(11) Anon., "A Method of Coding Chemicals for Correlation and Classification," National Academy of Sciences–National Research Council, Chemical-Biological Coordination Center, Washington, D. C.
(12) H. A. Geer, *et al.*, *J. Chem. Doc.*, **2**, 110 (1962).
(13) A. Opler and T. R. Norton, *Chem. Eng. News*, **34**, 2812 (1956).
(14) "Rules for I.U.P.A.C. Notation for Organic Chemistry," Longmans, Green and Co., London, 1961.
(15) W. J. Wiswesser, "A Line-Formula Chemical Notation," Thomas Y. Crowell Co., New York, N. Y., 1952, revised 1962.
(16) J. A. Silk, *J. Chem. Doc.*, **3**, 189 (1963).
(17) H. W. Hayward, U. S. Patent Office Research and Development Report, No. 21, 1961.
(18) E. Meyer and K. Wenke, *Nachr. Document*, **13**, 13 (1962).
(19) L. Spialter, *J. Am. Chem. Soc.*, **85**, 2012 (1963).
(20) D. Gould, *et al.*, *J. Chem. Doc.*, **5**, 24 (1965); see also *Chem. Eng. News*, **41**, No. 46, 5 (1963).
(21) D. J. Gluck, *J. Chem. Doc.*, **5**, 43 (1965); see also *Chem. Eng. News*, **41**, No. 49, 35 (1963).
(22) D. L. Ballard and F. Neeland, *J. Chem. Doc.*, **3**, 196 (1963).
(23) H. A. Geer and C. C. Howard, *ibid.*, **2**, 51 (1962).
(24) H. T. Bonnett and D. W. Calhoun, *ibid.*, **2**, 2 (1962).
(25) E. H. Sussenguth Jr., *ibid.*, **5**, 36 (1965).
(26) "Survey of Chemical Notation Systems," Publication 1150, National Academy of Sciences–National Research Council, Washington, D. C., 1964.
(27) J. Frome, *J. Chem. Doc.*, **4**, 43 (1964).

Syntactic Scanning of Chemical Information

PAUL L. FEHDER* and M. P. BARNETT**
Cooperative Computing Laboratory, Massachusetts Institute of Technology, Cambridge 39, Massachusetts
Received October 1, 1964

1. INTRODUCTION

Digital computers were designed originally to perform numerical computations, but they have been used increasingly during the last few years to deal also with nonnumerical processes in many fields of study. Nonnumerical work now accounts for a substantial portion of computer usage in many organizations, and it is likely to increase considerably in the future, with an attendant trend toward the design of computers to facilitate such work.

Several of the techniques of nonnumeric information processing are being used in various laboratories to deal with chemical problems. This present paper describes a particular technique, namely mechanized syntactic analysis, by reference to a chemical topic of an illustrative nature. This is done because the authors believe that future developments may make the technique of practical value in nonnumerical chemical information processing. It should be stressed, however, that the specific example of mechanized syntactic analysis which is described in this paper was chosen for explanatory purposes. It is not suggested as an application of immediate practical value.

In discussions of mechanized information processing, the term "syntax" is applied to linear notational systems, systems of nomenclature, stylized subsets of natural language, and other forms of alphabetized information, with a meaning that is covered by a slight extension of the following dictionary definition of the general usage of the term

> "The arrangement of words by which their connexion and relation in a sentence is shown"[1]

This definition may be modified, for use in discussions of information processing, to

> "The arrangement of substrings by which their connexion and relation in the string which they form is shown"

The term "string" is used for a sequence of characters, taken from a finite set of characters, such as a conventional alphabet, or the set of characters that can be typed on a given keyboard machine. The term "substring" is used for any string that forms part of a longer string. Notations that use subscripts and superscripts can be mapped into strings by the use of several simple conventions, and more elaborate two-dimensional notations can be linearized in rather less convenient ways.

Syntax description and syntactic analysis have been used extensively in connection with nonnumeric applications of computers. Early work of this type has been described.[2-5] Several methods of representing syntaxes are now used in mechanized information processing.[6] We use a method[4] that allows "generic names" to be given to the types of string to which frequent reference is made in an application that is under consideration, and allows the name for each of these types of string to be defined by reference to the generic names of the constituent substrings and to the ways in which these substrings can be arranged and varied. A syntax is defined, in this approach, by a set of sentences of a certain simple form that is described in section 2. The set of sentences is called a verbal definition table. The words that form the sentences are abbreviated, in accordance with certain conventions, for input to our computer programs. These

* Chemistry Department, California Institute of Technology, Pasadena, Calif.
** University of London, Institute of Computer Science, 44 Gordon Square, London, WC1, England.

programs can use any syntax which is provided in this way to analyze a string of characters that is provided as a further input. The set of abbreviated sentences is called a mnemonic definition table. The way in which the definitional sentences are constructed is explained in section 2, by reference to a simple chemical example. Section 3 deals with the way in which the results of a syntactic analysis are recorded by our programs.

It is customary in high speed computing to refer to programs, subroutines, and related entities by capitalized names which abbreviate or convey some impression to the programmer of the purpose which the entities serve. These names often are kept to six or fewer characters because they are used internally in computer operations, and for certain technical purposes this limitation is convenient. Very often the purport of the name gets lost in time, but the commitment of the name remains because of its use in programs and in documentation. We use some such names in our work simply as convenient labels, almost as trade names are sometimes used in discussions that pertain to the work of chemical industry. In particular, we make repeated use of the name SHADOW for our syntactic analysis subroutine. This name has no real significance other than as the label to which we have become committed in the past. We use it freely. An account of the subroutine is given in ref. 4.

The possible relevance of syntactic scanning to the mechanization of certain chemical information problems, and some future developments, are discussed briefly in section 4.

2. A SIMPLE DEFINITION TABLE

For explanatory purposes, we assume that the computer is to be used to determine whether or not a given IBM card is punched in Hollerith code with the representation of the structural formula of a saturated aliphatic hydrocarbon. We adopt a set of conventions that allows such a hydrocarbon formula to be represented in just the Hollerith character set. These conventions constitute a syntax and, in the discussion that follows, this syntax is defined in the style that the SHADOW subroutine uses to attempt the analysis of an arbitrary input string. Using this syntax, the SHADOW subroutine can complete the analysis of an input string successfully if, and only if, the input string represents the hydrocarbon formula in accordance with the conventions that have been adopted.

The Hollerith alphabet includes numerals, capital letters, the period, dash, and brackets, but when the contents of a punched card is printed, on say an IBM tabulator, the characters all appear with the same horizontal alignment. We linearize conventional representations of hydrocarbon formulas by raising numerical subscripts to be on a line with the letters that represent atoms. Thus, CH4 represents methane, and CH3CH-(CH3)CH3, CH(CH3)3, and CH3–CH(CH3)3 all represent isobutane.

Suppose first that the formulas of just unbranched saturated aliphatic hydrocarbons are to be considered. Suppose too that dots, dashes, and brackets are excluded, so that the only acceptable formulas contain no characters other than C, H, and the numerals, and are in fact CH4, CH3CH3, CH3CH2CH3, CH3CH2CH2CH3, etc. The syntax of unbranched saturated aliphatic hydrocarbon formulas could then be defined in conventional English by the set of sentences that follows:

Definition Table I

A methane molecule is "CH4".
A methyl group is "CH3".
A methylene group is "CH2".
An unbranched methylene chain is a sequence of one or more methylene groups.
An unbranched saturated aliphatic hydrocarbon molecule is either a methane molecule or a methyl group then possibly an unbranched methylene chain, then a methyl group.

The meaning of these sentences is obvious. They do, moreover, conform to the style of the verbal definitions that can be abbreviated to form mnemonic definitions for use by SHADOW. In these sentences, the nouns and noun phrases—methane molecule, methyl group, methylene group, unbranched methylene chain, unbranched saturated aliphatic hydrocarbon molecule—are generic names. It can be seen that each sentence consists of a simple subject, the verb "consists of" or "is," and a definitional expression. Each definitional expression may contain (i) reference strings displayed between quote marks; (ii) the names of certain types of characters, such as "letters" and "digits"; (iii) certain words such as "then," "either," "or," and phrases such as "a sequence of one or more"; and (iv) one or more generic names that are defined by the definition table. The words and phrases that form the items ii and iii are called meta-words and meta-phrases. A certain set of these meta-words and phrases can occur in the definition tables which SHADOW uses. They are discussed in detail in ref. 4. A comma is used to mark the end of the range of each "possibly" and "either... or..." construction that occurs before the end of a sentence in a definition table. When the SHADOW program deals with a construction of the form "a sequence of one or more X's," where X stands for a generic name, it searches for as many consecutive X's as there are in the input before proceeding with the next portion of the definition.

The verbal Definition Table I is converted to the corresponding SHADOW mnemonic definition table by a routine process of abbreviation, in accordance with some simple rules that have been described.[4] At present this conversion is performed manually, but it can be mechanized. The mnemonic definition table is punched in Hollerith cards when it has been constructed. These cards are read by a computer in which the instructions that form the SHADOW subroutine have been stored already. The computer then can read further Hollerith cards, and determine whether these latter cards are punched with strings of codes that constitute unbranched saturated aliphatic hydrocarbon formulas of the sort that the definition table describes.

In the Definition Table I, four subsidiary definitions are included, for direct or indirect use by the main definition of the over-all string that must be recognized.

The choice of substrings that are given generic names, to subdivide the over-all definition of a syntax into separate definitional sentences for use by SHADOW, is arbitrary. So is the order of the sentences. The generic names that are used also are arbitrary in just the same way as the symbols that are chosen to represent variables in the analysis of a mathematical problem.

The Definition Table I can be generalized to deal with wider classes of aliphatic formulas. A slight generalization, that allows brackets, and dots or dashes between chain units is given below.

Definition Table II

A methane molecule is "CH4".
A methyl group is "CH3".
A methylene group is "CH2".
A chain unit is either a methylene group or a bracketed chain, then possibly either "." or "-".
A bracketed chain consists of "("then a chain then")" then an integer.
An integer is a sequence of one or more digits.
A chain is a sequence of one or more chain units.
An unbranched saturated aliphatic hydrocarbon molecule is either a methane molecule or a methyl group then possibly either "." or "-",, then possibly a chain, then a methyl group.

In the last sentence, two commas occur together, since one is needed to end the range of the "either... or..." construction and one is needed to end the range of the "possibly..." construction. The definition table is said to be "recursive," since a chain unit is defined in terms of a chain unit, *via* the definitions of a bracketed chain and a chain, and this allows nests of brackets to occur within the formulas that fit the syntax which the table defines.

The principles that are used in the Definition Tables I and II can be used to build up longer definition tables for the formulas of aliphatic molecules in which branched chains, unsaturation, and substituents occur. Considerable variety is allowed in these formulas by some of the constructions which can appear in the definitional sentences. Thus, the "either... or..." construction is particularly useful when a single name is given to a variety of alternative formulations of a single unit, such as COOH, CO2H, CO·OH, HOCO, HOOC. The "possibly...," "sequence of any number...," "sequence of n or more...," and "sequence of n or fewer..." constructions are useful when a single name is to be given to compounds or structures in which certain symbols are optional, or in which variable numbers of some sub-item may occur. Further useful constructions that the programs allow in the definition tables have been described.[4] A definition table that applies to a fairly wide class of aliphatic compounds is given in the appendix to the present paper.

3. THE TRACE TABLE

The examples of the preceding section were developed by reference to the problem of testing whether or not the string that was punched in a given Hollerith card conformed to a given syntax. The SHADOW subroutine deals in general with input strings that may be up to several thousand characters long, and which are read from an input card deck or magnetic tape. The primary output of the subroutine is a table that refers to substrings by the use of "pointers," that is, the ordinal numbers of the first and last characters of these substrings, counting within the entire input string. The general nature of this "trace table" is illustrated by its contents when the Definition Table I is used to analyze the string CH3CH2-CH2CH3. The numbers in the trace table refer to the characters that form this input string, in a simple left to right enumeration, as

C	H	3	C	H	2	C	H	2	C	H	3
1	2	3	4	5	6	7	8	9	10	11	12

The trace table for this string, analyzed by the Definition Table I, would be printed by the computer in the verbalized form (Trace Table I), if it was required for demonstration purposes. In general, however, the trace table is formed in just an internal representation within the computer, by the SHADOW subroutine, and used by further programs, as described in section 4, and the results of these latter programs are printed. The SHADOW subroutine forms the trace table in successive words of a block of storage starting from a standard origin. When the SHADOW subroutine performs a syntactic analysis, it provides the calling program with the count of the number of lines in the trace table that is formed. This count gives an immediate indication of whether or not the input string fits the given syntax, since the count is greater than zero if the string does fit, and is zero if the string does not fit. The simple syntactic testing process that was used in section 2 to introduce the concept of definition tables thus involves testing if the trace table line count that is produced by SHADOW is nonzero.

Trace Table I

Generic name	Left pointer	Right pointer
Unbranched saturated aliphatic hydrocarbon molecule	1	12
Methyl group	1	3
Unbranched methylene chain	4	9
Methylene group	4	6
Methylene group	7	9
Methyl group	10	12

For a second example of a SHADOW trace table, we refer to the Definition Table II in section 2, and the linearization of the formula $CH_3(CH_2(CH_2CH_2)_2CH_2-CH_2)_3CH_3$. The characters in the linearized formula are enumerated from left to right as

C	H	3	(	C	H	2	(	C	H	2	C	H	2	)	2	C	H	2
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

C	H	2	)	3	C	H	3
20	21	22	23	24	25	26	27

It can be seen that each line of the trace table relates to a portion of the input that fits the definition of a generic Trace Table II is produced in this instance, and is printed in its verbalized form.

Trace Table II

Generic name	Left pointer	Right pointer
Unbranched saturated aliphatic hydrocarbon molecule	1	27
Methyl group	1	3
Chain	4	24
Chain unit	4	24
Bracketed chain	4	24
Chain	5	22
Chain unit	5	7
Methylene group	5	7
Chain unit	8	16
Bracketed chain	8	16
Chain	9	14
Chain unit	9	11
Methylene group	9	11
Chain unit	12	14
Methylene group	12	14
Integer	16	16
Chain unit	17	19
Methylene group	17	19
Chain unit	20	22
Methylene group	20	22
Integer	24	24
Methyl group	25	27

name in the definition table, and which is either (i) the same portion of the input as the preceding line, or (ii) a substring of the portion of the input that is identified by the preceding line, or (iii) a portion of the input that occurs entirely after the portion that is identified by the preceding line, and not necessarily contiguous to it.

In the internal representation of the trace table, each line is stored in three computer words. Each generic name is represented by the mnemonic which was used for it in the mnemonic definition table that served as actual computer input for the SHADOW subroutine.

When using the SHADOW subroutine it is possible to produce condensed trace tables that do not include references to substrings of specified generic names or any portions of these substrings. This provision is needed at times to prevent the trace tables becoming excessively long.

Some of the ways in which trace tables, and other equivalent representations of the results of mechanized syntactic analysis, may be used in chemical information processing are discussed in the next section.

4. SOME POSSIBLE APPLICATIONS IN CHEMISTRY

Syntactic scanning may be of use in chemical information processing in several connections. These include

(i) Scanning linearized chemical formulas to determine if they contain substructures that have been specified by a user of a retrieval system.

(ii) Scanning linearized chemical formulas that are expressed in one notation, as a preliminary to their conversion to another notation. This conversion may be from one standard notation to another standard notation, or from a nonstandard notation into a standard notation.

(iii) Scanning chemical formulas as a preliminary to their conversion into chemical names, and *vice versa*.

(iv) Scanning chemical formulas and names as a preliminary to their conversion into the code of a tape-controlled type-setting machine, to set linear or built-up structural formulas.

(v) Scanning chemical formulas as a preliminary to the mechanical production of lists of possible products that may be formed with specified reagents.

(vi) Scanning mixed verbal and notational reference material for relevance to specific topics, using criteria that involve the presence of certain combinations of words, roots, formulas, and portions of formulas with much greater flexibility than is now allowed by just key word searches.

(vii) Scanning stylized accounts of specialized topics, such as descriptions of properties and syntheses, as a preliminary to translation into other languages.

The trace table provides a very convenient method of storing the results of the syntactic scan for processing by further programs that would deal with these various applications. Some further comments on a few of these topics are provided in the paragraphs that follow.

To demonstrate the scanning of linearized structural formulas, the authors have written some SHADOW definition tables that deal with aliphatic compounds which contain a variety of substituent groups, and which are linearized and expressed in the Hollerith character set by use of a few simple conventions. These definition tables are of interest for demonstrational purposes to show how trace tables can be formed and used by further programs. The present scanning subroutines are too slow for the approach to be of immediate practical value. Some recent developments in the design of certain information processing equipment may enable scanning processes that use definition tables to operate at very much greater speeds. These developments are mentioned briefly at the end of the present section. They may provide a practical motivation for the construction of definition tables to describe the syntax of the chemical information that must be scanned in actual working situations of various types.

The definition tables which the authors have written for linearized aliphatic compounds require extension and modification for application to any practically useful computer input, for two reasons: The definitions do not cover a sufficient set of compounds, and the conventions which we use for the formulas and for their mapping into the Hollerith character set are simple and obvious and convenient for explanatory purposes but probably differ from the conventions that govern the files of chemical information which will be processed mechanically for practical purposes. The modification of our definition table to deal with the same set of compounds in another linearized notation would be quite easy. So would the generalization of the resulting table to deal with a larger set of compounds. One advantage of the definition table approach to syntax description is the ease with which individual definitions can be changed or added with incremental effort, to deal with a modified or extended syntax, without altering the definitions that relate to the part of the syntax that is unchanged.

The particular definition table which the authors use at present for exploratory and demonstrational purposes is given in the two Appendices I and II of this paper. As explained in the Introduction, it is customary in computing work to use short names for entities such as these and we use the name STRKTR for reference. This name was chosen because it has an immediate phonetic associa-

tion with chemical structure for members of the authors' laboratory, and it has the convenience for computer programming mentioned earlier of not exceeding six characters in length. The present version of the table is called STRKTR IV. It supercedes the STRKTR III definition table that was described previously.[7] The STRKTR IV definition table is listed in Appendix I in the precise form that is read by the computer. Appendix II verbalizes the mnemonic generic names that occur in the definition table. The notational conventions that the STRKTR IV definition table describes can be inferred from these Appendices I and II. The STRKTR IV definition table contains relatively few of the mnemonics that represent meta-words and meta-phases with which the SHADOW subroutine can deal. The words EITHER and OR have their usual meaning, and the word THEN is implied between every two mnemonic generic names that are separated by just a space. A construction of the form "ARBNO X" stands for "an arbitrary number of strings X," where, as explained in section 2, "arbitrary number" means "as many as there are in succession, or none if none are present." A comma delimits the range of each EITHER and ARBNO, unless the range ends at the period which terminates the definition. These matters are explained at greater length in ref. 4. A further explanation of a definition table that uses a variety of meta-words and meta-phases, and which deals with a stylized instruction language for a text editing problem, is given in ref. 8.

The STRKTR IV definition table actually corresponds to the "clerical notation system" (that is abbreviated to CNS hereafter), which one of us (P. L. F.) is using in some further studies that are reported separately.[9] A definition table that summarizes a notation, in the way that STRKTR IV summarizes the CNS conventions, can be used to analyze input strings that conform to the notation, for post-processing by subroutines that relate to specific types of application and which the user provides. A definition table, such as STRKTR IV, also can be augmented by further definitions which the user provides for specific applications. Both of these approaches can often be used for the same problem. Suppose, for example, that a succession of formulas are to be tested, to determine which of these are 1,2-dihalides. The STRKTR IV definition table can be used without modification to analyze the formulas syntactically, and a very short subroutine coded in FORTRAN to test the trace table for the presence of two or more HALOGN mnemonics, and, if these occur, to test if they relate to halogen atoms on adjacent carbon atoms. Alternatively, the characteristics of a 1,2-dihalide could be defined by a syntax that consists mostly of definitions which appear in STRKTR IV. The SHADOW subroutine allows an input string to be analyzed as an example of any of the types of string whose generic names occur in the definition table. A definition table thus can contain definitions of the syntaxes of several types of string that depend collaterally on the same set of subsidiary definitions, and which are used optionally to analyze an input string on different occasions.

Some further subroutines have been coded recently to facilitate the extraction of information from a trace table and the formation of output strings from input that has been analyzed syntactically by SHADOW.[10] These subroutines have been given the name SASP for convenient reference as this abbreviates the term "syntactically analyzed string processor." They could be used, for example, to construct lists of products that might be obtained from a given molecule under specified circumstances.

The scanning programs that are in use in the authors' laboratory have gone through several stages of evolution in recent years. Reference 4 deals with a version of the SHADOW subroutine and of some associated programs that are available from SHARE.[11] These programs were used in the authors' chemical scanning work. Some slight modifications of these programs were made to allow a succession of records on magnetic tape to be analyzed syntactically by use of a single control card. A further variation of the SHADOW subroutine has been used to scan chemical formulas that were punched on Flexowriter tape.

The design of syntactic scanning programs and their speed of operation are affected by the characteristics of the computers that are used. Modern high speed digital computers, though manufactured primarily for arithmetic work, do show a trend toward the inclusion of hardware features and instruction codes that facilitate syntactic analysis and allied operations. The authors believe, however, that machines of a rather different type, which involve symbol matching and gating operations rather than arithmetic, would be much more powerful for scanning work. A machine of this general type was developed some years ago for mechanical language translation.[12] The relationship between the form of definition table that is used with SHADOW subroutine and the representation of syntaxes for a hypothetical machine, that is rather analogous to the machine which has just been mentioned, has been discussed by one of us.[13]

In summary then, syntactic scanning provides a potential aid to the solution of several problems of chemical information processing that may become of more practical value when information processing equipment has evolved further in design and manufacture. The hardware developments that would make syntactic scanning more useful would probably enhance the value of the other information processing techniques that are also being mooted. There does seem to be scope however for further exploration of the syntactic approach, particularly since it may allow flexibility, nonuniqueness, change and evolution of input conventions, and an avoidance of some of the problems of standardization, agreement, conformity, and long term, if not permanent, commitments to decisions that are rather arbitrary.

Acknowledgment.—This work was supported by Grant GM 10-430 of the National Institute of General Medical Science, National Institutes of Health.

REFERENCES

(1) "The Oxford Universal Dictionary," Oxford University Press, New York, N. Y., 1955, p. 2114.
(2) J. Backus, "The Syntax and Semantics of the Proposed International Algebraic Language of the Zurich ACM-GAMM Conference," Proc. First Inst. Conf. Inf. Proc., UNESCO, Paris, 1960.
(3) E. T. Irons, *Comm. ACM*, 4, 51 (1961).

(4) M. P. Barnett and R. P. Futrelle, *ibid.*, **5**, 515 (1962).
(5) R. A. Brooker and D. J. Morris, *J. ACM*, **9**, 1 (1962).
(6) See, for example, *Comm. ACM*, **7**, 51–134 (1964).
(7) P. L. Fehder, "The Scanning of Chemical Formulas by Use of the SHADOW Subroutine, the STRKTR III System," Technical Note No. 8, Cooperative Computing Laboratory, Massachusetts Institute of Technology, Cambridge, Mass., 1962.
(8) M. P. Barnett and K. L. Kelley, *Am. Doc.*, **14**, 99 (1963).
(9) P. L. Fehder, "A System for Handling Chemical Information."
(10) M. J. Bailey, P. B. Burleson, and M. P. Barnett, *Comm. ACM*, **7**, 339 (1964).
(11) M. J. Bailey, M. P. Barnett, E. J. D. Carter, and R. P. Futrelle, "MWFSHDW 4-the SHADOW IV F System," SHARE Distribution No. 1401, 1963.
(12) G. W. King, *I.B.M. J. Res. Dev.*, **5**, 86 (1961).
(13) M. P. Barnett, "A Hypothetical Machine for Syntax Texts," Technical Note No. 18, Cooperative Computing Laboratory, Massachusetts Institute of Technology, Cambridge, Mass., 1962.

APPENDIX I

Definition Table. The STRKTR IV Table for CNS Notation

ONSNTN ARBNO LINK BNDMK, THEN TLINK BLANK.
LINK CATOM ARBNO SUBST, MAXNO 1 VHYDSP.
BNDMK EITHER SNGLMK OR DBLMK OR RESMK OR TPLMK.
TLINK TATOM ARBNO SUBST, MAXNO 1 VHYDSP. BLANK = =.
CATOM EITHER CARBON OR OXYGEN OR NITRGN OR SULFUR OR PHOSPS OR BROMIN OR CLORIN OR IODIN OR FLORIN.
VHYDSP EITHER TRHYD OR TWHYD OR SHYD.
SNGLMK =-=. DBLMK ===. TPLMK =+=. RESMK =*=.
CARBON =C=. OXYGEN =O=. NITRGN =N=.
SULFUR =S=. PHOSPS =P=.
BROMIN =E=. CLORIN =G=. IODIN =I=. FLORIN =F=.
TRHYD =H3=. TWHYD =H2=. SHYD =H=.
SUBST EITHER SABRCH OR SHTBCH OR TPSBNT OR BRANCH, MAXNO 1 MTPLNO.
MTPLNO EITHER MTWO OR MTHREE. MTWO =2=. MTHREE =3=.
SABRCH =(=BNDMK SATOM =)=.
SATOM EITHER OXYGEN OR BROMIN OR CLORIN OR IODIN OR FLORIN OR NITRGN OR SULFUR.
SHTBCH =(= EITHER BNDMK SBATOM VHYDSP OR PHENYL, =)=.
SBATOM EITHER CARBON OR OXYGEN OR NITRGN OR SULFUR OR PHOSPS.
PHENYL EITHER SPHENL OR NPHENL. NPHENL =-R=.
SPHENL =-R,= ARBNO SUBDNG =,=, THEN SUBDNG.
SUBDNG SNO =-= SUBST. SNO EITHER TWO OR THREE OR FOUR OR FIVE OR SIX.
TWO =2=. THREE =3=. FOUR =4=. FIVE =5=. SIX =6=.
TPSBNT =(= BNDMK IDENT =)=. IDENT EXACNO 2 NUMBER.
BRANCH =(= ARBNO BNKMK LINK, THEN MAXNO 1 BNDMK TLINK, =)=.

Although mnemonics can be chosen that have a fairly obvious meaning when a definition table is not too long, the restriction to six characters could reduce the mnemonics to being just an arbitrary code in definition tables of considerable length. If this were to become a practical difficulty it would be possible to write a simple program to translate definition tables that contained generic names of more than six characters into the abbreviated form that the scanning programs require.

APPENDIX II

CNSNTN	cns formula
LINK	chain link
BNDMK	bond
TLINK	terminal chain link
CATOM	chain atom
TATOM	terminal chain atom
VHYDSP	valence hydrogen
SNGLMK	single bond
DBLMK	double bond
TPLMK	triple bond
RESMK	resonant bond
CARBON	carbon
OXYGEN	oxygen
NITRGN	nitrogen
SULFUR	sulfur
PHOSPS	phosphorus
BROMIN	bromine
CLORIN	chlorine
IODIN	iodine
FLORIN	fluorine
TRHYD	trihydrogen
TWHYD	dihydrogen
SHYD	monohydrogen
SUBST	substituent
SABRCH	single atom branch
SATOM	terminal branch atom
SHTBCH	multi-atom branch
SBATOM	nonterminal branch atom
PHENYL	phenyl
NPHENL	unsubstituted phenyl
SPHENL	substituted phenyl
SUBDNG	substitution designation
SNO	substitution position
TWO	2-substitution mark
THREE	3-substitution mark
FOUR	4-substitution mark
FIVE	5-substitution mark
SIX	6-substitution mark
TPSBNT	tie point substituent
IDENT	identifier
BRANCH	multi-atom branch
MTPLNO	substituent multiplier
MTWO	two-multiplier
MTHREE	three-multiplier