(11) The detailed algorithms of NEWMAN are described in Kao, J.; Watt, L. *Comput. Chem.*, in press.

(12) Nyburg, S. C. *Acta Crystallogr., Sect. B: Struct. Crystallogr. Cryst. Chem.* **1974**, *B30*, 251.

(13) The detailed algorithms of CDRAFT are described in Watt, L.; Kao, J. *Comput. Chem.*, in press.

(14) Houminer, Y.; Kao, J.; Seeman, J. I. *J. Chem. Soc., Chem. Commun.* **1984**, 1608.

Procedures for Sorting Chemical Names for *Chemical Abstracts*' Indexes

ALLEN C. ISENBERG, JOANN T. LEMASTERS, ABE F. MAXWELL, and GERALD G. VANDER STOUW*

Chemical Abstracts Service, Columbus, Ohio 43210

Received January 25, 1985

In the preparation of each *Chemical Substance Index* to *Chemical Abstracts* (CA), nearly three-quarters of a million chemical substance names must be sorted by computer program into an invariant order. This sorting is done on sortkeys that are generated from the character strings in the names and is done in a way that takes advantage of the data elements used by Chemical Abstracts Service (CAS) in preparing these names. The organization of CA index nomenclature and the rules used in sortkey generation are described.

INTRODUCTION

The *Chemical Substance Index* to *Chemical Abstracts* (CA) each year includes index entries that refer to nearly three-quarters of a million different chemical substances. These alphabetical indexes, which are published twice annually, are merged every 5 years into a collective index. The preparation of these volume and collective indexes requires that very large lists of chemical substance names be sorted into a consistent order, so that the user of the printed indexes can locate a substance of interest with confidence that it has been placed at the correct point in the index.

For many years the preparation of the CA indexes required the efforts of a group of clerical staff who devoted their time to sorting thousands of index entries typed on separate cards. Although manual sorting achieved remarkably consistent results, the rapid growth of the indexes during the 1950s and 1960s made maintaining the quality of these efforts increasingly difficult and expensive. Since the early 1970s, Chemical Abstracts Service (CAS) has used computer processing extensively in the preparation of its indexes.[1] These computer systems include programs that carry out, with no human intervention, the sorting of chemical names that was formerly done by hand. Recently published descriptions of two algorithms for sorting chemical names[2,3] prompt us to describe the procedures that CAS uses in sorting names.

DATA ELEMENT STRUCTURE FOR CHEMICAL NAMES

To appreciate the way CAS sorts chemical names, it is necessary to understand two general aspects of the sorting process: first, the way in which CAS constructs a chemical name from individual data elements and ranks these data elements for sorting purposes; second, the way in which data elements that occur at the same ranking level are sorted by the use of sortkeys. The first two sections of this paper discuss data elements and their utility in sorting; the last section describes the use of sortkeys.

CAS uses an extensive and rigorous set of rules for generating chemical names. These rules, which are applied by human nomenclature experts with extensive computer support, ensure that a given chemical substance can be found at a predictable place in the printed *Chemical Substance Index*.[4] The systematic names that result from these rules appear in the *Index* in an "inverted" form; i.e., that portion of the name that refers to a "parent" structure is given before the names of the structural fragments that are attached to that parent structure. Thus, for example, the name

2-Butenedioic acid, 2-butyl-

gives the parent name 2-Butenedioic acid before the name of the attached substituent represented by the string 2-butyl-. The corresponding "uninverted" name would be

2-butyl-2-butenedioic acid

In the inverted form of this name, the characters before the first comma (sometimes referred to as the "comma of inversion") constitute the data element known as the *heading parent*. This data element normally has one of three forms: (a) a molecular skeleton name such as Butene, to which is attached the name of the principal functional group if one is present (dioic acid in this instance); (b) a functional parent compound in which no skeleton is expressed, such as Carbonic acid; (c) a trivially named parent such as Phenol or Urea. The names of the attached substituents, such as 2-butyl-, are included in a separate *substituent* data element.

The *heading parent* and *substituent* are two of the data elements that CAS uses for chemical names; the others are described later in this section. These data elements are assigned by the nomenclature specialist when a name is prepared. As described in the next section of this paper, the data element identifications play an important role in the sorting programs. They are also important in formating names for the printed indexes. The formating programs use the data elements to determine, for example, that two names which sort together have identical heading parents; the heading parent then needs to be printed only in the first name and can be represented by a long dash in the second name. Similarly, if two esters of an acid sort together, the formating process will cause the name of the acid to appear only once, with the two esters identified under it. The data element identifiers do not themselves appear in CAS printed services or online files, however.

Frequently a name contains a character string that describes a derivative of the principal functional group, such as the ester of an acid or the oxime or hydrazone of a ketone. Thus, for example, if the above name were modified to

2-Butenedioic acid, 2-butyl-, dimethyl ester

the string dimethyl ester would constitute the *name modifi-*

0095-2338/85/1625-0410$01.50/0

cation data element. This data element may also contain other information as, for example, in the following complex name modification: butyl ester, ion(1−), compd. with ethanamine (1:1).

Another data element that frequently occurs in names is the *stereochemical descriptor*, which contains stereochemical or spatial information. In this example, the string (*E*)- describes the stereochemistry and completes the name:

2-Butenedioic acid, 2-butyl-, dimethyl ester, (*E*)-

It is typically formated after the name modification, if there is one, or after the substituent or heading parent. This data element may also contain data such as *trans*-, (*R*)-, and similar terms. (Certain configurational descriptors for stereoparents are expressed or implied in data elements other than the stereochemical descriptor.)

Two other data elements used in chemical names to differentiate between otherwise identical heading parents are the *line formula* and the *homograph definition*. The line formula differentiates between parents of different stoichiometric composition, such as line formulas $CrCl_2$ and $CrCl_3$ in the headings Chromium chloride ($CrCl_2$) and Chromium chloride ($CrCl_3$). The homograph definition distinguishes between heading parents having different meanings, such as alkaloid and mineral in the headings Serpentine (alkaloid) and Serpentine (mineral). Another type of sorting differentiation results from the use of *heading subdivisions* to organize index headings with large numbers of entries. Four types of subdivisions are used: (1) *Qualifiers* divide the heading into separate areas of study according to the nature of the topics discussed in the original document, such as properties and reactions. (2) *Categories* divide the heading into different types of chemical derivatives such as esters, oximes, and polymers. (3) Six chemical substance particle headings (e.g., Alpha particle and Proton) are divided with special radiation qualifiers, biological effects and chemical and physical effects. (4) Alloy categories base and nonbase are used at alloy headings. The effect of any of these subdivisions is to group related index entries that would not otherwise be sorted together. Within any of these subdivided headings, all of the rest of the sorting described below applies.

SORTING BASED ON THE STRUCTURE OF CHEMICAL NAMES

The basic order of priority among the data elements used in sorting CAS chemical substance names is heading parent > line formula > homograph definition > substituent > qualifier > category > name modification > stereochemical descriptor. The homograph definition and line formula serve primarily to resolve sorting of heading parents in those cases where they are applicable; that is, they are considered to be part of the heading parent for sorting. Thus, all names with the same heading parent are brought together by the sorting of that data element. Names having the same heading parent are then sorted on the other data elements present, on the basis of their respective priorities.

An important principle invoked in this data element based sorting is that of "nothing before something". The implication of this principle is that, for example, all of the names having the heading parent Benzenepropanal and no substituent will sort ahead of all those that have that same heading but have a substituent present. In practice, names are sorted on the basis of a single sort with a sortkey formulated from all of the separate data elements, rather than by consecutive sorts each using a separate data element. However, it is easier to understand the sort order by initially thinking of it as a series of individual sorts. Consider, for example, the list of names with the same heading parent 2-Cyclohexen-1-ol shown in Table I. These names are shown in the table with the data elements labeled. As can be seen, all of those with no substituent come before all those with a substituent; in turn, all of those in each group without a name modification come before those with a name modification.

Table I. Names with the Same Heading Parent 2-Cyclohexen-1-ol

parent	substituent	name modification	stereochemical descriptor
2-Cyclohexen-1-ol			
2-Cyclohexen-1-ol			(*S*)-
2-Cyclohexen-1-ol		acetate	
2-Cyclohexen-1-ol		acetate	(*S*)-
2-Cyclohexen-1-ol	1-methyl-4-(1-methylethenyl)-		
2-Cyclohexen-1-ol	1-methyl-4-(1-methylethenyl)-		(1*R*-*trans*)-
2-Cyclohexen-1-ol	1-methyl-4-(1-methylethenyl)-	benzoate	
2-Cyclohexen-1-ol	1-methyl-4-(1-methylethenyl)-	benzoate	(1*S*-*cis*)-

The following list further illustrates the principle. All of the names with the heading parent 2-Butanol are brought together, as are those with the heading parent 2-Butanone. In both cases, the principle of nothing before something places all of the entries without a substituent before those with a substituent. In turn, within each of these groupings the entries without a name modification occur before those with a name modification. In the case of the heading parent 2-Butanone, the entries without a substituent are subdivided by the categories oximes and hydrazones:

2-Butanol
2-Butanol, sodium salt
2-Butanol, 1-chloro-
2-Butanol, 4-(trimethylstannyl)-
1-Butanone, 1-phenyl-
2-Butanone, hydrazones
  dimethylhydrazone
2-Butanone, oximes
  O-methyloxime
  oxime
2-Butanone, 3-(4-acetylphenyl)-
2-Butanone, 3-ethoxy-1,1-dihydroxy-
2-Butanone, 3-ethoxy-1,1-dihydroxy-, oxime
Butanoyl chloride

USE OF SORTKEYS

Sorting at the same data element level is accomplished with sortkeys generated from the data in each data element rather than the data values themselves. Simple character by character sorting of the data elements will give a different order than the order desired for the index. The problem is particulary acute with chemical nomenclature, since chemical names often begin with numerical locants; a character-by-character sorting would give these locants undue importance compared to the alphabetic characters in the names. Consider, for example, the following list of heading parents. On the left they are listed in the order that would result from character by character sorting; on the right, the heading parents are listed in the desired order, based primarily on alphabetical sorting:

character by character	alphabetical
1-Butene-3-yne	2-Butene
2-Butene	2-Butene-1,4-diol
2-Butene-1,4-diol	2-Butene-2,3-diol
2-Butene-2,3-diol	1-Buten-3-yne

The alphabetical sorting is achieved by generating a sortkey from each heading parent on the left. The process of generating a sortkey first divides the data element into three fields. The first field contains all the Roman alphabetics in the data element in the order of their occurrence from left to right. The second field contains any locants (italic, Greek, or numeric) that precede the first Roman alphabetic character. The third field contains the remaining italic, Greek, or numeric locants. For the heading parents above, these fields are as follows:

name	field 1	field 2	field 3
1-Buten-3-yne	butenyne	1	3
2-Butene	butene	2	
2-Butene-1,4-diol	butenediol	2	1 4
2-Butene-2,3-diol	butenediol	2	2 3

These fields are then concatenated to generate a sortkey for each heading parent. Character by character comparison of the sortkeys follows. At a point of difference in the sortkeys, one name takes precedence over the others. Each field is followed by a blank, so that the rule of nothing before something may be used to give the desired sorting on the basis of these sortkeys.

The actual representation of the locants is more complex than is represented here. There are three types of locants, including italic, Greek, and numeric, as well as versions of each type modified by primes or other locants. Locants are sorted in the order italic > Greek > numeric. Different sort fields are generated depending on whether a locant is unmodified, e.g., 3, or modified, e.g., 3′ or 3a. The sort field also depends on the type of character that is the modifier. Locants can include primes, numerics, italics, Roman alphabetics, Greek characters, or combinations of characters such as 4′a, 4′a*H*, etc. Punctuation characters are used to identify certain types of locants, but these punctuation characters do not appear in the sortkey itself; for example, the Roman alphabetic a in 5a is recognized as part of a locant rather than as part of the basic alphabetic sort field because it is preceded by a numeric and followed by punctuation. Numeric locants or numeric locant modifiers are represented as binary numbers so that, for example, 2 will sort before 11. The various types of locants are modified through the addition of leading blanks, following blanks, and so on, in order to make them sort in the right order. Another technique is used in the case of the three abbreviations C.I., E.C., and U.S., which are expanded to COLOURINDEX, ENZYMECOMMISSION, and UNITEDSTATES, respectively, before the sortkeys are generated.

For the great majority of chemical names, the total sortkeys are generated by combining the sortkeys generated from the heading parent, substituent, and name modification sortkeys as shown above. Such names can be considered to belong to the broad category of "alphabetic" names, in which there are discrete strings of either alphabetic or locant characters, and the alphabetic characters take precedence for sorting as discussed above. There are two other cases in which somewhat different sortkey-generation rules apply; these are referred to as *numeric* and *alphanumeric* names. All names fall into one of the three categories: alphabetic, numeric, or alphanumeric.

Numeric names are those that consist of a heading parent made up entirely of numeric characters. For these names the sortkey generation routine converts the numeric into its corresponding binary representation; the binary number is preceded by two hexadecimal zeros so that numeric names will sort ahead of all other names.

Alphanumeric names consist of numeric characters and short alphabetic strings, which in some cases may be mixed directly with the numerics, as in A2C or A20/22. Use of the same sortkey generation routine used for most names on alphanumeric names would give undesired sorting results. It would treat strings such as those shown here as locants and thus would not give adequate weight to the alphabetic characters. There are two steps in handling an alphanumeric name: recognizing the names as alphanumeric and then generating the correct sortkey. Alphanumeric names are recognized with a few simple rules such as the following:

(1) The name contains a slash, as in A20/22.

(2) The name contains no string of four or more Roman alphabetics; thus, 2,4-DNP is recognized as alphanumeric.

(3) The name contains three consecutive numerics that are not directly followed by an element symbol; for example, ST-141 or US-238.

(4) The name consists of one character string that contains only Roman alphabetics and subscript numerics, for example J_2.

The sortkey for an alphanumeric name has a somewhat more complex structure than a usual sortkey, using five fields instead of three. These fields are (1) alphabetics, (2) numerics, (3) Greek characters, (4) number of alphabetics preceding the first nonzero numeric, and (5) number of zeros preceding the first nonzero numeric. The first character of an alphanumeric sortkey is a blank. Thus, all alphanumeric names sort before all alphabetic names but after pure numeric names.

A simple example of an alphanumeric sortkey is the following:

name	field 1	field 2	field 3	field 4	field 5
A-1540-BB9686-A	ABBA	15409686		1	0

It should be noted that this classification of alphanumeric names is used only for the purpose of sorting. Many names such as Freon 11, which might be considered as alphanumeric for other purposes, receive the alphabetic sortkey, since that sortkey will lead to their being sorted correctly.

The sortkeys described herein have been used by CAS for approximately 15 years with only minimal adjustment. As described, the algorithm is specifically designed for sorting chemical substance names appearing in the CAS indexes. It has, however, been slightly modified on occasion to accommodate special characteristics of listings of names from other sources. It is hoped that this overview will be helpful to users in understanding the organization of CAS indexes and valuable to those who may need to develop name sorting procedures for other applications.

REFERENCES AND NOTES

(1) Wilson, G. A.; Swartzentruber, P. E.; Flick, R. A. "Report on the Fifteenth Chemical Abstracts Service Open Forum", Los Angeles, CA, March 30, 1971, and Columbus, OH, July 1971, American Chemical Society: Washington, DC, 1971.

(2) Sage, G. W.; Lamacchia, A. B. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 183–186.

(3) Burnside, J.; Craig, P. N.; Guthrie, G. T. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 39–41.

(4) "Chemical Abstracts Index Guide—Appendix IV"; Chemical Abstracts Service: Columbus, OH, 1985.