The Shell Chemical Structure File System

F. G. STOCKTON*

Shell Development Company, Bellaire Research Center, Houston, Texas 77001

R. L. MERRITT

Shell Development Company, Biological Sciences Research Center, Modesto, California 95352

Received August 8, 1974

The Chemical Structure File (CSF) system services a computer-based chemical information file of about 60,000 compounds. The file is searched, on request, by programs which have substructure matching capabilities based on atom-to-atom connections. The system features input of chemical structures by typewriter, using simple typing rules. A typed structure is converted automatically to produce a unique linear encodement for storage and processing. The computer output from a file search is pictured as originally typed. Other system features are arbitrary definition of chemical fragments, an automatic and adaptive method of screen generation, and a flexible query logic. CSF handles multiple queries and all required file updating and maintenance in one file pass.

INTRODUCTION

The Chemical Structure File (CSF) is a computer-based chemical information system developed by Shell Oil Company beginning in 1963. CSF uses a structural formula input and output. The system does not depend on any external linear notation code, such as Wiswesser[1] or IUPAC,[2] and encoding structures and search fragments merely involves typing the molecular structures using a few conventions. The system also features fragment searching capabilities based on atom-to-atom representations whereby fragments with specified atoms and bonds may be compared to every structure in the file. Structures and fragments may be of any size up to 150 atoms, not counting hydrogens bonded to carbon. Fragment specification must be definite. Compounds with structures incompletely known or specified may be filed and retrieved, but may not respond to a search based on fragment matching. A system feature, called Dynamic Indexing, permits the retention of previous search results and the associated fragments, and these retained fragments can be referenced in subsequent searches. Techniques developed for use in CSF may find use in other systems and programs and should be applicable in any case where retrieval based on structural relationships is important.

A computer-based file is useful in order to answer many of the questions our chemists and biologists need to have answered. Questions related to specific compounds filed by registration numbers or molecular formulas can be answered with manual card files. Complex questions related to classes of compounds or compounds having various molecular features can be answered efficiently only by a computer-based system.

A principal design goal of the CSF system and one of its distinguishing characteristics is the use of the chemical structure itself as an index for the filed information. This index is generated from the input graphic representation of the structure by an automatic process. In contrast, some other systems arbitrarily assign an index to a structure, and some require the services of a specialist who encodes the structure using rules devised for the purpose. Arbitrary index assignment makes no provision for recognition of identical structures. Encodement by a specialist may be error-prone and expensive. Automatic encodement, for our limited purposes, solves the recognition problem and is error-free and economical.

The current Shell file includes chemical information on 60,000 compounds of agricultural and biological interest. One of the principal current uses of the Shell file is the preparation of lists of structures which are characterized by the presence (or absence) of a particular chemical fragment of interest, or by a logical combination of the presence (or absence) of several fragments. The Shell file is an automation of a manual file which existed beforehand and still exists and which is indexed by an arbitrarily assigned registration number. The registration numbers have the general form SD 30165, a prefix of two alphabetic characters, followed by a blank and one to five digits. The Shell file is indexed in three different ways, and all three can be used in retrievals: (1) by structure, (2) by registration number, and (3) by the results (including past results) of fragment matching. The retained results from prior runs may be used as a screen to limit the number of compounds explicitly matched against and so save effort.

SYSTEM FUNCTIONS

CSF was designed as a multifunction system to answer complex chemical questions with fragment matching techniques. The six functions and their system code names are FILE, FIND, DELETE, FIND NO, DELETE NO, and MATCH.

FILE—adds compounds to the file
FIND—finds one particular compound having a given structure
DELETE—deletes compounds from the file by structure
FIND NO—finds compounds by specified registration numbers
DELETE NO—deletes compounds from the file by specified registration numbers
MATCH—matches search fragments against compound structures stored in the file

All of the above functions may be used in any one CSF run.

CSF offers also a query logic which selects or rejects structures based on matching of multiple fragments, and a variant of the FIND NO function which provides for inclusion of extraneous categorical information in the dynamic indexing.

*To whom correspondence should be addressed.

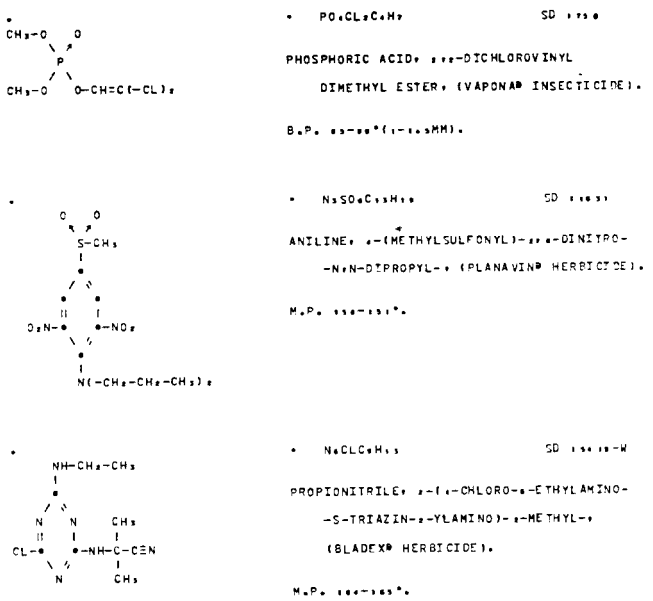


Figure 1. Computer generated output.

GENERAL PHILOSOPHY AND ORGANIZATION

The CSF file and retrieval system is designed to be approached with one or more drawn structures representing entire molecules or fragments. Except for the file manager's role, subsequent steps in the act of filing or retrieval are either clerical or automatic. There is no necessity for the submitter or inquirer to be instructed in special representational or encodement techniques or to learn a special language or procedures.

The system is designed for control by a File Manager. The functions of the file manager are

(1) Act as an intermediary and interpreter between the file and its users. Translate inquiries into the query logic of the CSF system.
(2) Monitor and control the quantity and quality of the filed information, using CSF functions provided for updating, deleting, etc. CSF provides adequate file population statistics, on request, on any run.
(3) Manage and be responsible for file security and backup, using available procedures.
(4) Control the indexing of the file, using options available with the dynamic indexing feature.
(5) Initiate improvements in the CSF programs, techniques, or procedures.

The CSF data file is serial, on magnetic tape, and unordered. On each run the entire file is read and copied to new reels. There are an average of twelve logical records (compounds) per physical record. The logical records are of variable length. Each contains

(1) Information about the record length and the lengths of its various segments.
(2) The registration number of the compound.
(3) The graphic structure of the compound as input (the Picture).
(4) A unique linear encodement of the structure, produced from the picture by automatic means.
(5) Text associated with the compound. The text includes the name, the molecular formula, and one or more physical properties.
(6) A bit string for use in retrieval logic, which contains the results of past attempted fragment matchings against the structure. It is called the dynamic indexing bit string.

Special records occur at the beginning of the file and at the end of each reel. The one at the beginning of the file contains all information needed to create the dynamic indexing bit string for new compounds added to the file.

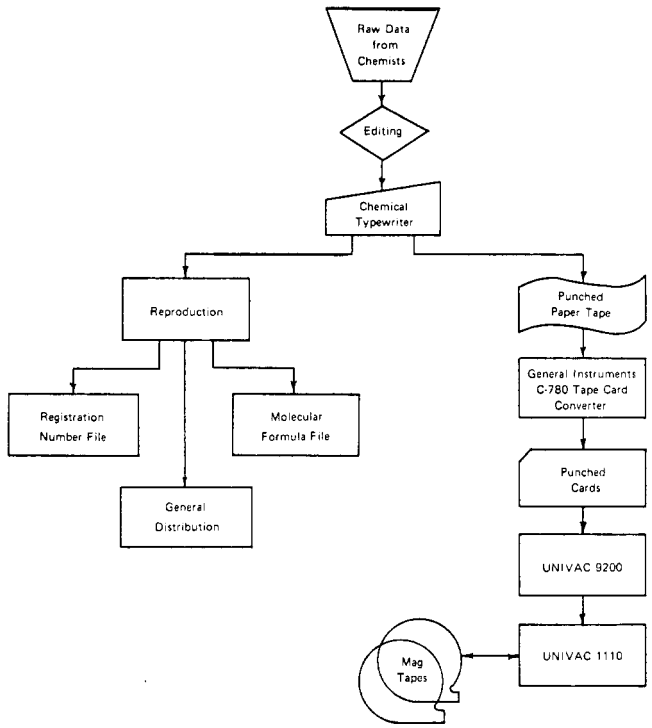


Figure 2. Chemical Structure File operations.

The CSF programs are written entirely in UNIVAC Fortran V and operate under the EXEC 10 operating system on a remotely located UNIVAC 1110 computer. The file manager has access to this computing facility through a UNIVAC 9200 terminal at his location. A special print bar[3] for the 9200 provides for output of chemical structure diagrams that are of acceptable pictorial quality for display to chemists and other file users as shown in Figure 1. The print bar has upper case letters only.

Input data for the file are prepared in conjunction with internal reporting procedures, so that either activity might be considered a byproduct. As is shown in Figure 2, raw data are submitted by the chemist. The editing process is handled by the typing clerk or the file manager, depending on the complexity of the situation. The edited information is then typed by the clerk, producing both a hard copy and a punched paper tape. The hard copy is typed in the form of 5 × 8 in. data cards and these are reproduced. The resulting cards are filed in a Registration Number File and a Molecular Formula File. These are manual files which complement and offer a backup protection for the CSF computer file.

The paper tape serves as an input to the CSF system and is converted to punched cards by a General Instruments C-780 Tape/Card Converter. The cards are then input to the CSF file in an 1110 run initiated at the remote 9200 terminal.

EXTERNAL INTERFACE

The keyboard of the Shell Chemical Typewriter[4] is shown in Figure 3. Besides the upper and lower case letters and the ten digits, the typewriter font includes other characters useful for typing chemical structure diagrams, including six orientations of single bonds, four of double bonds, a large dot to represent carbon (with an appropriate complement of hydrogens),[5] an intermediate size dot used in representation of free radicals, salt bonds, and complex bonds (upper case, key 42), and two orientations of an arrow (upper case, keys 36 and 40) to represent a coordi-

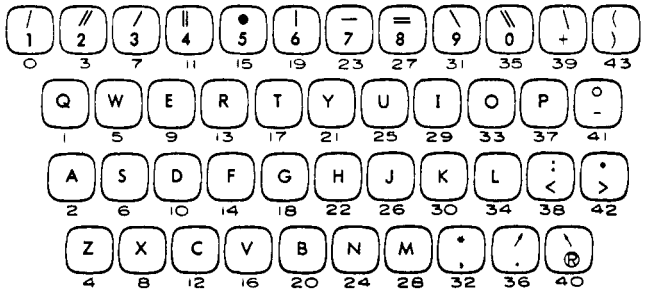


Figure 3. Typewriter key layout, Shell Chemical Typewriter.

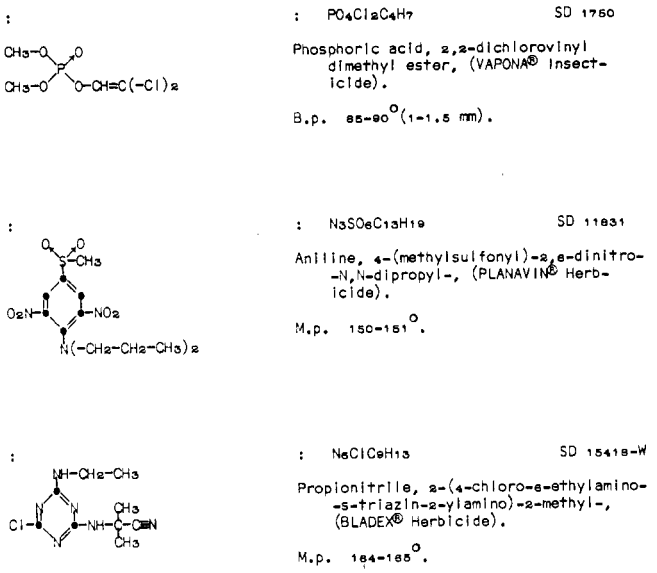


Figure 4. Typical input page copy.

nate-covalent bond. Ionic bonds are indicated with + and −. The triple bond is not a specific symbol and is typed by overstriking horizontal double and single bonds. The digit type faces are small, to provide the effect of subscripts, as shown in Figures 1 and 4.

Among keys not shown in Figure 3 are an index key, a reverse index key, a tabulation (tab) key, and a carriage return. The carriage return advances the paper two lines, simultaneously returning the typing unit to the left margin. The index key advances the paper one line, leaving the typing unit in its current position. The reverse index key retracts the paper one line without moving the typing unit. The tab key advances the typing unit to the next tab stop. Any key which causes a character to be printed also causes the typing unit to advance (move right) one print position after printing.

The typist is under instructions not to position the typewriter platen or typing unit by hand, but to do everything by key operations. Because of the reverse index function, this is possible. For each key operation used in preparing CSF input, an associated code is punched in paper tape by the typewriter. The tape provides the machine readable input for CSF processing.

An example of typical page copy produced by the typewriter is given in Figure 4. This page organization, which is associated with the use of 5 × 8 in. cards for internal reporting, does not allow typing of structures with a horizontal extent of more than 32 character positions. Large structures are often symmetrical, and nested parentheses are used to fit them into the space. In the few cases where a large structure has no symmetry, the typist arranges the picture to make best use of both the horizontal and vertical space (the vertical limit is 39 lines). Bonds may be extended (stretched) or crossed, but they may not turn corners. Legitimate atom symbols are hydrogen (H) through lawrencium (Lw) plus deuterium (D).

Bond varieties explicitly recognized include single, double, triple, coordinate-covalent, complex (=coordinate-covalent), ionic, and salt (=dot disconnect). Cyclic resonant bonds are recognized by implication from single–double bond alternation. CSF also provides a representation for free radicals.

Close examination of Figure 4 will reveal that some atoms are typed without connecting bonds. Hydrogen is always so typed, the bond to any other atom being implied by simple adjacency. Other conventional groupings, called glyphs, include NO_2, O_2N, HCl, H_2O, Me for methyl, Et for ethyl, Ph for phenyl, and $-(CH_2)_n-$, where n is an integer. Other glyphs could be added.

STANDARDIZATION FOR STORAGE

Several computer processing steps are needed to prepare an input structure for storage. From the input typewriter keystrokes, an image of the structure as typed is created within the computer. The image is scanned and an atom connection table representation of the structure is constructed from the information found by scanning. The connection table is transformed in several steps to a standard form, then condensed to a linear symbol string and stored. Henceforth, the structure is represented in file storage and, for some purposes, in the computer by the symbol string. For other purposes, it is represented in the computer as the standardized connection table. Conversions from one representation to another are done automatically by CSF programs as required. Any two different drawings of the same compound (with stereoisomers counted as identical, see below) will produce the same standard connection table and therefore the same symbol string. If an attempt is made to file a new compound with a structure identical with one already on file, the identity is detected and reported.

The processing from the initial connection table to the standard form has two motivations: (1) to remove ambiguities present in the drawn structure and (2) to decrease storage requirements. The ambiguities which CSF addresses arise from representation of cyclic resonant bonds by single–double bond alternation, from the use of glyphs, and from the multiple ways in which the atoms of a structure may be ordered for representation by a connection table. CSF makes no provision for stereochemistry in structure processing, so ambiguities of stereoisomerism remain unresolved, unless by a note added to the text associated with the structure.

First, cyclic resonant bonds implied by single–double bond alternation in even-numbered rings are identified and are assigned an explicit internal representation. Next, glyphs are replaced in the structure by the explicit atom and bond groupings they represent. Then the carbon atoms are classified into five types by the number of hydrogens attached to them and are represented internally by five different symbols, Cv, Cw, Cx, Cy, Cz (C, CH, CH_2, CH_3, CH_4). Hydrogens attached to carbon do not appear explicitly in the internal representation and do not count toward the 150 atom size limit for CSF structures.

Finally, the connection table is converted by a standardization algorithm to a standard form. This amounts to derivation from the structure alone of a unique ordering of its atoms, followed by condensation of the standardized connection table to a standard linear symbol string. The standardization algorithm we use is rigorous (nonheuristic). The ordering it produces is based on arbitrary precedence assignments for atom and bond symbols (*e.g.*, N > O > Cy > Cv > 2 > 1, where 2 and 1 are internal symbols for double and single bonds) and on rules for selecting a "next" atom. In the selection, the algorithm prefers atoms con-

nected to the last previous atom, or to the one before that, etc. This tends to organize the structure into long connected chains of atoms, a useful organization for fragment matching.

Because of valence limits, the connectivity and symmetry of chemical structures are limited as compared with possible mathematical structures (graphs). The current CSF standardization algorithm will not produce an incorrect standardization of any graph, but it will fail to standardize certain graphs of high symmetry. It has not failed for any chemical structure even though some in our file are very large and symmetrical. Nevertheless, we have studied the standardization problem further and are provided with an improved algorithm if we should need one.

FRAGMENT SPECIFICATION AND MATCHING

In our terminology, a fragment may range from monatomic, like P (phosphorus), to complex arrangements of many atoms and bonds. It is specified by a drawing, just as compounds are, but some additional external symbols are needed and are reserved for use in fragments.

(1) Carbon atoms
  C = any type carbon (C, CH, CH_2, CH_3, or CH_4)
  Cv = C (no hydrogens)
  Cw = CH
  Cx = CH_2
  Cy = CH_3
  Cz = CH_4
(2) Any of the carbon atoms above or any other atom may be specified generally by the symbol J. That is, J in a fragment is matched by any atom (except hydrogen attached to carbon).
(3) Any set of atoms may be grouped in an atom table. The symbols Ja, Jb, and the like mean "any atom of atom table A," "any atom of atom table B," etc. Thus one symbol can stand for "a halogen," another for "sulfur or oxygen," etc. This often reduces the number of fragments that have to be specified for a query.
(4) An unspecified bond in a fragment is represented by an asterisk (*) and will be matched by any bond.
(5) Any set of bonds may be grouped in a bond table. The symbols *a*, *b*, and the like mean "any bond of bond table A," "any bond of bond table B," etc.

Internally, a fragment is specified by a connection table, with associated atom and bond tables, if any are used in the fragment specification. A filed structure must also be expanded from the linear form to a connection table for matching. The matching algorithm operates on the two connection tables to see if assignments can be made in the structure for all the atoms and bonds of the fragment. Because fragment matching is a lengthy calculation, it is attempted only for structures which have passed a preliminary screen, the molecular formula screen. The molecular formula screen requires that the molecular formula of the fragment be included in the molecular formula of the structure.

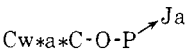
A few sample fragments illustrate the atom and bond specifications. Consider first the fragments

The first fragment will match any structure containing a coordinate-covalent bond. The second will match any structure with a five-membered ring. Next consider

C*C-O-P

This fragment states that phosphorus (P) must be attached by a single bond to oxygen (O). The oxygen must in turn be attached with a single bond to an any-type carbon atom (C, CH, CH_2 in this fragment). This carbon must then be attached to at least one other any-type carbon atom with any possible bond (single, double, triple, or resonant).

The previous fragment might be further restricted as follows



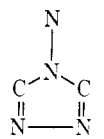
where

Ja = any atom from atom table A
a = any bond from bond table A
atom table A = O or S (oxygen or sulfur)
bond table A = double or triple bonds
Cw = carbon with one hydrogen attached (CH)

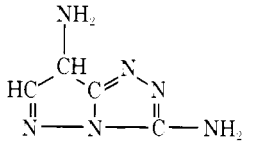
This fragment says that a phosphorus atom must be attached with a coordinate covalent bond to either oxygen or sulfur. Also, the phosphorus must be attached to an oxygen with a single bond. The oxygen must be attached to an any-type carbon with a single bond. This carbon atom must be attached to a carbon with only one hydrogen (CH) with either a double or triple bond. The use of the bond table will prevent response by a structure with a resonant bond connecting the two carbon atoms.

CSF gives matches for compounds containing the fragment in any arrangement, even when embedded in a complex structure. An example of a compound containing an embedded fragment in the molecule is shown next.

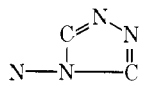
Search fragment



Compound with embedded fragment

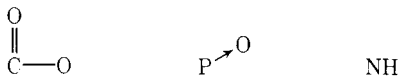


The search fragment requests all compounds having the basic atomic arrangement for 4-amino-4*H*-1,2,4-triazoles and also allows the amino nitrogen and the carbons to be substituted in any manner. Therefore, the responding compound does indeed contain the search fragment because of the following atom arrangement.

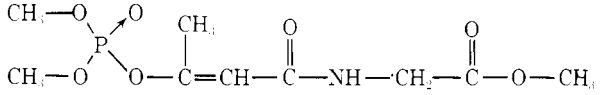


If such compounds are not wanted, the fragment will have to be more stringently specified.

In CSF fragment matching, disjoint components can be handled as a single fragment. For such a fragment, all structures will respond which have the various components somewhere in the molecule. An example of a disjoint fragment follows.



The disjoint components may be connected or not connected in the responding structure; however, all must be independently present. An example compound that will respond to the above disjoint fragment is



DYNAMIC INDEXING AND QUERY LOGIC

To exploit the capabilities provided by fragment matching, a feature known as Dynamic Indexing has been included in the CSF system. Dynamic indexing keeps track of

compounds that have satisfied past fragment matchings, and it also indexes all compounds added to the file against the past fragments. The indices so generated serve as a screen, which limits unnecessary fragment matching and shortens the computer runs. The screen is not fixed but reflects the fragment population which has been posed to the file. It is therefore adaptive to file usage patterns and will shift to facilitate the work if file queries shift to a new area of interest.

Dynamic indexing provides three CSF classifications of fragments. The first classification is called Current Fragments. These are fragments being matched in the current CSF computer run. At the end of the run, the current fragments are automatically added to the set of Historical Fragments, which is the second fragment classification. CSF will retain up to 72 historical fragments and when more than 72 have been added, the oldest are deleted from the system and are irrecoverable. Permanent Fragments, the third classification, are retained indefinitely by the system and can accumulate to any number desired. They are created, from historical fragments, only at the discretion of the file manager and may be deleted by him. In this way the file manager's foresight, knowledge, and wisdom may be progressively incorporated in the file indexing.

Thus on any run a query may be based on fragments introduced with the run, fragments which were introduced in the last few runs, and fragments from still earlier runs which have been selectively retained. The limit of 72 historical fragments is not fixed and can be increased (or decreased) at any time.

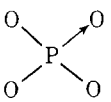
Any identified subset of the structures on file may also be assigned an index which represents membership in the subset. Once assigned, such indices are handled by the same mechanism as those originating from fragments, and can be referenced in queries on the same basis. For example, an index might represent insecticidal activity in a specific range against the housefly. In this way any available categorical information acquired about the file population may be summarized in the indexing and preserved for later use. Such indices are not automatically updated for new structures added to the file, as those for fragments are.

It is possible to use the logic operators AND, OR, and NOT among any of the three types of fragments, Current, Historical, and Permanent, in a CSF query. For example, a user may have queries based on the following fragments:

A. Current fragment  (pyridine)

B. Current fragment  (pyrimidine)

C. Historical fragment NO_2 (nitro)

D. Permanent fragment  (phosphate)

Three of many questions which might be asked are:

Query 1: Find all compounds containing A but not C or D.

Query 2: Find all compounds containing B but not C or D.

Query 3: Find all compounds containing either A or B, but not C or D.

The first query produces a list of compounds containing all pyridines which do not have the nitro or phosphate groups; the second query, all pyrimidines not having nitro or phosphate groups; and the third query, a combined list of the responses to the previous two queries. Each query will result in an ordered listing of all responding compounds, by descending registration number. Each response will contain the graphic structure, molecular formula, registration number, chemical name, and a melting and/or boiling point, as originally typed. No fragment matching is required for the nitro or phosphate fragments, since their indices are already available for use by the queries. The query logic available for query construction is more flexible and powerful than has been illustrated here, more so than we have found any need for in our work.

EFFICIENCY AND OPERATIONAL COST

We do not give dollar figures because they are commonly so affected by arbitrary intracompany cost distribution policies that comparison is meaningless. The length of CSF computer runs on the UNIVAC 1110 depends primarily on the amount of computation to be done rather than on the size of the file. Fragment matching is the most expensive system activity as well as the most valuable. Runs are longer if there are many new fragments (current fragments) to be matched. They are longer still if some of the fragments reference atom tables. Typical recent run times on the UNIVAC 1108 (a precursor of the 1110) vary from 23 to 50 min. We expect overnight, rather than immediate, service. We are charged at a favorable rate and are satisfied with these run times and the value produced. The 1110 is faster but we do not yet have comparable performance figures for it.

CSF uses the molecular formula of the compound as a preliminary screen, and will not, for example, expand a filed structure to a connection table for matching against a phosphate fragment if the structure contains no phosphorus, or fewer than four oxygen atoms.

The fragment matching capabilities, dynamic indexing, and query logic make CSF a powerful tool for searching a file of chemical structures, one that cannot be approached by manual methods. The system is in frequent and growing use by its intended clients. We are improving CSF and planning further improvements including, perhaps, conversion to run on IBM equipment. We expect to offer the system for use by others.

ACKNOWLEDGMENTS

We wish to acknowledge the work and support of J. M. Mullen, L. F. Ward, C. B. Witze, S. B. Soloway, S. M. Lambert, E. M. Ellentuck, C. R. Delaune, H. J. Beetle, and J. H. Guinn, all of whom have contributed significantly to realization of the Chemical Structure File system.

LITERATURE CITED

(1) Smith, E. G., "The Wiswesser Line-Formula Chemical Notation," McGraw-Hill, New York, N. Y., 1968.

(2) Anonymous, "Rules for I.U.P.A.C. Notation for Organic Compounds," Wiley, New York, N. Y., 1961.

(3) UNIVAC Part No. 5032966-00.

(4) Mullen, J. M., "Atom-by-Atom Typewriter Input for Computerized Storage and Retrieval of Chemical Structures," *J. Chem. Doc.*, **7**, 88–93 (1967).

(5) This representation was a suggestion of Mr. H. P. Luhn, International Business Machines Corp., in a private communication to Mr. J. M. Mullen.