(5) Skolnik, H., and W. R. Payson, "A New Posting Method for the Preparation of a Cumulative List," *J. Chem. Doc.*, **3**, 21–24 (1963).
(6) Skolnik, H., and A. Clow, "A Notation System for Indexing Pesticides," *J. Chem. Doc.*, **4**, 221–227 (1964).
(7) Skolnik, H., and M. H. Payson, "Designing an Author-Based Correspondence Information System," *J. Chem. Doc.*, **6**, 240–244 (1966).
(8) Skolnik, H., "The Art and Science of Chemical Documentation," *Ind. Chim. Belge*, **32**, Special Suppl. No. 1, 100–102 (1967).
(9) Skolnik, H., and R. E. Curtiss, "An Evaluation of TEXT360 for Producing Reports," *J. Chem. Doc.*, **9**, 150–154 (1969).
(10) Skolnik, H., "Management of Operations and Services in the Hercules Technical Information Division," *J. Chem. Doc.*, **9**, 213–217 (1969).
(11) Skolnik, H., "A New Linear Notation System Based on Combinations of Carbon and Hydrogen," *J. Heterocycl. Chem.*, **6**, 689–695 (1969).
(12) Skolnik, H., "The Multiterm Index: A New Concept in Information Storage and Retrieval," *J. Chem. Doc.*, **10**, 81–84 (1970).
(13) Skolnik, H., and B. E. Clouser, "Designing an Information Awareness and Retrieval System for Chemical Propulsion Literature," *J. Chem. Doc.*, **11**, 39–43 (1971).
(14) Skolnik, H., and L. F. McBurney, "Technical Reports and Decision Making," *Chem. Tech.*, **1**, 82–85 (1971).
(15) Skolnik, H., and W. L. Jenkins, "Evaluation of the IBM Administrative Terminal System and Magnetic Tape Selectric Typewriter for Text Processing," *J. Chem. Doc.*, **11**, 170–173 (1971).
(16) Skolnik, H., "A Computerized System for Storing, Retrieving, and Correlating NMR Data," *Appl. Spectrosc.*, **26**, 173–182 (1972).
(17) Skolnik, H., "A Multitlingual Index via the Multiterm System," *J. Chem. Doc.*, **12**, 128–132 (1972).

On-Line Substructure Searching Using Fragment Code—A Proposal

MICHAEL E. D. KOENIG

Institute for Scientific Information, Philadelphia, Pennsylvania 19106

Received February 11, 1974

A file structure is proposed which makes use of a fragment code and patterns and "families" of patterns to accomplish on-line substructure searching.

Now that, thanks to reduced data storage and communication costs, the day of large scale on-line literature search systems is actually upon us—the adaptation of computerized chemical substructure searching to on-line techniques is of great importance.

This paper proposes a method of organizing fragment code representations of chemical structure so that on-line substructure searches can be accomplished with reasonable storage requirements and relatively responsive data manipulation times.

DISCUSSION

In order to make the exposition of this proposal as concrete as possible, it will be described in terms of Ringcode[1] (probably the most commonly used and widely known fragment code) as used in the literature services Ringdoc, Pestdoc, and Vetdoc, produced by Derwent Ltd.† Let us examine various approaches to organizing fragment code information on-line.

The most straightforward approach would be to treat each fragment or code punch as a separate entry point in an inverted file, but it is equally obvious that the number of postings per entry point would be rather formidable. In the Derwent literature services to date, there are nearly half a million abstracts with an average of about six Ringcode cards per posting, and about 25 punches per card. This amounts to about 75 million total Ringcode punch postings. File storage is obviously a nontrivial problem, but even more of a problem is the manipulation of the huge data sets that would be necessary. A typical search would require Boolean logic operations on numerous lists, frequently hundreds of thousands of items in size. Response time would be intolerably long.

A second approach would be to treat patterns as index terms, and to do substructure searching by searching the inverted file for an appropriately broad pattern, then bringing in the corresponding document records from some quickly accessible sequential file, and doing a bit search on-line of those records. Substantial storage must still be relatively quickly accessible for the sequential file. Such a system is also critically dependent upon the previous definition of an appropriate pattern. Patterns that are rather specific run the risk of being too narrow, and patterns that are more general run the risk of matching and pulling in very large portions of the total file, in effect, becoming sequential searches, with serious degradation of response time resulting.

A third approach, which I am proposing, is an extension of the second, or perhaps more accurately, a combination of the first and the second. It combines carrying both patterns and some individual punch postings in an on-line inverted file.

Let us first look at the most direct method of constructing such a system. Each pattern, with its associated document numbers, would be an entry in the inverted file. In addition, when each document was processed to see which patterns (if any) fit, that pattern which matched the greatest number of punches would be designated as the Most Complete Pattern (MCP). The punches not included in that pattern would be separately entered in the inverted file. This, plus a directory of what patterns each punch was to be found in, would in fact be a complete representation of all the Ringcode information in inverted file format.

Storage requirements are substantially reduced. Assuming two pattern hits per Ringcode card (allowing for overcoding, and for both general and specific patterns), and an average of six Ringcode cards per document, the pattern portions of the file requires approximately six million postings. Assuming an average of three punches per pattern card that are not included in the most complete pattern (MPC) for that card (many MCP's will of course be perfect fits, but one must allow for cases where no pattern is appropriate), the punch directory portion of the inverted file requires another nine million postings. The directory of

† Further information and specifications available from Derwent Publications, Ltd., Rochdale House, 128 Theobolds Rd., London, England.

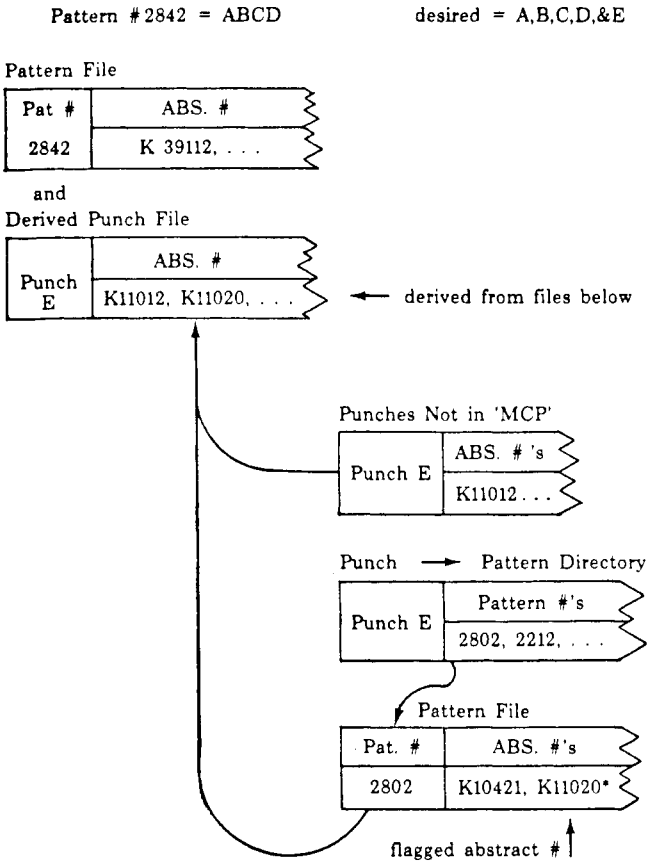

Figure 1

what punches appear on what cards will be comparatively trivial in size. Assuming 10,000 patterns to be generous, and an average of 25 punches per pattern, this results in only a quarter of a million postings.

The full Ringcode information then, for all the Derwent literature services to date, could be represented by approximately 15 million postings, not egregiously more than the six million required to carry merely patterns, but half an order of magnitude less than the 75 million required to carry the full Ringcode by method 1.

On-line manipulation, however, would be cumbersome and perhaps impracticable. Typically, a query would combine (and) a pattern with additional specific punches. There could, of course, be "or" relationships among those specific punches. The occurrences of the pattern could simply be obtained directly from the inverted file. The occurrences of each additional specific punch could be obtained in a two-step fashion. First, it could be obtained directly for those records in which the punch had occurred additionally to the most complete pattern. Second, in those cases where the query pattern was not the most complete pattern, the punch to pattern directory could direct us to which patterns contained that punch, and from those patterns we could be directed to specific punch cards (see Figure 1). It is important to note that we need not be concerned with all postings for any pattern, but for only those instances where the pattern was the most complete pattern (MCP). When the file is initially processed, any given punch will either be within the most complete pattern (MCP) or it will not be. If not, then it will be in the inverted "Punch File." If so, then if the pattern file flags those abstract numbers for which that pattern was the (MCP), then when we are constituting our "derived Punch File," we need only pay attention to those flagged abstracts.

An obvious difficulty with the above solutions is that the "derived punch files" will be substantial in size. Both deriving them and processing them will consume substantial amounts of CPU time. Response would not be immediate. It might be acceptable if such a system were the best we could achieve.

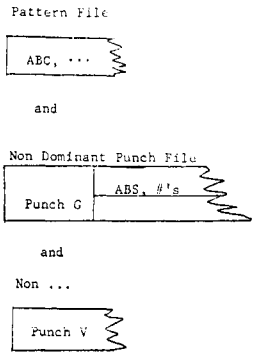

Figure 2. Logic using "nondominant punch file," rather than "punches not in MCP."

It is possible, however, to substantially reduce manipulation time and response time, but at the cost of more complex programming, and a more complex and extensive file structure—the classic tradeoff. This can be accomplished by taking advantage of the fact that many patterns are in fact members of a family of patents. That is, for example, the basic pattern for the penicillin family is a subset of the pattern for penicillin V. Typically, when multiple patterns apply to one record, there will be a familial or hierachial relationship among them. Let us define a subordinate pattern as one, all of whose punches are included in some other pattern, and a dominant pattern as one which is not a subordinate pattern, *i.e.*, is not a subset of some other pattern. Now, when processing a document, instead of defining a single most complete pattern, and making entries for all punches not in that pattern, the system would make an entry for each punch that was not part of any dominant pattern that was a match. Or, describing it in another fashion, an entry would be made for all punches that were not common to those dominant patterns that matched.

This could, of course, entail a substantial increase in the number of postings in the punch file—what was called the "punches not in MCP" file, and what will be called the "non-dominant punches file."

The attractiveness of this procedure is that we no longer need to build the derived punch file. Furthermore, we are manipulating and comparing files of much more manageable size. Substructure searching can be accomplished using only the pattern file and the nondominant punch file. The advantage of the concept of dominant patterns is that we can use general skeletal patterns that provide for power and flexibility in searching, while at the same time not having to carry as a separate posting every punch that occurs additionally to that skeletal pattern in each structural representation (see Figure 2).

We do, however, need a system that is cognizant of the relationship of pattern hierarchies and that can construct our query logic accordingly. To illustrate; suppose we have patterns

ABC		Penicillin nucleus
ABCDG	Let's say	Penicillin G
ABCEV		Penicillin V

and we are looking for *ABC* and G and V.

ABCGV exists, as does ABCDGV; both would be hits
ABCGV yields patterns *ABC* and punch postings G and V
ABCDGV yields patterns *ABC* (subordinate) and *ABCDG* (dominant) and punch posting V

A search logic of pattern *ABC* and punches G and V would hit ABCGV, but would miss ABCDGV, since G would not be carried as a punch posting.

What is needed to retrieve ABCDGV is an algorithm that would examine the query *ABC* and G and V, and in effect ask—are there patterns that combine *ABC* and G, or *ABC* and V, or *ABC* and G and V. If so, then these must be used to write an alternate query strategy. For example, since there is a pattern *ABCDG* that combines *ABC* and G, then we must also ask for *ABCDG* and V—which would retrieve ABCDGV. In effect the logic becomes

((*ABC* and G) or *ABCDG*) and V

We can also, of course, write logic that would exclude *ABCDG* and V if we are looking for *ABC* and G and V only.

A further reduction in search time, at the cost of increased storage, could be accomplished by expanding the "nondominant punch file" to carry an indication of what pattern caused the posting. The postings within the file would be ordered by what pattern caused the posting, and within that by abstract and card number order. The fact that several patterns (particularly in the case of overcoding) could match one card could now cause several postings to be necessary for what had been only one posting in the "nondominant punch file." This, however, could be substantially alleviated by posting only for the parent pattern (a subordinate pattern with no subordinates) of each family of patterns. This would result in only a trivial increase in storage.

In other words, instead of searching for *ABC* and any G and any V punch in the "nondominant punch file," we can now search for *ABC* and only those G and V postings in the "nondominant punch file" whose postings were caused by pattern *ABC*. The size of the files to be "*and*ed" and "*or*ed" is substantially reduced. The elaboration proposes that rather than posting the G in compound ABCDGV for both patterns *ABC* and *ABCDG*, it be posted for *ABC* only, thus saving storage. Rather than search for *ABCDG* and V's associated with that pattern only, one must search for *ABCDG* and any V's associated with *ABC* (see Figure 3).

Two questions that remain are: for what percentage of the file will there not be pattern cards, and how does one retrieve when there is no appropriate pattern? Using the Derwent provided patterns and *user* patterns, produced to expedite and simplify searches, with no eye to simplifying file structure, approximately some 75% of the Ringcode cards have a pattern match. It is our estimate that a reasonable number of appropriate patterns could account for much of the remainder, leaving less than 10% of nonpattern matches. That elicits the question of how to search that material. An obvious possibility is to post all punches when there is no pattern match. These could then be searched by a query that combined punch positions only and made no use of patterns (essentially defining a null pattern to be invoked when no other pattern fits).

The second question is how do we search the entire file when there is no appropriate pattern? For example, assume that we are looking for B and C and G and V. ABCGV and ABCDGV are still hits, but patterns ABC, ABCDG, etc., are now irrelevant. The answer is that we need an algorithm that will match the query against the profiles, in a similar fashion to the previously mentioned algorithm that matched subordinate patterns against dominant patterns. If a sufficient degree of fit were obtained, then a query could be derived in a fashion similar to the above. The tricky question would be to define the sufficient degree of fit. The more rigid the requirements, the less the processing time, but the greater the risk of not deriving some appropriate strategy. For example, the algorithm could require that 50% of our required punches appear in a pattern before using that pattern as the basis for a search. Thus we would search for $A\bar{B}\bar{C}$ and G and V, and $A\bar{B}\bar{C}D\bar{G}$ and V. The user at a terminal could define his algorithm, starting perhaps with a rather rigid fit, to ensure relatively prompt processing, then loosening it on the basis of his experience. The use of exclusions or negations (for example, B and C and G and V, but not D) would be very helpful, for they would legitimately exclude many possibilities and would permit us to allow a greater latitude in degree of fit.

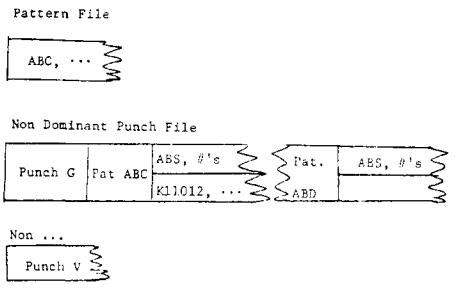

Figure 3. Nondominant punch file by pattern order.

A completely exhaustive search is of course possible, simply by examining all patterns, and deriving a separate search strategy for each pattern. For pattern ABC, it would be *ABC* and G and V, as above. For a pattern with no overlap, for example, DFHK, it would be DFHK and *B* and *C* and *G* and *V*. Again, the use of negations would be extremely powerful, and probably necessary to achieve a reasonable response time.

The handling of nonpattern queries then would be possible and practicable, if not so prompt as that for pattern based queries. It has been our experience that only approximately 15% of substructure queries make no use of a pattern, and even this percentage would decline if more patterns were used as proposed. For that percentage, we can probably accept a somewhat slower and more expensive response.

These added complexities require greater storage. The "nondominant punch file" will certainly be larger than the "punches not in MCP file." A reasonable estimate is that it will be between two and three times larger than the "punches not in MCP file." This would require then an additional 9 to 18 million postings, approximately doubling the 15 million postings estimated for the MCP approach, but it would still be only 40% of the storage required to individually post the punches.

The real advantage, of course, is that through the use of patterns and the maintenance of the relationship of what patterns, or at least what families of patterns were present when punches additional to those patterns occurred, the data sets to be manipulated and merged in an on-line search are drastically reduced. This, even more than reduced storage, is what can make on-line substructure searching feasible.

CONCLUSION

On-line substructure searching is feasible and practical. The "pattern phenomenon" makes fragment codes such as Ringcode particularly well suited for on-line applications. The work being done currently on the convertibility of various methods of structural representation (for example, Granito's work for ISI on the conversion of Wiswesser Line Nation to Ringcode *via* a connectivity table)[2,3] means not only that search systems can use various representation techniques, *i.e.*, WLN and Ringcode, but more importantly that a decision on structural representation need no longer be made as an either–or choice. A linear notation system or a connectivity table system can become a fragment code system for the purpose of on-line manipulation with WLN capability for further more detailed search.[4]

LITERATURE CITED

(1) Nubling, W., and Steidle, W., "The Documentationsring der Chemisch-pharmazeutischen Industrie; Aims and Methods," *Angew. Chem., Int. Ed. Engl.*, **9**, 596–8 (1970).

(2) Granito, C. E., Roberts, S., and Gibson, G. W., "The Conversion of Wiswesser Line Notation to Ring Codes. I. The Conversion of Ring Systems," *J. Chem. Doc.*, **12**, 190–196 (1972).

(3) Granito, C. E., "Chemtron and the Interconversion of Chemical Substructure Systems," *J. Chem. Doc.*, **13**, 72–74 (1973).

(4) Granito, C. E., Becker, G. T., Roberts, S., Wiswesser, W. J., and Windlinx, K. J. "Computer Generated Substructure Codes (Bit-Screens), *J. Chem. Doc.*, **11**, pp 106–110 (1971).

ALWIN—Algorithmic Wiswesser Notation System for Organic Compounds†

E. V. KRISHNAMURTHY,* P. V. SANKAR,** and S. KRISHNAN

Molecular Biophysics Unit, Indian Institute of Science, Bangalore 560012, India

Received April 14, 1973

ALWIN, a new chemical notation system for organic compounds, based on the Wiswesser Line Notation, is described. Procedures and rules are given for constructing ALWIN for acyclic structures and cyclic structures, *viz.*, benzene and its derivatives, monocyclic, bicyclic, polycyclic, perifused, spiro, bridged ring, and ring of rings systems. A new method called "tessellation" is introduced for the topological description of fused and spiro ring systems. Also new concepts are introduced for describing bridged ring and ring of rings systems.

The enormous increase in the number of organic compounds, natural and synthetic, especially the latter, has necessitated the development of nonconventional chemical codes for the representation of organic compounds.[1-3] This paper introduces a new chemical coding or notation system called ALWIN (Algorithmic Wiswesser Notation) which retains the salient features of WLN.[3] The logical and information structure of ALWIN has been designed to algorithmize efficiently the encoding and decoding procedures.

ALWIN differs from WLN in the following aspects.

(1) The acyclic structures in ALWIN can be represented in two equivalent forms: the parentheses form and the parentheses-free form.

The former is useful for human comprehension; here the parentheses differentiate the side group symbols from the main chain symbols. (Note that the symbol ampersand "&" plays a similar role in WLN.)

The parentheses-free form of ALWIN resembles the Polish notation (Korfhage[4] and Knuth[5]) and is obtained by dropping the parentheses; this is economical from the point of view of storage and computation.

(2) When certain chemical groups occur repeatedly in the chemical structures, it is possible to contact the corresponding notations. For this purpose, a set of new rules for multiplier contraction, based on the symmetry of the structures, has been developed.

(3) The contraction of notation can be effected in some cases using the well-known concept of factorization in algebra. Here, whenever certain sets of symbols occur repeatedly in a string of symbols, they are factored out much the same way as in algebra.

(4) A novel method of delineating the cyclic structures as a tessellation (or tile-filling on a floor) is used.

It is to be noted that WLN makes use of a Hamilton path[3] passing through the chemical graph, whenever it exists, or a minimal spanning tree for the topological description of the chemical graph. For large chemical graphs such a description is not desirable, since the existence and the choice of a Hamilton path in these graphs appear to be very difficult problems in graph theory.[6]

(5) A new set of rules for effecting the uniqueness of the notation in fused cyclic structures has been proposed.

(6) The main ring and the bridging branches of a bridged ring system are encoded separately, and these are then cited with suitable punctuation; this eliminates the necessity for the introduction of concepts, such as the "pseudo-bridges" used in WLN.

(7) Also distinction is made between the two different types of ring of rings systems, *viz.*, a ring system connected back to itself and another in which many rings are fused together, enclosing an inner ring which shares two or more edges with at least one of the outer rings.

(8) Several novel algorithms for processing the chemical graphs have been developed,[7] and these serve as the basis for encoding into the decoding from ALWIN.

(a) An algorithm, similar to the parsing techniques of the Polish notation, which rapidly decodes the parentheses-free ALWIN.

(b) A procedure to detect a set of chemically significant fundamental rings which constitute a fused ring system; this can in general find planar projections of chemical graphs (graphs of maximal valency four).

(c) Another procedure which differentiates the bridged ring systems from the ring of rings systems.

(9) A new set of rules for representing the stereoisomers has also been developed.[8]

GLOSSARY

ALWIN comprises a set of symbols and a set of rules for the encoding and decoding of organic chemical structures. The symbols are listed first, and the rules are presented with examples later.

† Contribution No. 46, Molecular Biophysics Unit, Indian Institute of Science, Bangalore, India.

* Author to whom correspondence should be addressed.

** Presently at Tata Institute of Fundamental Research, Bombay 400005.