameter on each recursion of *TREETRACE*. CHILD is initialized to the unique Internalrep. Constantpart.Childgate.

```
caa: N [a1]   12 C [a2]   11 C [a6]
caa: C [a2]   12 N [a1]   11 N [a3]    7 C [a7]
caa: N [a3]   11 C [a2]   12 C [a4]
caa: C [a4]   12 N [a3]   11 C [a5]   { 7 C [b1]
                                      { 7 N [c2]
caa: C [a5]   11 C [a4]   12 C [a6]
caa: C [a6]   12 C [a5]   11 N [a1]   { 7 C [b1]
                                      { 7 N [c2]
```

Figure 5. Composite augmented atoms from PS(a).

purpose of screen generation. On accessing a PS, *TREETRACE* calls the procedure *PSPROCESS*, which, in the case of a *Specific* PS, causes the set of *caa*'s to be generated from the partial connection table and, in the case of a *Generic* PS, causes TOPOGRAM to be invoked. The operation of *PSPROCESS* in the first case is described below after the description of the *TREETRACE* algorithm, and that in the second case is described in section 4.

TREETRACE begins by calling *PSPROCESS* for the PS of the invariant part of the generic structure. Once processing of a PS is complete, *TREETRACE* proceeds by recursion down through the *childgate* (if any) of that PS, thereby carrying the processing in turn to each of the PSs that occupy the next subordinate level of the ECTR. In certain cases, a *childgate* leads directly to a combination of further *childgates*, rather than immediately to one or more PSs.[4] In this case, *TREETRACE* continues its recursion in turn through each *childgate* in the *combination bar* and does not call *PSPROCESS* until a further PS is encountered. *TREETRACE* proceeds in this manner until a PS is encountered that, after being processed by *PSPROCESS*, fails to cause further descent. At this point, *TREETRACE* ascends to the *parent* PS of which the currently processed PS is a *child*. Further recursion is immediately possible if the *childgate* provides access to one or more subordinate PSs or to *childgate* combinations through a *combination bar*. If no further descent is possible, *TREETRACE* ascends directly. The tracing of the ECTR continues until *TREETRACE* ascends to the PS that represents the invariant part of the generic structure, with which *TREETRACE* was initiated. At this point, each PS in the ECTR has been accessed and processed by *PSPROCESS*, and *TREETRACE* terminates. The *TREETRACE* algorithm is stated in Figure 4.

PSPROCESS processes a *Specific* PS by generating a *caa* from each row of the partial connection table. The *caa* describes the *focal* atom, each of the *internal* atoms to which it is connected, and each of the *external* atoms to which it may be connected. The *caa*'s generated from PS(a) of Figure 3 are shown in Figure 5. Where the *focal* atom is externally connected, as in the cases of atoms [a4] and [a6], the identity of each *external* atom is determined by accessing the respective

```
caa: C [b1]   8 O [b2]   { 7 C [a4]   { 7 O [d1]
                         { 7 C [a6]   { 7 N [e1]
                                      { 7 Z [f0]
caa: O [b2]   8 C [b1]
```

Figure 6. Composite augmented atoms from PS(b).

Table III. Structural Parameters for Generic Radical Class Description

parameter identifier	description
C	carbon atom count
T	no. of acyclic ternary branching atoms
Q	no. of acyclic quaternary branching atoms
E	no. of localized olefinic unsaturations
Y	no. of localized acetylenic unsaturations
RC	no. of rings
RN	no. of ring atoms
RS	no. of substituted ring atoms
RF	no. of ring fusion atoms
RA	no. of delocalized rings
Z	no. of heteroatoms

PS via the appropriate *childgate* or *parentgate*, thereby spanning all permitted connections between the respective PSs. For this purpose, the *position sets*,[2] which specify the fixed or variable positions of attachment in each PS, are utilized from the appropriate *gates*. In the case where a connected PS describes a hydrogen atom only, for example, when hydrogen is specified singly or as one of a group of alternative substituents, the connected atom is not recorded in the *caa*, as hydrogen atoms are not represented in the search screens. Where a connected PS is of type *Unknown* or *Other*,[4] the connected atom is recorded in the *caa* by the generalized atom-type "A". Where the connected PS is of type *Generic*, a special symbol "Z" is recorded in the *caa*, which is subsequently replaced by the appropriate atoms from screens generated from the *Generic* PS by TOPOGRAM. The *caa*'s for PS(b) are shown in Figure 6. *PSPROCESS* generates *caa*'s in this manner from every *Specific* PS accessed by *TREETRACE*. In the case of a *Generic* PS, for which a connection table representation is not available, screens are generated directly by TOPOGRAM, as described in the following section.

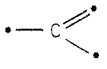
(4) SCREEN GENERATION FROM GENERIC PS

Generic substituents, for example, homologous radical series such as alkyl, alkenyl, and cycloalkyl, are present in the ECTR as *Generic* PSs in the form of an *intensional* description of the radical class. An intensional description comprises the set of properties that is peculiar to the members of that class. In the present case, the properties of interest are the substructural features that characterize the radicals of a particular class. For example, alkenyl radicals are characterized by the presence in each radical of (at least) one olefinic unsaturation. Similarly, cycloalkyl radicals are characterized by the required presence in each radical of (at least) one saturated carbocycle. In certain cases, a class is more appropriately defined in terms of its *exclusion*, that is, by the absence of characteristic features. For example, alkyl radicals are recognized as such primarily by the absence of cycles, unsaturations, and heteroatoms. A combination of intensional and exclusive descriptions is most commonly used to define a class, if only intuitively. For example, an equivalent but more explicit characterization of the class of cycloalkyl radicals requires the presence in each radical of (at least) one cycle, together with the absence of unsaturations and heteroatoms. A relatively small number of structural features have been used in this way to describe a wide variety of generic radical classes.[3] The structural features and their *parameter identifiers* that are

presently used to provide a *class representation* for generic radical classes are listed in Table III, and the class representations of "2–4C alkenyl" and "1–6C cycloalkyl" are shown in PS(f) and PS(g), respectively, of Figure 3.

It is clearly not practicable to generate exhaustively specific radicals from these classes and subsequently to generate screens from each radical. Only by enumerating the entire class, if this were possible, could exhaustive screen generation be ensured. Even for simple classes of acyclic radicals this approach is both expensive and extremely inefficient. The class "1–10C alkyl", while encompassing 879 unique radicals, is characterized by a total of only four augmented atom fragments. Generation of screens that are characteristic of generic radical classes is achieved by a two-stage algorithm, which proceeds by the controlled assignment of the replacement rules of TOPOGRAM.[3] Each member of the *terminal* vocabulary of TOPOGRAM symbolizes an atom and its bonding pattern. Single-rule assignments therefore permit the generation of AA screens (strictly, bonded atoms[21]), in which all attached atoms are assumed initially to be carbon. The generation of AS and BS screens involves multiple rule assignments, where for each assignment sequence the structural features cumulated over the generation of that sequence remain consistent with the class representation. It should be noted that screen generation from generic radical classes is achieved with TOPOGRAM without the need to generate a single radical instanced by the class. In as far as TOPOGRAM is sufficient to describe that class (and there are classes for which TOPOGRAM is not yet able to provide an adequate description, for example, any class whose members include perifused ring systems), the resulting set of screens contains descriptors that are characteristic of the full variety of radicals defined by that class.

TOPOGRAM distinguishes between apical and nonapical rule assignments during the generation of AA screens. Apical assignments result in screens whose *focal* atom is externally connected, that is, has at least one connection to another PS. These screens describe the environments at the point of connection of the radical class substituent and its *parent* PS. (The generation of screens from PSs that are accessed through the *childgate* of a *Generic* PS is not yet possible.) Nonapical assignments generate screens that are characteristic of the possible nonapical environments of a generic radical class. A number of conditions must be satisfied for both apical and nonapical rule assignments. The assignment of any rule commits one or more structural features to the AA screen it generates. For example, the TOPOGRAM assignment rule A ----> 8BA generates the atom-centered fragment



and commits a ternary branch, a double bond, and a total of four atoms, of which three are *virtual* atoms; one of these is *external* if the screen describes an apical environment. Similarly, the ring-fusion rule G ----> d<G>H generates the atom-centered fragment

in which the symbol "*" signifies a delocalized ring bond, and thereby commits two fused delocalized rings and a total of four ring atoms of which three are *virtual*. Generic radical classes whose class representation forbids the presence of such features cannot be characterized by these screens. Accordingly, assignment rules that give rise to noncharacteristic screens are disabled by prior analysis of the class representation. Similarly, generation of screens that contain features in excess of the permitted range specified in the class representation is prevented.

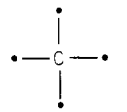
```
AA:   C      7 C
AA:   C      8 C
AA:   C      7 C    7 C
AA:   C[fO]  7 Y    7 C
AA:   C      7 C    8 C
AA:   C[fO]  7 Y    8 C
AA:   C      7 C    7 C    7 C
AA:   C[fO]  7 Y    7 C    7 C
AA:   C[fO]  7 Y    7 C    8 C
```

Figure 7. AA screens from PS(f), "2–4C alkenyl".

```
AA:   C      7 C
AA:   C     11 C
AA:   C[gO]  7 Y
AA:   C      7 C    7 C
AA:   C     11 C   11 C
AA:   C[gO]  7 Y    7 C
AA:   C      7 C    7 C    7 C
AA:   C      7 C   11 C   11 C
AA:   C[gO]  7 Y    7 C    7 C
AA:   C[gO]  7 Y   11 C   11 C
AA:   C      7 C    7 C    7 C    7 C
AA:   C      7 C    7 C   11 C   11 C
AA:   C[gO]  7 Y    7 C    7 C    7 C
```

Figure 8. AA screens from PS(g), "1–6C cycloalkyl".

If the number of atoms committed by an assignment rule equals the maximum permitted by the class representation, then that rule can only be assigned if it also satisfies all of the structural features required by the class representation. The rule A ----> 0AAA generates the atom-centered fragment



and commits a saturated quaternary atom and four *virtual* atoms, of which one is *external* in the case of an apical assignment. This rule is disabled for both apical and nonapical assignment in the case of AA screen generation for the radical class "2–4C alkenyl", as the rule fails to assign the double bond required by the class representation <2–4>E<1>Y<0>RC-<0>Z<0>.

Apical assignments are subject to further conditions in addition to those required for nonapical assignments. Clearly, the external connection generated by an apical rule assignment must be identical with that specified in the *parentgate* of the *Generic* PS. Thus, the rule B ----> 9AA, which assigns an unsaturated ternary atom in which the external connection may be made only through the double bond, cannot be applied as an apical assignment if the *parentgate* does not specify an olefinic connecting bond. Similarly, apical assignment of terminating rules, for example, A ----> *x*, which assign one atom only, is prevented if the lower bound of the atom-count parameter in the class representation exceeds 1. The AA screens generated for PS(f) and PS(g) are illustrated in Figures 7 and 8, respectively. The *external* atom to which the *focal* atom of each apical screen is connected is represented by the reserved atom-type "Y".

Sequence screens are generated from generic radical classes by an iterative algorithm, which builds atom sequences by the repeated assignment of TOPOGRAM rules and terminates once sequences of the required length have been constructed. The algorithm is initiated with *seeds* derived from the AA screens previously generated for that class. Each seed consists of a two-atom sequence, comprising the *focal* atom of an AA screen and one of its connected atoms. Where this connected atom is of atom-type "Y", the sequence screens constructed from that seed describe possible apical environments of the generic radical class. The TOPOGRAM rules are subject to the same assignment conditions as those described above for AA screen generation. However, since the seeds for sequence generation are derived from the AA screens, apical assignment conditions are no longer required. In order that these conditions can be evaluated prior to each rule assignment, and

```
AS4:  C    7 C    7 C    8 C
AS4:  C    7 C    8 C    7 C
AS4:  Y    7 C[f0] 7 C    7 C
AS4:  Y    7 C[f0] 7 C    8 C
AS4:  Y    7 C[f0] 8 C    7 C

AS5:  Y    7 C[f0] 7 C    7 C    7 C
AS5:  Y    7 C[f0] 7 C    8 C    7 C
AS5:  Y    7 C[f0] 8 C    7 C    7 C
```

Figure 9. AS screens from PS(f), "2–4C alkenyl".

```
caa:  C [b1]   8 O [b2]  { 7 C [a4]   { 7 O [d1]
                         { 7 C [a6]   { 7 N [e1]
                                      { 7 C [f0]

caa:  O [b2]   8 C [b1]
```

Figure 10. Fully specified composite augmented atoms from PS(b).

```
AA:  C    7 C
AA:  C    8 C
AA:  C    7 C    7 C
AA:  C    7 C    8 C
AA:  C    7 C    7 C    7 C
AA:  C    7 C    7 C    8 C
```

Figure 11. Fully specified AA screens from PS(f), "2–4C alkenyl".

thereby prevent prospectively the assignment of rules that lead to screens that are not characteristic of the radical class, each sequence carries with it during its construction a cumulated count of the structural features committed by previous assignments. Where the assignment of a rule would lead to a violation of the class representation, then assignment of that rule is prevented. The sequence screens for PS(f) are shown in Figure 9. The absence of AS6 screens is a consequence of the maximum of four atoms permitted for the class "2–4C alkenyl".

(5) SCREEN GENERATION FROM SPECIFIC PS

AA, AS, and BS screens from *Specific* PSs of the ECTR and those that span the connections between *Specific* PSs and between a *Specific* PS and a *Generic* PS are derived directly from the set of *caa*'s generated by *TREETRACE*. Prior to this, individual *caa*'s require the identities of *external* atoms derived from *Generic* PSs to be specified. For each *caa*, these identities are established by accessing the apical AA screens generated by TOPOGRAM for each appropriate *Generic* PS and determining from these the apical atom types for that generic radical class. This set of atom types replaces the *external* atom previously recorded in the *caa* by the reserved symbol "Z". At the same time, the *external* atom of each apical screen of the *Generic* PS is specified by replacing the reserved symbol "Y" by the identity of the *focal* atom of that *caa*. Additional identities may be added subsequently in the case where this same *Generic* PS screen set is accessed during similar processing of remaining *caa*'s. Figure 10 illustrates the *caa*'s of PS(b) in which the *external* atom belonging to PS(f) has been specified. The identities of *external* and externally connected atoms are shown where appropriate. The designation (f0) distinguishes the possible apical atoms of the *Generic* PS(f). Figure 11 illustrates the fully specified AA screens of PS(f). Note that because the *external* atom of each apical screen is identified from PS(b) as a carbon atom, certain of the apical screens become indistinguishable from nonapical screens.

A *caa* represents a single AA screen only in the case where every congener is *internal*. When one or more congeners are *external*, the *caa* encompasses a number of AA screens, and each of these is enumerated from the *caa* by a recursive algorithm that generates all possible combinations of *internal* and *external* congeners and that recurses in cases where two or more of the *external* congeners have two or more possible identities. In addition to this, a second recursive algorithm generates from each AA screen all the derivative AA screens

```
AA:  C    8 O    7 C    7 O
AA:  C    8 O    7 C    7 N
AA:  C    8 O    7 C    7 C
AA:  C    8 O    7 C
AA:  C    8 O    7 O
AA:  C    8 O    7 N
AA:  C    8 O

AA:  C    7 C    7 O
AA:  C    7 C
AA:  C    7 O
AA:  C    7 C    7 N
AA:  C    7 N
AA:  C    7 C    7 C

AA:  O    8 C
```

Figure 12. AA screens from PS(b).

```
caa:  N[a3]   11 C[a2]   12 C[a4]
+
caa:  C[a4]   12 N[a3]   11 C[a5]   { 7 C[b1]
                                     { 7 N[c2]

C[a2] 11 }                          { 11 C[a5]
         } N[a3]  12 C[a4]          {  7 C[b1]
                                    {  7 N[c2]

AS4:  C[a2]  11 N[a3]  12 C[a4]  11 C[a5]
AS4:  C[a2]  11 N[a3]  12 C[a4]   7 C[b1]
AS4:  C[a2]  11 N[a3]  12 C[a4]   7 N[c2]
```

Figure 13. AS4 screens arising from the concatenation of two composite augmented atoms of PS(a).

that can be obtained by successive omission of congener atoms. The AA screens generated by these procedures from the *caa*'s of PS(b) are illustrated in Figure 12.

Extensive path tracing within the ECTR is expensive. Consequently, sequence screens are generated by an algorithm that operates on the sets of *caa*'s and the screens generated from *Generic* PSs by TOPOGRAM, rather than directly on the ECTR itself. This algorithm generates AS screens from within each *Specific* PS, as well as those that span the possible connections between two or more *Specific* PSs and between a *Specific* PS and a *Generic* PS. (Generation of screens that span the connections between two or more *Generic* PSs is not yet possible.) For those screens within a *Specific* PS and those that span two or more *Specific* PSs, the algorithm constructs AS screens by a process of pairwise concatenation of *caa*'s. The algorithm is applied to each *caa* generated by *TREETRACE* and for each bond in the *caa* constructs a (notional) *concatenated pair* from which AS4 screens are enumerated directly and from which, in turn, AS5 and AS6 screens are generated by further concatenations. Each *caa* retains for this purpose the identity of each of its constituent atoms, both *internal* and *external*, in the manner illustrated in Figures 5 and 6. Concatenation of two *caa*'s is prevented where the symmetric concatenation has previously been made and in the case where either of the two atoms joined by the bond across which the concatenation is to be made is univalent or is otherwise prohibited from carrying additional connections. In this case, sequence screens of sufficient length cannot be generated. In the case of AS screens that span a connection between a *Specific* PS and a *Generic* PS, the algorithm interfaces the appropriate *caa* with the apical screens of the *Generic* PS generated by TOPOGRAM.

In the former case, AS4 screens are enumerated from each *concatenated pair* by adding connected atoms to each of the two atoms joined by the bond across which the concatenation was made, such that each AS4 screen has that bond at its center. For each successive addition to one atom of this pair, each of the connected atoms of the second atom is successively

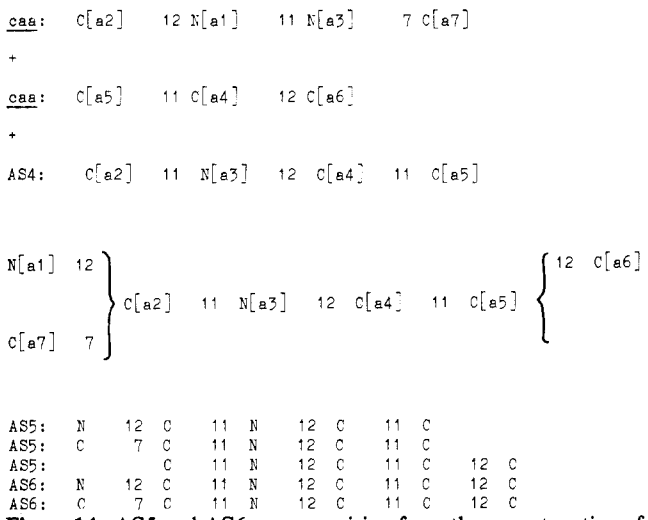


Figure 14. AS5 and AS6 screens arising from the concatenation of AS4 screens and composite augmented atoms of PS(a).

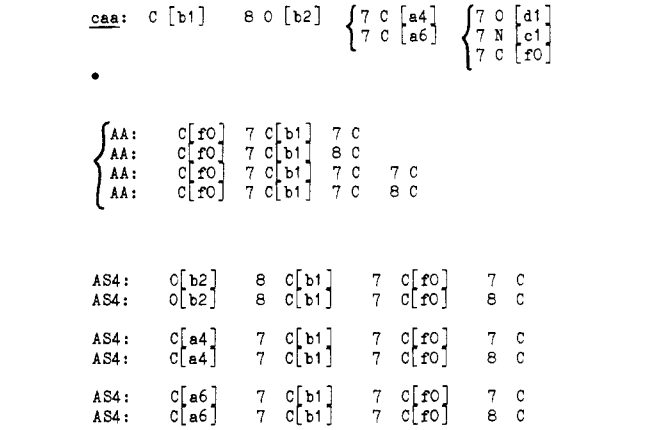
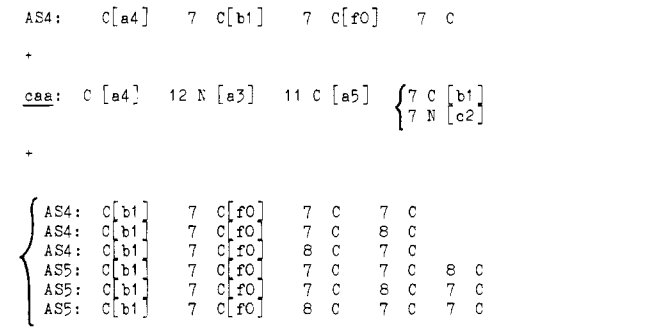


Figure 15. AS4 screens arising from interfacing a composite augmented atom of PS(b) with the apical screens of PS(f).

added. Figure 13 illustrates the concatenation of two *caa*'s from PS(a) and the resulting AS4 screens. The concatenation operator is depicted by the symbol "+", and atom identities are shown where appropriate. AS5 and AS6 screens are subsequently generated from each AS4 screen by further concatenation of the appropriate *caa* to each *terminus* atom of the AS4 screen, as illustrated in Figure 14 for the first of the AS4 screens shown in Figure 13. In practice, the number of AS screens generated in this way may be fewer than the number expected from every possible combination of the connected atoms recorded in the *caa*'s. This is so in cases where two or more of these atoms are derived from different PSs, which, by virtue of any GENSAL *condition* statements that limit the permitted combinations of these PSs, are not permitted to occur together. Conditional information of this type is not yet recorded in the ECTR and cannot yet be accounted for in screen generation. However, as noted earlier, the nature of a screening record based only on degenerate screens is such that this information need not, indeed cannot, be utilized in a first-level screening search.

In the case in which the bond under consideration represents the connection between an atom in a *Specific* PS and the apical atom of a *Generic* PS, AS4 screens are generated not by concatenation but by the successive addition to the first atom of the connected atoms in the *caa* and by interfacing to the second atom atoms derived from the apical screens of the *Generic* PS. Figure 15 illustrates the interfacing in this way of the first of the two *caa*'s of PS(b) (Figure 10) with the apical AA screens of PS(f) and lists the AS4 screens that are generated across the bond that connects these PSs. AS5 and AS6



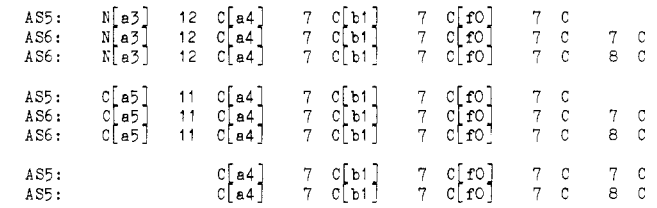


Figure 16. AS5 and AS6 screens arising from concatenation of a composite augmented atom to the terminus of an AS4 screen and interfacing the other terminus with apical AS screens of PS(f).

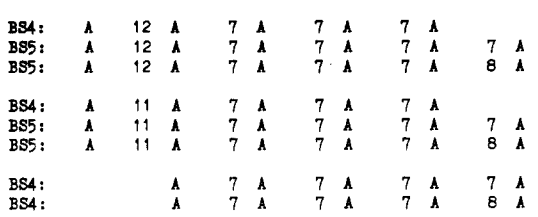


Figure 17. BS screens derived from AS screens of Figure 16.

screens, which span this same bond, are generated from the AS4 screens by replacing the generic *terminus* atom with a sequence of atoms of appropriate length from certain of the apical AS screens of the *Generic* PS and the further concatenation to the specific *terminus* of the *caa* of each of its connected atoms. Figure 16 illustrates the generation of AS5 and AS6 screens in this way from the third of the AS4 screens shown in Figure 15. (The first two AS4 screens in Figure 15 can give rise to only AS5 screens and are not suitable for this illustration.) Where *both* terminus atoms of the AS4 screen belong to *Generic* PSs, AS5 and AS6 screens are generated by replacing each terminus atom by a sequence of atoms from the apical AS screens of the corresponding *Generic* PSs.

Bond sequence screens BS3, BS4, and BS5 are derived directly from AS4, AS5, and AS6 screens, respectively, and are generated by suppressing the identity of each atom in the AS screen. Figure 17 illustrates the BS4 and BS5 screens derived from the AS5 and AS6 screens listed in Figure 16.

There is a number of characteristics of generic chemical structures for which the screen generation procedures are not yet comprehensive. The variety of generic nomenclatural terms that may be processed accurately by TOPOGRAM is still small, and the TOPOGRAM screen generation algorithm in these cases still requires validation. Extension of the rule base of TOPOGRAM is necessary, specifically to include context-sensitive rules capable of describing explicitly structural features that are larger than those represented at present. These extensions to TOPOGRAM will enable a wider variety of radical classes to be represented, and additional *parameter identifiers* will be defined as appropriate. The treatment of GENSAL multiplier variables is not yet comprehensive in respect of screen generation, nor are screens yet generated to describe environments defined by certain types of substituent combination, for example, the additional fused ring defined in the GENSAL sentence in Figure 18. Improved treatment

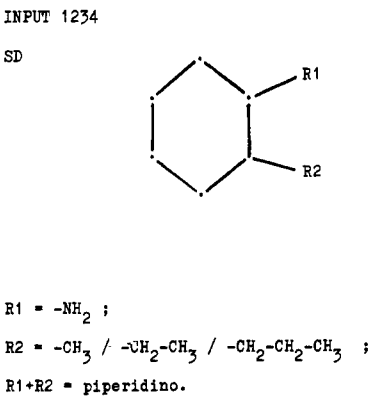


Figure 18. GENSAL substituent combination.

of these and other instances is being introduced into the screen generation procedures on a continuing basis.

(6) BITSCREEN RECORD FOR GENERIC STRUCTURE SCREENING

A screening record for generic database structures and for specific, generic, and substructure query structures has been implemented in the form of a *twin-bitstring* and is described below. The screening record is based on the degenerate screens generated from the ECTR by the procedures outlined in this paper. Each bitstring of the screening record is assigned by matching against the screen dictionary (Table I) the screens generated from the ECTR. If a screen is present in the dictionary, the corresponding bit in the bitstring is set. If it is absent, the screen is ignored.

It is argued in this section that elimination of generic database structures can be made at the level of degenerate screens only on the basis of the MUST screens of the query and that query MAY screens play an essential but minor role in the screening search. The distinction between MUST and MAY screens is imperative and ensures against erroneous screenout and the consequent loss in recall. Accordingly, a simple bitstring, in which each bit records the presence or absence in the structure of one or more (superimposed) screens, is inadequate for this purpose, as the bitstring cannot also provide the necessary distinction between MUST and MAY screens.

It would be convenient to record the presence of MUST screens in one bitstring of the twin-bitstring record and of MAY screens in the second bitstring. However, further consideration reveals that elimination of a generic database structure S is possible only if the specific, generic, or substructure query Q contains one or more MUST screens that are present among *neither* the MUST screens nor the MAY screens of S. In other words, the essential parts of Q must be represented in either or both of the invariant and variable parts of structure S for S to be considered as a candidate for retrieval. Accordingly, the second bitstring in the screening record of S is more suitably implemented as the union of MUST and MAY screens, $S_{MUST} \cup S_{MAY}$.

The simplest bitstring matching algorithm can be stated as

$$\text{if } (Q_{MUST} \subseteq S_{MUST} \cup S_{MAY})$$

$$\text{then } S \text{ is a candidate for retrieval}$$

where Q_{MUST} is the bitstring in the screening record of Q in which the presence of only MUST screens is recorded, and $S_{MUST} \cup S_{MAY}$ is the bitstring in the screening record of S in which the presence of both MUST and MAY screens is recorded. This screening algorithm is valid for specific, generic, and substructure queries, where a generic query is understood to contain one or more variable parts, of which each is defined in terms of a number of structural entities from which screens can be generated exhaustively, and a substructural query is understood to contain wholly undefined parts for which screens cannot be generated or assigned.

The expected screenout of this screening algorithm is seen to depend upon the number and efficacy of MUST screens in the query and the "bit density" of the $S_{MUST} \cup S_{MAY}$ bitstring of each database structure S. Specifically, screenout is likely to be poor where the number of MUST screens in the query is small, as might be the case for highly variable substructure queries and for generic queries in which the invariant part(s) is (are) only small. Where the invariant part is of a trivial nature, the efficacy of the MUST screens for the query is likely to be low, and consequently, the screenout will be reduced. Furthermore, the effectiveness of this screening algorithm will be reduced in the case of a highly generic database structure, for which the number of MAY screens is likely to be large according to the nature of the variable parts of the structure, and the "bit density" of the $S_{MUST} \cup S_{MAY}$ bitstring will be high. The greatest screenout is likely in the case of a specific query structure from which a large number of highly discriminating screens are generated and assigned.

Screening performance for specific and generic queries is likely to be improved upon by taking further account of the nature of generic structure comparisons and the retrieval criteria considered to be appropriate for a generic structure search system. In the case of a specific query structure, a generic database structure S should only be retrieved if one of the specific structures encompassed by S matches exactly the query structure. As the invariant part of S is common to each of these structures, it must also be common to the query structure Q. In other words, every MUST screen of S must find a correspondence with a MUST screen of Q (there can be no MAY screens for a specific query structure, $Q_{MAY} = 0$). Consequently, in addition to the bitstring match

$$Q_{MUST} \subseteq S_{MUST} \cup S_{MAY}$$

the bitstring match

$$S_{MUST} \subseteq Q_{MUST}$$

must also hold if S is to be retrieved in a search with a specific query. Therefore, for a specific query search an improved screening algorithm can be stated as

$$\text{if } (Q_{MUST} \subseteq S_{MUST} \cup S_{MAY}) \text{ and } (S_{MUST} \subseteq Q_{MUST})$$

$$\text{then } S \text{ is a candidate for retrieval}$$

For a generic query, retrieval of a generic database structure S is considered to be acceptable if at least one of the specific structures encompassed by Q is also common to S. If this is so, the invariant part(s) of S must be common to a subset of the structures encompassed by Q. Equivalently, every MUST screen of S must find a correspondence among the MUST *or* the MAY screens of Q. Consequently, for a generic query an improved screening algorithm can be stated as

$$\text{if } (Q_{MUST} \subseteq S_{MUST} \cup S_{MAY}) \text{ and } (S_{MUST} \subseteq Q_{MUST} \cup Q_{MAY})$$

$$\text{then } S \text{ is a candidate for retrieval}$$

As in the case of a generic database structure, the second bitstring in the screening record for a generic query is used to record the union of MUST and MAY screens, $Q_{MUST} \cup Q_{MAY}$. The improved screening algorithms for specific and generic queries can be implemented in a unified manner, on account of the fact that the second bitstring $Q_{MUST} \cup Q_{MAY}$ in the screening record of a specific query Q is indistinguishable from the bitstring Q_{MUST} (where $Q_{MAY} = 0$).

It is not yet clear whether similar improvement can be made at the level of degenerate screens to the screening of substructure queries. Here, the most general criterion for retrieval is that the substructure be contained within the generic da-

tabase structure, irrespective of whether it lies in part or wholly within variable parts of the structure. For a substructure query Q, the number and variety of MAY screens cannot be determined. Consequently, in order to use in this case the improved algorithm described above and to ensure against loss in recall, $Q_{MUST} \cup Q_{MAY}$ would have to be assigned such that every bit was set ("bit density" = 100%). The bitstring match $S_{MUST} \subseteq Q_{MUST} \cup Q_{MAY}$, therefore, becomes valueless in the case of a substructure query.

The two degenerate screening algorithms described here have been implemented, and detailed investigation of screening performance will begin shortly when a test database of generic structures from chemical patents becomes available for experimental purposes. The results of these investigations together with details of the implementation of higher level generic structure screening and substructure search algorithms will be reported separately.

ACKNOWLEDGMENT

We are indebted to Dr. J. Silk, whose advice has contributed to the formulation of the search strategies described in this paper, and Dr. M. Elder and Dr. P. Willett for helpful discussions. We also acknowledge the British Library Research and Development Department for provision of research funding during the period October 1981 to October 1983.

REFERENCES AND NOTES

(1) 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.
(2) 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.
(3) 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.
(4) Barnard, J. M.; Lynch, M. F.; Welford, S. M. "Computer Storage and Retrieval of Generic Structures in Chemical Patents. 4. An Extended Connection Table Representation for Generic Structures". *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 160-164.
(5) Welford, S. M.; Lynch, M. F.; Barnard, J. M. "Towards Simplified Access to Chemical Structure Information in the Patent Literature". *J. Inf. Sci.* **1983**, *6*, 3-10.
(6) 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.*, following paper in this issue.
(7) Welford, S. M.; Lynch, M. F.; Barnard, J. M. "Computer Storage and Retrieval of Generic Chemical Structures in Patents. 7. TOPOGRAM: A Topological Grammar for the Representation, Generation and Recognition of Chemical Radical Classes". In preparation.
(8) Lynch, M. F.; Harrison, J. M.; Town, W. G.; Ash, J. E. "Computer Handling of Chemical Structure Information"; Macdonald: London, 1971.
(9) O'Korn, L. J. "Algorithms in the Computer Handling of Chemical Information". *ACS Symp. Ser.* **1977**, *46*, 122-148.
(10) Lynch, M. F. In "Chemical Information Systems"; Ash, J. E.; Hyde, E., Eds.; Ellis Horwood: Chichester, England, 1975; Chapter 12.
(11) Adamson, G. W.; Cowell, J.; Lynch, M. F.; McLure, A. H. W.; Town, W. G.; Yapp, A. M. "Strategic Considerations in the Design of a Screening System for Substructure Searches of Chemical Structure Files". *J. Chem. Doc.* **1973**, *13*, 153-157.
(12) Feldman, A. J.; Hodes, L. "An Efficient Design for Chemical Structure Searching. I. The Screens". *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 147-152.
(13) Von Scholley, A. submitted for publication in *J. Chem. Inf. Comput. Sci.*
(14) Farmer, N. A.; O'Hara, M. F. "CAS ONLINE—A New Source of Substance Information from Chemical Abstracts Service". *Database* **1980**, 10-25.
(15) 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**, *23*, 93-102.
(16) Graf, W.; Kaindl, H. K.; Kniess, H.; Schmidt, B.; Warszawski, R. "Substructure Retrieval by Means of the BASIC Fragment Search Dictionary Based on the Chemical Abstracts Service Chemical Registry III System". *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 51-55.
(17) Graf, W.; Kaindl, H. K.; Kniess, H.; Warszawski, R. "The Third BASIC Fragment Search Dictionary". *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 177-181.
(18) Feldman, A.; Hodes, L. "Substructure Search with Queries of Varying Specificity". *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 125-129.
(19) "CAS ONLINE Screen Dictionary for Substructure Search", 2nd ed.; Chemical Abstracts Service: Columbus, OH, 1981.
(20) Balent, M. Z.; Emberger, J. M. "A Unique Chemical Fragmentation System for Indexing Patent Literature". *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 100-104.
(21) Adamson, G. W.; Lynch, M. F.; Town, W. G. "Analysis of Structural Characteristics of Chemical Compounds in a Large Computer-Based File. Part 2. Atom-Centered Fragments". *J. Chem. Soc. C* **1971**, 3702-3706.

Computer Storage and Retrieval of Generic Chemical Structures in Patents. 6. An Interpreter Program for the Generic Structure Description Language GENSAL

JOHN M. BARNARD, MICHAEL F. LYNCH,* and STEPHEN M. WELFORD
Department of Information Studies, University of Sheffield, Sheffield S10 2TN, U.K.

Received December 8, 1983

A computer program is described that carries out syntactic and semantic analysis of generic structures encoded in GENSAL, a formal language for the description of such structures, simultaneously generating an extended connection table representation of the structure. Desirable enhancements to the program in the areas of structure diagram input, nomenclature translation, and linear formula analysis are discussed.

INTRODUCTION

Part 2 of this series[1] presented a formal language, GENSAL, which is designed for the representation of generic chemical structures in a form that remains as close as possible to that used in chemical patent specifications and abstracts but that is sufficiently formalized to be amenable to automatic processing by computer. GENSAL consists of a mixture of conventional two-dimensional structure diagrams and text, the latter including various special *delimiter* words ("IF", "BEGIN", "OSB", etc.) and symbols ("/", "&", "$=", etc.), integers, chemical nomenclatural terms, and linear formulas. A comprehensive instruction manual for GENSAL has recently been prepared.[2] GENSAL has some features in common with the R_k notation used for generic query structures in the COUSIN system developed at the Upjohn Co.[3]

Part 4 of this series[4] described an extended connection table representation (ECTR) for generic structures, which is a complete and unambiguous (though nonunique) representation of a generic structure. The ECTR is based on connection tables and the use of parameters to the "chemical grammars"

0095-2338/84/1624-0066$01.50/0