used to adjust a probability or the empirical formula. During this process the probabilities for the functional groups can be adjusted and given a final value. When this section is complete, every functional group has an assigned probability or the indication "UNKNOWN". In the third section, the results are reported in two ways: sorted by probability in decreasing order and sorted alphabetically by group name. These sorted lists are put in an output file following a summary of the spectral information.

CONCLUSIONS

A number of advantages have been realized by following this approach. The most important, we believe, is that information about IR interpretation remains at a high level, in explicit form rather than being encoded in FORTRAN in a way which makes it difficult to extract as information. This approach has proved valuable for the continued evolution of the rules by our efforts and by others not directly involved in this project. The time and effort required to develop this language and programs have been recovered already in savings over what would have been required to develop and maintain a similar system with the rules embodied in FORTRAN. Execution time for compilation of 7800 lines of rules is about 8 min on a VAX 11/780, carried out as a batch process. Interpretation requires several seconds for a typical spectrum of 25–30 peaks, which is within the bounds considered acceptable for an interactive program.

It is hoped that this discussion will serve to stimulate others, who may develop information-intensive software, to consider creation of special purpose languages for building their systems.

The compiler, interpreter, and current set of IR rules are available for distribution from the Quantum Chemistry Program Exchange, Bloomington, IN.[8]

APPENDIX A

The BNF grammar for the CONCISE language includes the following special symbols: "$" indicates a comment—not part of the grammar, ' ' indicates a literal—a word in the language, [] indicates an option—usually a word which will make a statement read better but does not add to the information, ! indicates alternative choices of words or characters, = defines the symbol on the left as the phrase on the right, . ends a phrase, () encloses options in a statement, {} indicates multiple repeat allowed (e.g., 0, 1, or more).

To read this grammar begin at "START" (see Chart V). This is defined as 0, 1, or more "DTREE"s followed by the word "COMPLETE". A "DTREE" is defined next and represents a block of rules for a major group (e.g., ketone) and its subtypes (e.g., α-β unsaturated). A DTREE starts with a major group name, followed by its empirical formula and an index which is used by the compiler to test if the groups of rules are in order. The body of the rules are made up of one or more statements and is concluded with "END". Each of the terms not in single quotes is further defined in the grammar. All terms must ultimately stop at an expression in single quotes. N.B.: reading the grammar is not the recommended way to learn a language; it is for construction of the compiler.

REFERENCES AND NOTES

(1) (a) Woodruff, H. B.; Smith, G. M. *Anal. Chem.* **1980**, *52*, 2321. (b) Woodruff, H. B.; Smith, G. M. *Anal. Chim. Acta* **1981**, *133*, 545. (c) Tomellini, S. A.; Saperstein, D. D.; Stevenson, J. M.; Smith, G. M.; Woodruff, H. B.; Seelig, P. F. *Anal. Chem.* **1981**, *53*, 2367. (d) Woodruff, H. B. "Progress in Industrial Microbiology"; Bushell, M. E., Ed.; Elsevier: Amsterdam, 1983; Vol. 17, p 71.

(2) Munk, M. E.; Shelley, C. A.; Woodruff, H. B.; Trulson, M. O. *Fresenius' Z. Anal. Chem.* **1982**, *313*, 473.

(3) Corey, E. J.; Wipke, W. T.; Crammer, R. D.; Howe, J. W. *J. Am. Chem. Soc.* **1972**, *94*, 421.

(4) Wipke, W. T.; Gund, P. *J. Am. Chem. Soc.* **1974**, *96*, 229.

(5) Shortliffe, E. H.; Buchanan, B. G. *Math. Biosci.* **1975**, *23*, 351.

(6) Aho, A. V.; Ullman, J. D. "Principles of Compiler Design"; Addison Wesley: Reading, MA, 1977.

(7) Wirth, N. *Commun. ACM* **1977**, *20* (11), 822.

(8) Smith, G. M.; Woodruff, H. B. *QCPE* **1981**, *13*, 426.

NLM-CHEMSORT: An Algorithm and Computer Program for Sorting Chemical Names

JOAN BURNSIDE, PAUL N. CRAIG, and GERARD T. GUTHRIE*

National Library of Medicine, Bethesda, Maryland 20209

Received August 8, 1983

An algorithm is described that has been designed to sort medium-sized lists of chemical names, including common, generic, trivial, and systematic names and code numbers, into a logical sequence. It successfully sorted more than 99.5% of 3767 names in its first application. Minor revisions then resulted in more than 99.9% success with the same set of names. The algorithm generates an 80-character primary sort key (alphabetic characters only) and a 16-character secondary level sort key (alphanumeric characters). These sort keys are generated de novo from the name as needed and, thus, do not require increased permanent-storage costs. Sorting on the primary sort key (and secondary sort keys when identical primary keys exist) results in logical sequences of chemical names.

BACKGROUND

Since its inception in 1969, the Toxicology Information Program (TIP) at the National Library of Medicine (NLM) has built various on-line files, based on either bibliographic records or chemical substances.[1] For large files of chemicals such as CHEMLINE (>500 000 compounds), there is no need for a printed list of names, since these records can be best accessed randomly by on-line searching. But for smaller on-line files such as the Toxicology Data Bank (TDB),[1] which contains detailed records for some 4000 compounds, there is a recurring need for printing lists by chemical names.

Until recently, TIP scientists relied on printed lists that were sorted by standard computer programs. With the initiation of a collaborative effort between NLM and the National Toxicology Program (NTP) in 1979, the need for sorted chemical name listings increased. When the standard computer sort routine was used on these lists (ranging from 100 to 5000 names), the resulting indexes often separated related

n-Butyl chloride
o-Aminophenol
o-Xylene
ortho-Aminophenol
p-Xylene
Aminophylline
Butylene oxide
N-nitrosodimethylamine
N-Allyl-N-methylaniline
Nitromethane
Nitrosodimethylamine
Xylene
1,2-Dimethylbenzene
11-Hydroxystearic acid
2-Aminophenol

Figure 1. Names ordered by NLM standard sort. In the utility sort program at NLM, special characters precede lower case letters, which precede upper case letters; numbers sort last.

chemical names, thereby causing user frustration. The lists contained few systematic names and consisted mostly of trivial, generic, and trade names and numbers and, thus, were shorter, on the average, than systematic *Chemical Abstracts* names.

STATEMENT OF PROBLEM

When 100 or fewer chemicals are listed by name, as in tables, one can readily scan the lists prepared by a standard sorting routine without missing chemicals, even if they are scattering throughout the listing. When many hundreds or thousands of names are listed, one may fail to locate closely related substances or identical substances with synonymous names, because of the way they are dispersed throughout the listing due to various standard sort sequences (see Figure 1). This is especially true if one is not familiar with problems of chemical nomenclature and synonymy or with the rigorous sequencing produced by a standard sort.

The sorting of numbers can be especially confusing if large lists of code numbers are sorted by this routine. The numbers beginning with 1 are sorted before those beginning with 2, e.g., 1, 10, 111, 191, 2, etc. To avoid this problem, one must attach leading zeros such as 001, 010, 111, 191, 002, etc.; these now would sort properly (001 precedes 002, which precedes 010, etc.). A discussion of related problems caused by sorting chemical names was given by Swartzentruber.[2]

EXISTING SOLUTION

A solution, developed by Chemical Abstracts Service (CAS), is the assignment of sort keys and use of these sort keys with the standard sorting procedure rather than to sort on the name itself. This solution requires the storage of the nonprint sort key along with the name. A description of the development and use of these sort keys was presented by Flick.[3] Each introduction to the *Chemical Substance Index* of *Chemical Abstracts* (CA) contains a summary of the rules used in sorting CA names.[4] The CAS sort keys are about the same length as the chemical name, and their storage results in essentially doubling the size of the name record.

PROPOSED ALTERNATIVE SOLUTION

Shorter lists of common (nonsystematic) names do not demand as complex a solution as that required for handling an entire issue of CA. Therefore, the algorithm described in Figure 2 was developed to sort these relatively small lists of names. The sort key is a fixed length field of 96 characters that is (for processing purposes only) appended to the front of the original record containing the chemical name. The first 80 characters of the field comprise the primary sort key (level 1), and characters 81–96 comprise the secondary sort key (level 2). When generation of the sort key is completed, it is sorted as a unit by a standard sort program. Inevitably, it is recognized that, when such a simplistic algorithm is used, a small number of the names may appear out of the desired sequence; therefore, the following step is included to allow manual editing of the sort sequences, if desired.

```
A.  All special characters are converted to blanks and all alpha
    characters are converted to upper case.

B.  The name is divided into fragments using blanks as the delimiters.

C.  If only one fragment is found, it becomes the primary sort key.

D.  The first fragment is checked by a special routine developed for names
    of dyes.

E.  For all fragments (except in special case C above):

    E1 - all fragments of length one are dropped to second level.

    E2 - all fragments beginning with numeric characters are dropped to
         second level - if it also is the last fragment in the name, and is
         all numeric and less than 100, it is expanded to three characters
         by adding leading zero(s).

    E3 - Those fragments which are 2 to 7 characters long are checked against
         a list of greek letters, isomerism descriptors and other terms (see
         Figure 4).  If a match is found, then a pre-selected portion of the term is
         dropped into the secondary sort key; (e.g., DELTA→DEL, DEXTRO→D, etc).

    E4 - If a fragment does not meet any of these criteria it is dropped
         in toto into the next available slot in level 1.

F.       When the last fragment has been processed, the length of the primary
         sort key is checked.  If it is zero, the secondary sort key
         becomes the primary sort key.
```

Figure 2. NLM-CHEMSORT algorithm to generate sort keys.

Prefix	Use	Prefix	Use	Prefix	Use
DL	DL	VIC	VIC	THREO	THR
MU	MU	ALLO	ALL	TRANS	TRA
NU	NU	ANTI	ANT	UNSYM	UNS
XI	XI	ASYM	ASY	DEXTRO	D
AND	--	BETA	B	LAMBDA	LAM
ACI	ACI	ENDO	END	PSEUDO	PSE
CHI	CHI	HOMO	HOM	EPSILON	EPS
CIS	CIS	IOTA	IOT	UPSILON	UPS
EPI	EPI	LEVO	L		
ETA	ETA	MESO	MES		
EXO	EXO	META	M		
NOR	NOR	PARA	P		
PHI	PHI	PERI	PER		
PSI	PSI	RING	RIN		
RAC	RAC	TERT	TER		
RHO	RHO	ALPHA	A		
SEC	SEC	DELTA	D		
SYM	SYM	GAMMA	GAM		
SYN	SYN	KAPPA	KAP		
TAU	TAU	OMEGA	OME		
UNS	UNS	ORTHO	O		
VAN	VAN	SIGMA	SIG		
		THETA	THE		

Figure 3. Prefixes and their surrogates.

A second program replaces this 96-character field in the sorted list with a 10-character numeric field that contains a generated sequence number, with the 10th character as zero. This program then prints both the sort numbers and the names in the sorted sequence. The sequence numbers can then be modified for the small number of names that require relocation; the list is then re-sorted, and the sequence numbers are dropped, leaving (on final printing) a sorted listing in the desired sequence.

A list of commonly used character-string prefixes of from two to seven characters was developed, and all fields of two to seven characters in length were matched against this list (Figure 3). If a match was found, a preselected portion of the field was added to the secondary sort key. Thus, the 16 characters in the secondary sort key were not taken up by, for example, "alphaalphaalpha"; instead, "AAA" was employed. This procedure permits limiting the secondary sort key to 16 characters without losing desired discrimination. Additions to, or removals from, this list can be made as desired, based upon experience with the particular mix of chemical names involved.

Application of the algorithm to the names in Figure 1 resulted in the reordering shown in Figure 4. It should be emphasized that there is no one "correct" way to sort complex lists; there is a decidedly subjective component to this task.[5]

N-Allyl-N-methylaniline
O-Aminophenol
Ortho-Aminophenol
2-Aminophenol
Aminophylline
n-Butyl chloride
Butylene oxide
1,2-Dimethylbenzene
11-Hydroxystearic acid
Nitromethane
Nitrosodimethylamine
N-Nitrosodimethylamine
Xylene
o-Xylene
p-Xylene

Figure 4. Chemical names from Figure 1 after sorting by NLM-CHEMSORT.

RESULTS

The first test of the algorithm was to sort the name index to the FY 1982 *National Toxicology Program Annual Plan.*[6] This was accomplished manually by using the algorithm to prepare the two-level sort keys for the approximately 1450 names in the index. This process, although time consuming, was rewarding in that several unanticipated problems were encountered and the algorithm was subsequently modified to resolve them. Examples of the sort keys generated are shown in Figure 5.

The first computerized application was then made for sorting the chemical index to the FY 1982 *Review of DHHS, DOE and EPA Research Related to Toxicology*, published as part 2 of the NTP annual plan.[6] This task involved sorting 3767 names. A careful edit revealed 18 names for which the algorithm failed to provide a proper sort. This success rate (99.52%) far exceeded expectations. Because the output from the program includes a generated sequence number, ending with an added zero digit, it was a simple task to force these 18 names to sort where desired (by changing the sequence number to that of the name which should precede the desired name and adding a 1, 2, 3, etc. in place of the terminal zero digit). This manual editing is not difficult when files of this size are involved. Obviously, on extrapolation to a 10- or 100-fold level, the replacement of 180 or 1800 names would be much more time consuming.

The limitations were almost all due to names such as "N4-methyldeoxycytidine" and "O6-ethyldeoxyguanosine", both of which sorted under the leading letter (N and O, respectively). The algorithm would handle them properly if they read "N-4", etc., but as then formulated, "N4" and "O6" become part of the first level (primary) sort key. To resolve this minor problem, it was decided to drop any fragment that contained a number into the second level sort key. This adjustment resulted in the proper sorting of all but 2 out of 3767 names. It appears, therefore, that the intended goal to sort lists of up to several tens of thousands of names was met.

OPERATIONAL DETAILS

The NLM computer system used consists of two IBM 3033 units operating in the multiprocessor mode with 24 megabytes of main memory. The operating system is MVS/SP (1.3). The chemical name sort key generation program was tested in batch mode with a utility sort and a utility print program.

```
     Primary                           Secondary

 1.  ALLYLMETHYLANILINE......................NN
 2.  AMINOPHENOL.............................O
 3.  AMINOPHENOL.............................ORTHO
 4.  AMINOPHENOL.............................2
 5.  AMINOPHYLLINE...........................(none)
 6.  BUTYLCHLORIDE...........................N
 7.  BUTYLENEOXIDE...........................(none)
 8.  DIMETHYLBENZENE.........................12
 9.  HYDROXYSTEARICACID......................11
10.  NITROMETHANE............................(none)
11.  NITROSODIMETHYLAMINE....................(none)
12.  NITROSODIMETHYLAMINE....................N
13.  XYLENE..................................(none)
14.  XYLENE..................................O
15.  XYLENE..................................P
```

Figure 5. Sort keys developed for names in Figures 1 and 3.

On sorting an index of 2820 names, the chemical name sorting program took 1.68 cpu seconds. The utility sort and print run with the same data without the key generation program took 0.68 cpu second. On sorting the TDB index of 4059 names, the sort program took 2.27 cpu seconds; the utility sort and print run with the same data without the key generation program took 0.85 cpu second. All programs were written in PL-1. If sufficient interest is shown by others, the NLM-CHEMSORT software could be made available for public distribution via the National Technical Information Service.

ACKNOWLEDGMENT

We acknowledge helpful discussions with Jerry Edge, Dr. Tamas Doszkocs, and Dr. Henry Kissman.

REFERENCES AND NOTES

(1) Cosmides, G. J. In "Symposium on the Handling of Toxicological Information"; Cosmides, G. J., Ed.; National Technical Information Service, U.S. Department of Commerce: Springfield, VA, 1978; National Institutes of Health, Bethesda, MD, May, 1976, pp 21–28, PB283-164.

(2) Swartzentruber, P. E. "Report on the Fifteenth Chemical Abstracts Service Open Forum", Los Angeles, CA, March 30, 1971, and Columbus, OH, July 1971, pp 5–10.

(3) Flick, R. A., reference 2, pp 11–15.

(4) "Introduction to Chemical Substances Index to Chemical Abstracts"; American Chemical Society: Washington, D.C., 1979; Vol. 91, pp 2-I-4-I.

(5) Subsequent to the original submission of this paper (August 1983), a paper has appeared that takes a different approach to this problem (Sage, G. W.; La Macchia, A. B. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 183–186).

(6) "National Toxicology Program Annual Plan, FY 1982" and "Review of DHHS, DOE and EPA Research Related to Toxicology, Part 2, National Toxicology Program Annual Plan FY 1982 and FY 1983", available from Information Office, NTP, P.O. Box 12233, Research Triangle Park, NC 27709.