program and use the fragment-vectors output from this program as structure input in the correlation study. If we succeed in establishing a model which allows reasonably good prediction of the toxicity of compounds with known structures, we would have a tool which could be used not only to estimate toxicity data but for validation of toxicity data in the data bank. In particular, gross transcriptions errors in LD50 data or mistakes with dose units could be detected in this manner.

CONCLUSIONS

The inclusion of data from RTECS in the ECDIN data bank has made part of the Registry accessible for on-line retrieval. The object of the exercise was not merely to create an "on-line" version of the Registry, but rather to use RTECS as a tool in the development of the ECDIN data bank. Nevertheless, our experiment has shown that is is possible to handle the data from the magnetic tape versions of RTECS automatically and to convert them into a form suitable for input into a data retrieval system without any manual editing. The toxicity data have thus been made available in searchable form with ECDIN, to facilitate their combination using Boolean logic with production estimates, use patterns, and other data elements in the ECDIN data bank.

ACKNOWLEDGMENT

We wish to thank the editors of RTECS and NIOSH for making the magnetic tape versions of RTECS available to us, and to acknowledge the grants from the Commission of European Communities and from the Danish Natural Science Research Council to Ole Nørager, which have made this work possible. We also wish to acknowledge the advice given by John C. Gage (consultant to the Commission of the European Communities) on the conversion of the RTECS codes into free text and the assignment of data to ECDIN fields, and to thank Marinus de Groot of the Joint Research Centre for assistance with the translation of the bibliographic codes. We also wish to thank Philippe Bourdeau, Friedrich Geiss, and Heinrich Ott for their encouragement and advice.

REFERENCES AND NOTES

(1) F. Geiss and Ph. Bourdeau, "ECDIN, an EC Data Bank for Environmental Chemicals", *Environ. Qual. Saf.*, **5**, 15–24 (1976).
(2) M. Boni, F. Geiss, J. H. Petrie, and W. G. Town, "The Development of a Data Network on Chemicals and Their Effects on the Environment. The Environmental Chemicals Data and Information Network (ECDIN) of the European Communities", Proceedings from the EURIM II Conference, 23–25 March 1976, Aslib, London, 1977, pp 145–147.
(3) G. Gaggero, C. Lunghi, and C. Mongini-Tamagnini, paper presented at the International Computing Symposium, Venice, April 12–14, 1972.
(4) The Toxic Substances List, 1974 edition.
(5) The Registry of Toxic Effects of Chemical Substances, 1975 and 1976 editions, The National Institute for Occupational Safety and Health, Rockville, Md.
(6) CODEN for periodical titles, American Society for Testing and Materials, Philadelphia, Pa., 1966.
(7) G. W. Adamson and D. Bawden, "A Method for Structure-Activity Correlation Using Wiswesser Line Notation", *J. Chem. Inf. Comput. Sci.*, **15**, 55–58 (1975).
(8) D. R. Eakin, E. Hyde, and G. Palmer, "The Use of Computers with Chemical Structural Information: The ICI CROSSBOW System", *Pestic. Sci.*, **5**, 319–326 (1974).

PAGODE†: The Computer-Based Chemical Information System of CLIN MIDY Research Center

SAMUEL BERDUGO,* JEAN BOITARD, JEAN PAUL GERVOIS, ANNE MARIE SEGRETAIN, and ODILE PIETREMENT

Centre de Recherches CLIN MIDY, 34082 Montpellier Cedex, France

Received May 9, 1977

In collaboration with ARDIC, CLIN MIDY Research Laboratories have implemented an in-house database using the DARC topological coding system. This paper describes the general organization of the PAGODE system and its capabilities.

This paper describes CLIN MIDY'S chemical information system. Chemical information must be completed by biological information as is usually done in pharmaceutical firms. But creating a numerical comparative biological database is not an easy task. However, chemical information exists in a standard form ready for computer processing. So our first step was to store the in-house chemical structures for computer handling. The in-house documentation problem requires a more accurate coding system than the huge international chemical system of documentation, and it has quite a different purpose, since, in getting a precise description of any sequence of atoms in a series of molecules, structure–activity relationships use sophisticated coding systems. So the DARC system was adopted. Professor Dubois and his team of scientists in Paris created the DARC[1] system whose diffusion in the chemical industry is promoted by ARDIC[4] (Association pour la Recherche et le Développement de l'Informatique Chimique; Research and Development of Chemical Data Processing Association) which helped us with the general organization of the CM database and with the coding of the compounds.

The whole project was divided into several steps. The first step was to create the chemical database and to test it in batch processing. This article deals with the problems encountered during the first phase. According to our first results, the DARC system meets our initial requirements.

CODING SYSTEM

Choice of the Coding System. Before choosing the DARC system, we compared other chemical systems of coding commonly used in the pharmaceutical industry, mainly WLN and Ringcode used by DERWENT. Our requirements concerning a coding system were as follows: (1) a one-to-one correspondence between structure and code; (2) the possibility of constructing search keys or screens from the code, which is, of course, a computer requirement and also necessary if one wants to create an inverted file and to get a quick answer from the computer; (3) a description of the molecular topology (structure–activity relationships require the capability of describing any structure, including the sequence of atoms

† Programme Automatique de Gestion et d'Organisation de Données.

0095-2338/78/1618-0140$01.00/0

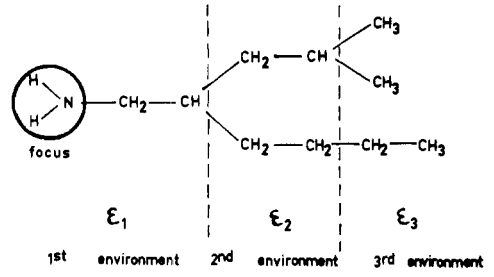

Figure 1. Propagating a limited environment.

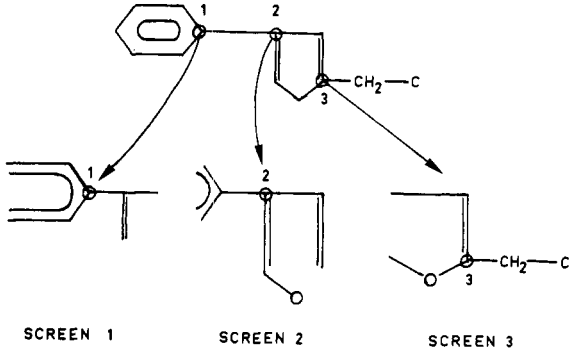

Figure 2. Generating the FRELs.

belonging to cycles as well as to side chains); and (4) the possibility in the near future of automatically generating the code from the picture of the chemical structure on a CRT terminal and by so doing avoiding manual coding.

From our point of view, the DARC system is one of the codes which best fulfill these conditions. It is not as compact as WLN, but the in-house compounds are not numerous enough to be a real disadvantage in this respect. The screen system is easy to generate and gives a fairly good overlapping of the molecules.

As in other systems, we must use the screen system for the first search phase and then an atom-by-atom matching routine to cancel out noise.

Basic Principles of the DARC System. The DARC system is based on a topological code. A chemical formula is considered as a graph described by three main parts;[2,3] first the DEX (Existence Descriptor), then the DLI (Bond Descriptor), and the DNA (Nature of Atom Descriptor). The starting point of the description is the focus; its choice is based on formal rules. The description covers the molecule by successive layers of two bonded atoms each. When the first environment is described, the terminal atoms are new origins for the next description, and this process goes on until the molecule is completely described (Figure 1).

The topological screens (search keys) can be obtained as follows. The topological screens, called FRELs in DARC terminology, are centered on each atom of the molecule whose connectivity is three or more, and are composed of two consecutive layers of bonded atoms (one limited environment) (Figure 2).

In the first phase of testing the system, we kept only these topological screens. Other chemical screens, such as side-chain screens, may be introduced later.

Principles of Structure Searching. The 12 000 compounds so far coded constitute the experimental database. The topological screens were automatically generated from the DARC code. The search procedures were tested on a batch-processing computer. The general organization is based on inverted files to get a quick answer in the future, and to simulate the next step, i.e., interactive processing. During this trial period, the effectiveness of topological screens was tested.

Coding the Structures. ARDIC carried out the coding of all the CM structures, as well as the generation of topological

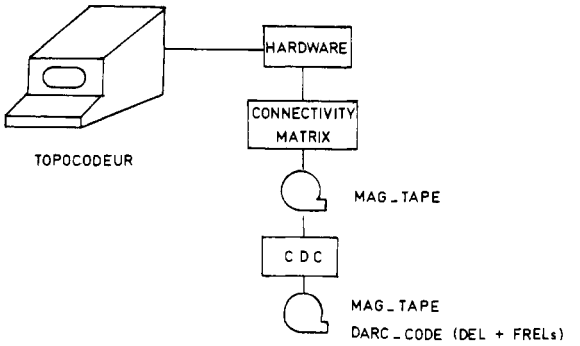

Figure 3. ARDIC Coding System.

screens, by means of "TOPOCODEUR". The picture is typed on the off-line graphic terminal keyboard, and the hardware generates the connectivity matrix and writes the result on a magnetic tape. Then the tape is processed by a specific computer program which first generates the DARC code from the connectivity table, and then the corresponding screens (see Figure 3).

PAGODE SYSTEM

The name PAGODE was given to the computer program set written by CLIN MIDY's data processing team, for consulting the chemical database.

The General Organization. The PAGODE system, which is composed of about 30 assembly programs (the whole set reaching 6000 instructions), is implemented on an IBM 360-40 computer running under DOS. Data are stored on three files: dynamic data file (DD), screen file (SCREEN), and screen inverted file (INV).

1. DD File (Figure 4). In this file, we store all the data involving a compound: DARC encoded developed formula, molecular formula, and biological information. There is one record per compound which is directly retrieved with the five-digit CM number.

In fact, by means of a suitable mathematical function, it is possible to calculate from the five-digit CM number an address falling in the range zero to maximum number of compounds (hash-coding). However, this function does not provide a one-to-one correspondence and for two different numbers the calculated locations may be the same. For this reason the DD file is divided into two subfiles: a basic subfile where all the prime addresses are stored and an overflow subfile for the duplicate addresses (synonym subfile). To link these two subfiles, each record includes a pointer to the next synonym (Figure 4).

2. SCREEN File (Figure 5). This file contains all the screens present in all the structures. There is a large undetermined number of screens (FRELs) of general formula N (N = atomic number of focus) (DEX/DLI/DNA), and their DARC coding varies in length owing to the variable size of DLI and DNA descriptors (for more suitable computer-based storage, see Figure 10). However, the number of DEX being limited (55), we were able to divide the SCREEN file into two parts (Figure 5).

The screen part is divided into 55 memory areas (regions), one per DEX. Each region—of variable storage size—contains all the existent screens which have this DEX. With each screen one can see its address in the INV file (Figure 5). To accelerate screen retrieval and to minimize disk accesses, the blocking factor of logical records (screens) is at its highest; therefore, deblocking in core memory is performed quite rapidly.

3. INV File (Figure 6). Each screen of the SCREEN file has an entry in the INV file. This entry consists of an inverted list[7] of the CM compounds which go with this screen. Since

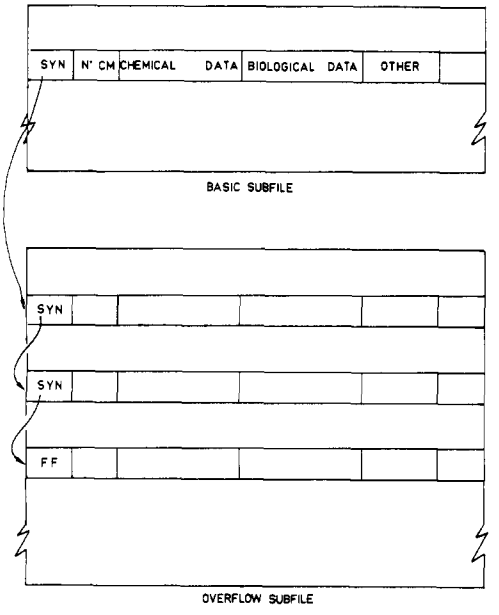

Figure 4. DD file organization.

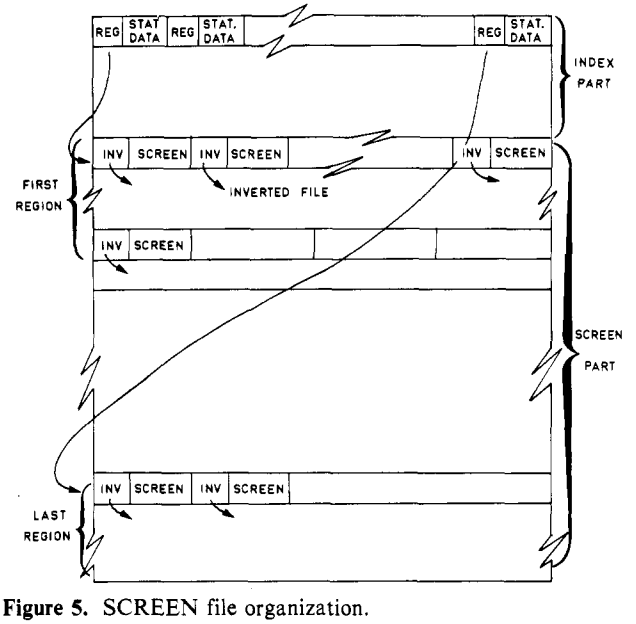

Figure 5. SCREEN file organization.

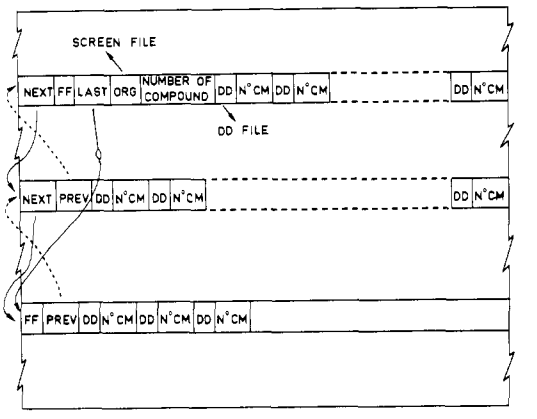

Figure 6. INV file organization.

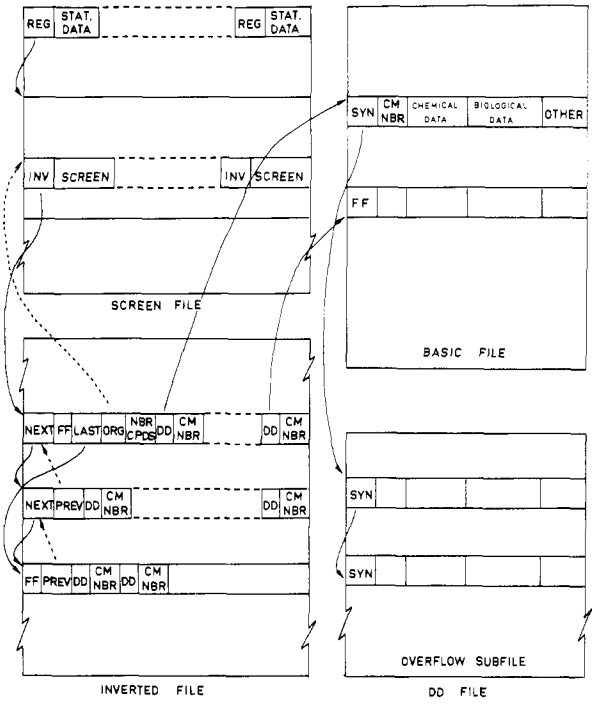

Figure 7. PAGODE file organization.

screen occurrence is variable, the list is of variable size. So in order to avoid unnecessary disk and buffer storage, this list is divided into physical records of fixed length. When one block of CM numbers is full, an additional record is required and PAGODE provides the first free storage in the INV file. To link the physical records which correspond to the same screen, the following pointers are provided: next block pointer (NEXT), previous block pointer (PREV), and last record pointer (LAST) (see Figure 6). In addition to these in-file pointers, there is also the origin pointer "ORG" (for linkage with the origin screen in SCREEN file) and the DD file pointer for each CM number (see also Figure 7).

Handling of Chemical Data. 1. Query. Chemical structure retrieval is done using DARC screens. The user is provided with a query language which enables him to have a combination of screens (operands) and Boolean operators (OR, AND, NOT, X: repetition operator). For a set of screens S_1, S_2, . . . belonging to a structure, a question may be composed as follows:

$$(S_1 \text{ AND } ((S_2 \text{ OR } S_3) \text{ OR } (S_4 \text{ AND } S_5)) \text{ NOT } (S_6 \text{ AND } S_7)) \quad (1)$$

The main features of this language are the following.

Parenthesis imbrication is indifferent and may be as complex as needed.

There is a free choice in card format, provided there is one screen per card.

A DARC screen may be open or closed; the descriptors of a closed screen are clearly defined: DEX, DLI, DNA, i.e., topology, bond, and atom nature. In the example given in Figure 8a, when R_1 and R_2 are determined, the FREL is closed; on the other hand (Figure 8b), R_1 and R_2 indetermination leads to a set of several screens.

Attention must be paid to the difference between indetermination of the connectivity node such as R_1 and any other node, such as R_2. In fact, an indetermination of R_1 type leads to a subset of screens with the same topological DEX descriptor (because, as R_1 belongs to the second atom layer of the DARC screen, the total number of connectivity nodes must be constant, as the variation is related only to R_1 atom and bond nature which maintains the DEX descriptor unchanged). However, an R_2-type indetermination leads to a maximum of four screen subsets with four different DEX's (one for a single atom in R_2, a second one for one substitution in R_2, . . .).

The PAGODE query subsystem links different operations as follows (flowchart of Figure 15): (a) decoding and syntax control of input screens; (b) retrieval in SCREEN file of the

Figure 8. (a) A closed screen (R_1 and R_2 determined). (b) An open screen (R_1 and R_2 undetermined).

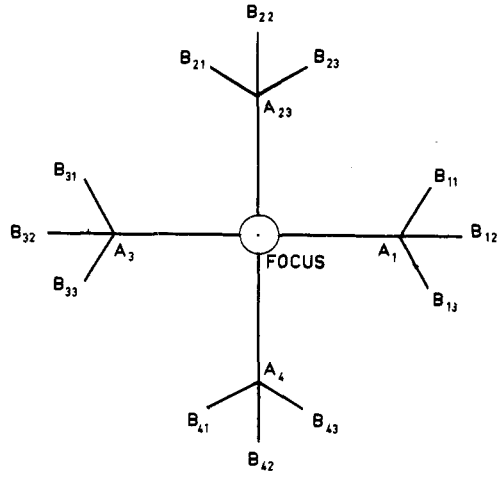

Figure 9. SCREEN graph.

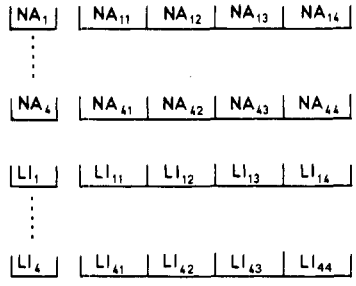

Figure 10. SCREEN internal code.

question screens. The SCREEN graph may be represented in general by Figure 9.

A FREL is clearly defined if, in addition to the topological DEX descriptor, the following two parameters are known for each node: nature of A_i or B_{ij} atom (NA); nature of A_i–focus bond or B_{ij}–A_i bond (LI). This is why, for a given graph, the SCREEN internal computer code is as indicated in Figure 10.

A_i and B_{ij} positions are not affected at random, but follow several priority rules which depend on the nature and number of atoms, bond nature, etc., as illustrated in the following example: let us assume that one asks for open screens of a given general structure (Figure 11a). The screen of an analogous graph with R_1 = C will fit and the B_{12} position will be attributed to connectivity node C.

However, if R_1 = Cl (Cl atomic number > N atomic number), R_1 will now take in B_{11} position; N will be changed into B_{12} position, and, in the internal coding (Figure 10), NA_{11} and NA_{12} "boxes" will be inverted as will be LI_{11} and the LI_{12} "boxes" ("box"-inversion).

Similarly, if indetermination concerns an A_i position (Figure 11b) there will be inversion between two lines NA_i, and the two corresponding lines LI_i according to the nature of the atom and the bond in R_1 (line inversion).

Therefore, in the case of an open-screen retrieval (e.g., S_2), PAGODE generates the maximum number of possible graphs (represented by dotted bonds in Figure 8b), i.e., the maximum number of DEX's and selects the regions which might contain suitable screens.

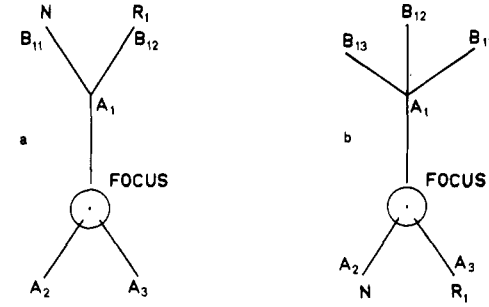

Figure 11.

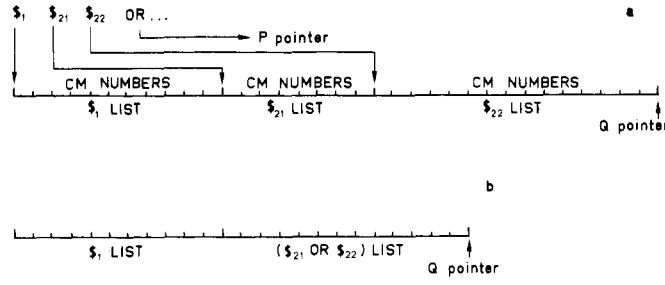

Figure 12. Work area (a) before operation, (b) after operation.

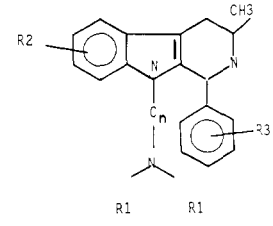

Figure 13. β-Carbolines.

Then, within these regions, the system searches, by line inversion (A_i indetermination) or "box" inversion (B_{ij} indetermination), the screens with the fixed part of a structure by identifying the sets (NA_i, NA_{ij}), (LI_i, Li_{ij}), (i = 1, 4; j = 1, 4) with those of the question screens.

The retrieved screens (e.g., S_{21}, S_{22}, . . ., S_{2n}) are all suitable and in expression 1 S_2 must be replaced by

$$(S_{21} \text{ OR } S_{22} \text{ OR } \ldots \text{ OR } S_{2n}) \quad (2)$$

The screens are then replaced in expression 1 by their addresses $\$_{21}$, $\$_{22}$, . . ., $\$_{27}$ in the INV file:

$$(\$_1 \text{ AND } (((\$_{21} \text{ OR } \$_{22}, \ldots, \$_{2n}) \text{ OR } \$_3) \text{ OR } (\$_4 \text{ AND } \$_5)) \text{ NOT } (\$_6 \text{ AND } \$_7)) \quad (3)$$

(c) Expression 3 is not easy to handle; it is translated into inverted Polish notation:

$$\$_1\ \$_{21}\ \$_{22} \text{ OR } \ldots \$_{2n} \text{ OR } \$_4\ \$_5 \text{ AND OR } \$_6\ \$_7 \text{ AND NOT} \quad (4)$$

(d) Query processing: while a pointer P scans expression 4, specified PAGODE modules read the corresponding lists in INV file at locations $\$_1$, $\$_{21}$, $\$_{22}$, . . ., $\$_7$ and store them in a buffer which is scanned by a pointer Q. Each time a Boolean operator is encountered, the requested operation is performed on the two earlier operands and the result list replaces the two lists. So in the example of expression 4 the first operation will be OR on $\$_{21}$ and $\$_{22}$ lists (see Figure 12).

This method offers some real advantages, such as speed-processing, omission of parentheses (compare expressions (3 and 4), and possible intermediate answers to the question, if requested.

(e) Result printing: the printing example in Figure 14 is a β-carboline retrieval of the general structure given in Figure 13.

(f) Next query (see Figure 15).

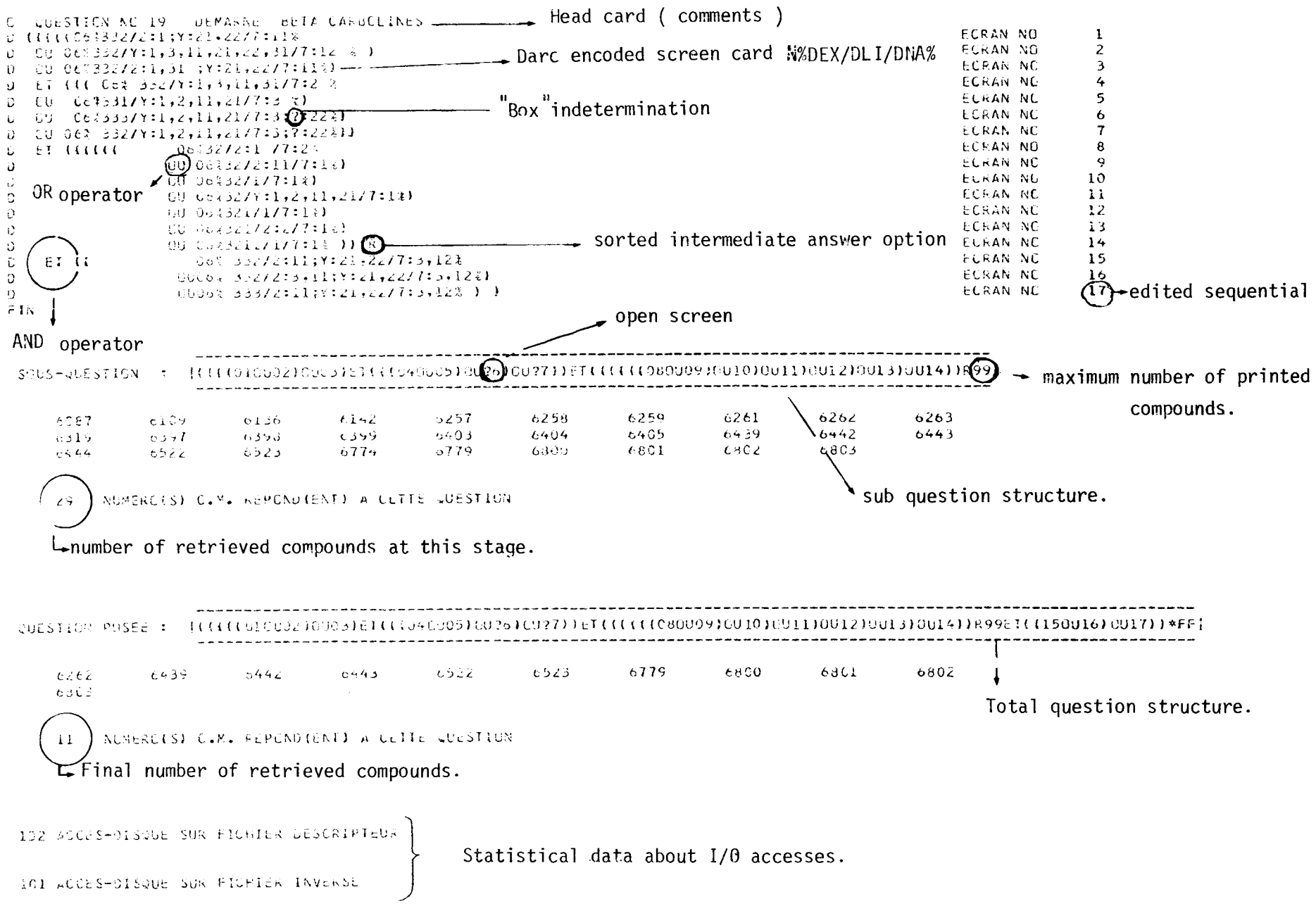

Figure 14. Example of printing.

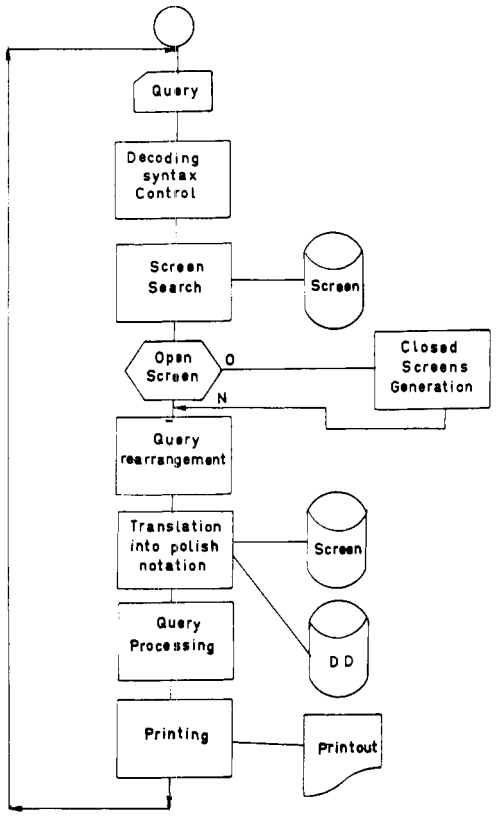

Figure 15. Inquiry.

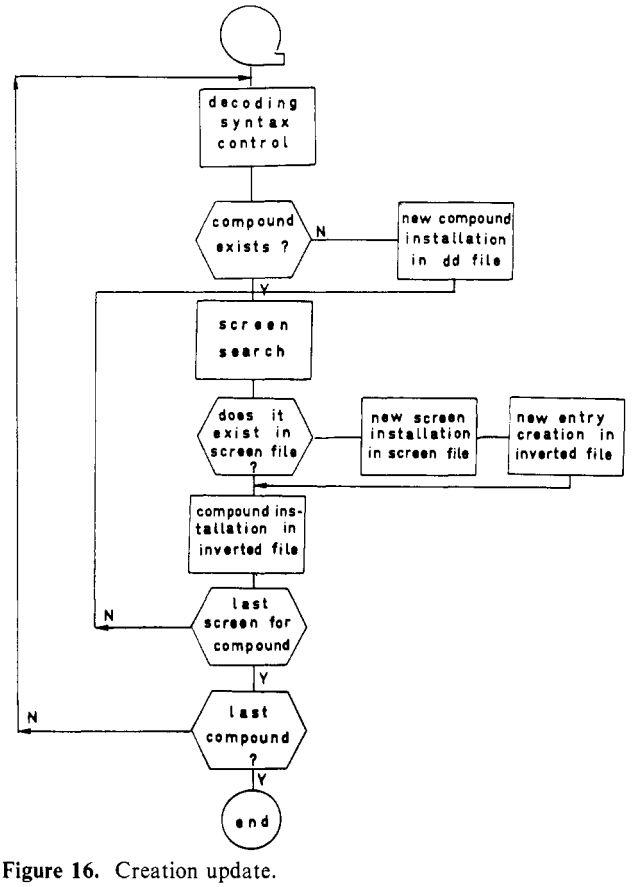

Figure 16. Creation update.

2. Creation-Updating of Compounds. When a new molecule is synthesized in the Research Center, it is coded according to the procedure in Figure 3. Then the corresponding PAGODE subsystem carries out the following tasks (see flowchart Figure 16): decoding and syntax control of input data; entry in DD file in the case of a new molecule; screen-retrieval loop (if the screen does not exist, it is stored in SCREEN file and a new entry is created in INV file); and compound writing in the corresponding inverted list of INV file.

3. Utility Programs. In order to make query and creation-updating easier and to perform a computer set as modular as possible, PAGODE provides some useful utility programs: hash-coding routine, screen retrieval module, closed-screen generation from an open screen, read–write of any record in any file, logical Boolean expression translation into Polish notation, and AND, OR, NOT, X operating function.

SYSTEM EVALUATION

Query execution time depends upon the following.

(1) Screen number: the larger this number, the longer retrieval time is. Obviously, when the number of open screens is high, the number of generated closed screens increases very rapidly, and consequently execution time increases proportionally.

(2) Question–structure and operators performed: for example, AND and NOT operators are slower than OR and X operators. In fact, if a two-operand list includes, respectively, n and m CM numbers, AND and NOT operators will require a maximum number of $n \cdot m$ comparisons to select the suitable compounds while an OR operation is performed simply by arranging the two lists end-to-end.

(3) Screen occurrence: in the case of frequent screens, inverted lists of INV file extend, and therefore execution time increases proportionally.

For these reasons, trying to evaluate question execution time is a problem. But we could safely say that an example-query composed of five closed screens, two open screens (each generating between 50 and 100 closed FRELs), and four sorted intermediate prints would work out between 50 and 70 s of CPU time on an IBM 360-40 computer. But, if all the screens of the question, i.e., between five and ten, are closed, which already represents a complicated structure, processing is much quicker (about 10 CPU s).

RESULTS

From our experience so far acquired, we cannot yet make a decisive statement on the DARC coding system. However, it is possible to state some observations.

PAGODE does not produce silence.

The noise which we noticed, and which was sometimes quite considerable, was to be expected. We knew the reasons for this which are that there were no screens in some carbon chains and that the FRELs already existing were badly arranged.

Condensed heterocycles are well described because of good overlapping. More generally when screens overlap the structure properly, noise rate is low.

The number of screens needed to retrieve a structure is small in most cases (five to seven).

FREL encoding requires much attention.

It may thus be concluded that FREL inquiry is generally satisfactory, although noise is its biggest disadvantage, and it could be worthwhile introducing new type screens, which would be more suitable in some unfavorable situations (chain screens, for example)

But we do not really think that noise is a major drawback for an inquiry system. In fact, within reasonable limits, it may even be an enrichment for research in structure–activity relationship,[5,6] as answers include some unexpected but interesting structures (for example, same atom arrangements but cyclized structure when expecting chain structure).

REFERENCES AND NOTES

(1) J. E. Dubois and H. Viellard, *Bull. Soc. Chim. Fr.*, 900 (1968).

(2) J. E. Dubois and H. Viellard, *Bull. Soc. Chim. Fr.*, 905 (1968).
(3) J. E. Dubois and H. Viellard, *Bull. Soc. Chim. Fr.*, 913 (1968).
(4) J. E. Dubois, *J. Chem. Doc.*, **13**, 8 (1973).
(5) J. E. Dubois, *Bull. Chim. Thérapeutique*, 65 (1972).
(6) J. E. Dubois and H. Herzog, *J. Chem. Soc., Chem. Commun.*, 932 (1972).
(7) D. Lefkovitz, *J. Chem. Inf. Comput. Sci.*, **15**, 14–19 (1975).

Structural Search Codes for On-Line Compound Registration

LINDSAY A. EVANS, MICHAEL F. LYNCH,* and PETER WILLETT

Postgraduate School of Librarianship and Information Science, University of Sheffield, Western Bank, Sheffield, S10 2TN, United Kingdom

Received November 8, 1977

A topological index has been developed which can discriminate between isomers in a molecular formula group. This index could be used, in combination with the molecular formula, to provide rapid access to those few compounds in a large chemical structure file which must be compared with a query structure at registration.

1. INTRODUCTION

One of the most common tasks performed by a chemical structure information system is that of registration. This involves determining whether a structure is already present in a machine-readable compound file or if it is new and must be added to the collection. As the compound is entered, it may be given a registration number which serves as a unique link to other information, such as bibliographic references or property data, subsequently added to the file. To ensure that information concerning two, or more, compounds is not to be confused, the structure representation chosen should, ideally, be both unique and unambiguous: a representation is unique if it is the only acceptable one for a given structure while if it describes one and only one possible structure, it is also unambiguous. However, substructure searching, one of the main uses of a compound file, does not require a unique molecular description, so instead of using a canonical representation, any unambiguous representation for a structure, e.g., a connection table derived from an arbitrarily numbered structure diagram, may be used to represent a compound in the file. Registration then involves searching the file for an identical structure; the simpler process of a search for an identical representation is no longer sufficient since a given structure may be represented differently, but equally correctly, when subsequently presented to the system. The atom-by-atom matching[1,2] necessary to establish identity between two differently encoded structures is time-consuming and is too expensive if a large number of structures have to be considered for each registration. To minimize the number of structures which must undergo a detailed investigation, the file is first partitioned into small, nonoverlapping subgroups: only those compounds in the group to which the new compound belongs need to be searched. Registration will therefore become more efficient if the number of compounds in each group is decreased, as fewer atom-by-atom comparisons will be necessary.

On-line registration, carried out at a terminal from which the structure diagram of a compound can be input, requires rapid entry to a direct-access compound file which implies the availability of some form of search code or key, based upon the structure diagram, to obtain access to the collection. Several methods are available for searching disk-based files:[3] one of the simplest is hash-coding[4] in which a key, calculated from the input record, is used to address the disk directly rather than proceeding via some form of directory. Registration would then consist of: input of a structure diagram from a terminal; the automatic generation of a connection table[5] and calculation of the search key; hashing the key to obtain an address in the file; retrieval of all compounds matching the query and display of these compounds on a VDU screen at the terminal. Visual inspection, or atom-by-atom comparison, would then reveal whether the query molecule is already present in the file. To make such a system feasible the number of structures retrieved should be kept as low as possible.

The molecular formula is commonly used to partition the file initially, and this registration method has become known as the "isomer sort" technique; it was used by Ray and Kirsch in the first structure search system.[6] Bragg et al.[7] studied the distribution of compounds appearing in the *Chemical Abstracts* Sixth Collective Formula Index among molecular formula groups and found that the size of molecular formula groups, while highly variable, is also regular, and may be predicted with some accuracy. Methods are therefore needed to partition the larger groups to obtain subfiles small enough to permit atom-by-atom matching. Dyson[8] described a configurational index, based upon the IUPAC linear notation, which indicated the presence of various chemical fragments and could be used in conjunction with the molecular formula index. Shaw[9] found that large molecular formula groups could be partitioned reasonably effectively using a classification of structures based upon the environment of their constituent heteroatoms. Registration with the Mechanical Chemical Code (MCC)[10] was accomplished by a variation of the isomer sort technique; the Coded Molecular Formula (CMF), derived from a symbol count of the atom symbols in the MCC notation, was used instead of the molecular formula, giving smaller groups and hence a more efficient registration process. The extent to which the CMF can distinguish between compounds in the same molecular formula group is discussed by Lynch et al.[11] who characterized each structure in several large molecular formula groups by sets of small, bond-centered fragments generated by an analysis of its connection table. Similar work has been reported by Mishchenko.[12]

While a molecular formula gives a description of the numbers and types of atoms present in a molecule, it does not take into account the manner in which they are interconnected: if this information were available the discriminatory power of molecular formulas would be much increased. The work reported in this paper describes an attempt to provide this information by means of a topological index[13] which condenses the connectivity data present in an adjacency matrix to a single numerical identifier or expression which has (ideally) a different value for every structure.

2. TOPOLOGICAL INDEXES INVESTIGATED

Wilcox[14] has used a simple connectivity index as a measure of molecular branching while calculating molecular π-orbital

0095-2338/78/1618-0146$01.00/0