and J. Note that the symmetry is now correctly perceived due to the 1547 ≠ 2057 tiebreak in row F.

The symmetry classification in row K is stable (recognized by being identical with previous classification in row I).

As described in the text, CANON continues by breaking the lowest tie (symmetry class 2, nitrogens) to produce 12 distinct labelings. Starting with the lowest labeled atom and branching to lower labeled atoms at forks in the structure, the unique SMILES, CCC(CO)CCC(CN)CN, is established by GENES.

REFERENCES AND NOTES

(1) Weininger, D. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31.
(2) Balaban, A. T. *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 334.
(3) Joachim, C.; Gasteiger, J. *Top. Curr. Chem.* **1987**, *74*, 93.
(4) Wipke, W. T.; Dyott, T. M. *J. Am. Chem. Soc.* **1974**, *96*, 4834.
(5) Morgan, H. L. *J. Chem. Doc.* **1965**, *5*, 107.
(6) Bersohn, M. *Comput. Chem.* **1987**, *2*, 113.
(7) Hagadone, T. R.; Howe, W. J. *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 182.
(8) Uchino, M. *J. Chem. Inf. Comput. Sci.* **1980**, *20*, 116.
(9) Freed, E. E. Harvey Mudd College, personal communication.
(10) Wenger, J. C.; Smith, D. H. *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 29.

Computer Translation of IUPAC Systematic Organic Chemical Nomenclature. 1. Introduction and Background to a Grammar-Based Approach

D. I. COOKE-FOX, G. H. KIRBY,* and J. D. RAYNER

Department of Computer Science, University of Hull, Hull HU6 7RX, England

Received May 6, 1988

Since Garfield's pioneering work over 25 years ago in the linguistic aspects of systematic chemical nomenclature, leading to an algorithm for translating chemical names to formulas, very few reports of grammar-based analysis of systematic chemical nomenclatures have appeared in the literature. These have applied only to a few specific classes of names. While the major abstracting services use automated methods to process chemical nomenclatures, the limited details that have been published point to ad hoc approaches based on dictionaries of morphemes. This paper introduces a series that covers in detail the various aspects of the application of grammar-based techniques to the recognition of IUPAC systematic chemical nomenclature and hence the translation of chemical names to structure diagrams. Some necessary elements of language and grammar are discussed here in the context of the automatic recognition of chemical nomenclature.

INTRODUCTION

There are three broad categories of chemical language by which structural information is represented and communicated. These are the nomenclatures used to name compounds, formulas and line notations used as shorthand representations of compounds, and structure diagrams used as the primary means of communication of structural information and compounds. Chemical structures are also represented by connection tables, which are used internally by most computer-based transformation techniques as a topological description of molecular structure. However, connection tables are rarely used for communication between people and are not regarded as languages. The translation or interconversion of these languages by automatic means is an important application of computer science to chemical structure representation and processing. A review with references to those interconversions that have been reported is given by Rush.[1]

Computer translation from and to a systematic nomenclature has received little attention, and a recent book[2] has said that existing programs are very large and complicated and will be successful in this translation in considerably less than 100% of cases. This situation is associated with the slowness with which systematic nomenclature, as typified by the schemes devised by the International Union of Pure and Applied Chemistry (IUPAC), is accepted and used, with the continuing use of much semisystematic and trivial nomenclature, and with the questionable need for fully systematic nomenclature as perceived by the chemical industry.[3] In the U.K., the Chemical Nomenclature Advisory Service of the Laboratory of the Government Chemist encourages the use of systematic nomenclature following the principles set by IUPAC and is prominent in advising European Commision Services on these matters. Egan[4] and Egan and Godly[5] have discussed some of the benefits of using IUPAC systematic nomenclature, while the issues and problems associated with the use of chemical nomenclature are covered in the book edited by Lees and Smith.[6]

Work supported by the Laboratory of the Government Chemist has been in progress in this department for some years to investigate the application of grammar-based techniques, as developed for compiling computer programming languages, to automatic name recognition and translation into structure diagrams. In this project attention has been paid to certain classes of compounds of industrial importance, including some cases of semisystematic and trivial nomenclature. A particular feature of the project has been the use of inexpensive and readily available computing facilities as exemplified by the IBM PC and compatible microcomputers. An outline of the project in its early stages has been published.[7]

The first step in the translation of chemical nomenclature by grammar-based techniques is to develop a grammar that formally describes the syntax of the nomenclature. From the grammar, a parser can be produced to recognize names that satisfy the grammar and to check the semantics, or meaning, of the names. Names that are syntactically correct may nevertheless be chemical nonsense. Only after satisfying semantic checking is an intermediate form of a name constructed, the concise connection table.[8] Further processing leads to representations suited to communication to other computer software or to the display of a structure diagram.

CHEMICAL NOMENCLATURES

Overview of Nomenclature Styles. An excellent review of the development of chemical nomenclature is given by Cahn and Dermer.[9] Following the Geneva Congress of 1892, the maintenance of the rules of chemical nomenclature was taken on by the International Union of Pure and Applied Chemistry (IUPAC), who published revisions to the rules of organic

0095-2338/89/1629-0101$01.50/0

chemical nomenclature in 1930, 1957, and again in 1979.[10]

The other internationally recognized scheme of nomenclature is that of Chemical Abstracts Service (CAS). The CAS development of a systematic method of naming compounds initially arose from the preparation of the first 10-year cumulative subject index. The CAS nomenclature is used to index *Chemical Abstracts*, and to that end CAS has dispensed with many trivial names.

Chemical nomenclatures can be divided into three parts: systematic, trivial, and semisystematic or semitrivial.

(1) *A systematic name* is composed wholly of syllables specially coined or selected to describe the structural features of the name, for example, pentane, where "pent" implies a structure of five atoms and "ane" implies a saturated carbon chain. By the use of systematic chemical nomenclature, it is possible to give a complete, unique, unambiguous representation of the structure of an organic chemical compound.

(2) *A trivial name* is a label for a compound and gives no information about the structure of the compound. Acetic acid is an example of a trivial name, corresponding to the systematic name ethanoic acid.

(3) *A semisystematic or semitrivial name* is part systematic and part trivial. Usually the parent structure is a trivial name that has been expanded by using systematic nomenclature, for example, 3-chlorophenol.

IUPAC has retained those trivial names that still have a significant use in industry and commerce and uses these trivial naming roots as the basis of semisystematic names. For example, the widespread use of the name acetic acid ensures that in the near future it is unlikely to be replaced in common usage by the systematic name ethanoic acid. Because in individual circumstances it is not always possible to agree on the most desirable type of name, there are cases where alternative names are equally correct according to IUPAC rules.

Chemical nomenclature has developed as a written, rather than a spoken language. Problems occur with oral communication, not only through the difficulty of pronouncing the long multisyllable systematic names but also through different names having the same sound, for example, fluorine, an inorganic gas, and fluorene, an organic compound.[11] However, there are few examples of two substances having the same written name.

It is interesting to note that, in the development of the rules from the Geneva Congress to the IUPAC rules of 1979, it is the syntax of the rules that has changed, rather than the morphology. This is not the way natural languages develop, where it is the morphology or word construction that changes most rapidly.[12] Even where radically new, more systematic, nomenclature systems have been proposed,[13,14] the familiar vocabulary and punctuation forms are largely retained.

Computer Processing of Chemical Nomenclatures. Around 1960, Garfield investigated general linguistic aspects of systematic nomenclature and developed a method for the calculation of molecular formulas based on the determination of significant parts of each chemical name. This was the first work on the direct conversion of names to formulas, and Garfield described a manual algorithm by which formulas could be formed from atom and bond information, derived in turn from a dictionary of name parts, or morphemes.[12,15] Garfield identified the commonly occurring groups of letters, termed "morphs", and proceeded to investigate which of these were significant and which were not. Those of significance, called "morphemes", were admitted to the vocabulary and included in the dictionary, while the others were discarded.

At this point, the morphemes were classified as denoting structure formation or structure modification. The recognition of structure-forming morphemes within a supplied name caused the addition of specific elements to the accumulating formula, while modifying morphemes caused adjustment up or down of particular element counts, typically of hydrogen. Hence, the molecular formula of the name could be computed. The only classification of morphemes in this work was as "former" or "modifier", and no formal syntax was developed to control the combinations of morphemes accepted. Names were analyzed from left to right, with the longest possible morpheme being matched at each stage, and ad hoc methods were used to handle the bracketing of substituents.

Chemical Abstracts Service is extensively involved with the automated computer handling of its own chemical nomenclature, designed to facilitate the indexing and retrieval of journal abstracts concerning chemical compounds. The introduction of computers to the CAS system in the early 1960s led to the development of well-defined translation procedures for converting CAS systematic names to atom–bond connection tables.[16] Computer programs based on these procedures were subsequently reported,[17] and these were used to validate the CAS index names contained in early records and to convert them to a common internal form for storage. These routines have since been extended to provide editing and verification facilities for newly appearing index names.[18]

The method used is essentially ad hoc in nature, but it can cope with a wide selection of CAS names due to the very tight rules of this nomenclature. It would be fair to say that some of the gradual adjustments to the CAS nomenclature, over successive indexing periods, have been due to the requirements of the developing nomenclature translation programs.

As with Garfield's work, the CAS procedures are based on dictionaries of basic word roots, and, once these are identified, their semantic data are again subject to essentially ad hoc treatment. The drawback to this approach is that special routines have to be invoked frequently, for instance, when bracketed, highly branched substituents are dealt with.[17]

The CAS nomenclature is fully systematic and related to that of IUPAC insofar as there are many terms in common between the two systems. However, the CAS nomenclature has very different rules for the relative ordering of terms within names, designed originally to aid the indexing and retrieval operations that are the main occupation of CAS. The appearance of the "heading parent"—the main indexing term—as the first part of the name when read from left to right, followed by a "substituent part" and finally an optional "name modification" prompts an apparently simple sequential algorithm for name processing.

In the report of the CAS conversion procedures,[17] a simple sequential example of processing is given, after which a number of "special characteristics" are introduced as needing extra processing. Of these, the occurrence of "complex radicals" within the substituent part caused problems through the multiple, bracketed nesting of terms. Such constructions are best defined and handled through recursive techniques that were not available at the time that the CAS procedures were first developed. In the IUPAC nomenclature of organic compounds, the parent clause appears in its "natural" place to the right of any substituents. Thus, if names are processed from right to left, as has been done in our work, the handling of substituents follows naturally, needing no special treatment.

The first use of a computerized grammar analysis process was by Elliott.[19] A set of programs was used that had been developed earlier to assist in the production of compilers for computer-programming languages. Elliott developed a context-free grammar of the type required by this system, which expressed the rules for constructing the names of simple hydrocarbons according to the IUPAC nomenclature. The grammar analysis produced a set of tables that, in conjunction with a dictionary of nomenclature terms, were used to drive a recognizer for names from this subset of the nomenclature.

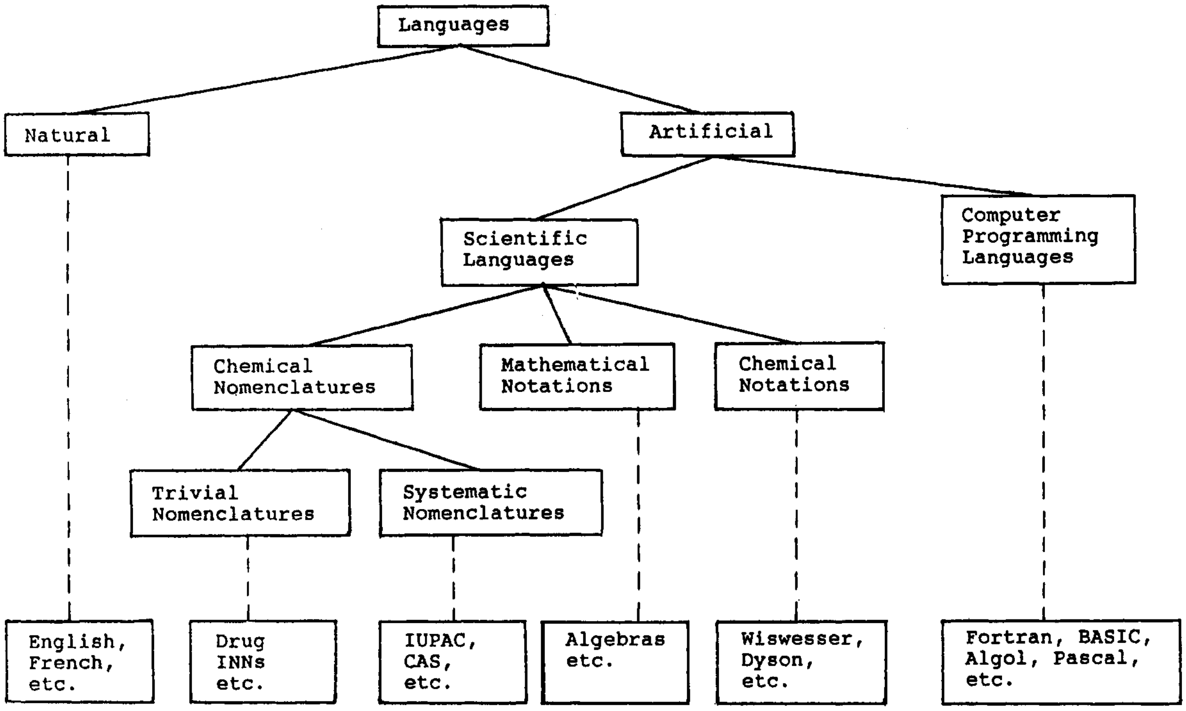

Figure 1. Place of chemical nomenclature within a classification of languages.

Not only was it possible with this scheme to extract structural information for the output of formulas, but due to the syntactic foundations of the work it was also possible to check the construction of input names and to handle bracketed and nested constructions within the same system. However, the scope of the program was limited in comparison with the broad range of constructions allowed in the IUPAC nomenclature.

Two other studies of nomenclature translation have been reported. Work by Stillwell[20] was limited to the area of steroid nomenclature, using a small dictionary of steroid nucleus names and common substituent terms. By means of ad hoc processing methods, a tabular representation of molecular geometry was formed for each input name, and diagrammatic output was then produced. Stillwell made no use of formal syntactic methods and concentrated on the stereochemical aspects of steroid structures in his output.

Carpenter[21] used syntactic methods to produce stick diagrams from nomenclature, via a connection table. As did Elliott earlier, Carpenter used a preexisting syntax-analysis package to process a grammar for simply branched hydrocarbons, and he also implemented the expansion of line formulas to names for the same grammar. This was possible since in this limited class of nomenclature it is not unreasonable to write bracketed formulas that contain the necessary structural information.

Carpenter considered that such translation schemes were of limited applicability owing to the use of trivial names and parents in IUPAC systematic nomenclature. The current project has demonstrated that the interpretation of semisystematic names is possible with grammar-based methods.

Grammars have been reported for other chemical languages. Chemical formulas can be recognized and parsed when presented in the single-line formats described by two published grammars.[22,23] GENSAL, developed for the Sheffield University Generic Chemical Structure Research Project,[24] is a specially devised formal language suited to use by chemists. It allows generic chemical structures to be described by the user and converted by computer to internal representations of structures that are used for searching databases. In the same project, a simple topological grammar has been devised with which chemical structure fragments, such as the alkyl, alkenyl, and alkynyl radicals, can be generated or recognized.[25] That paper reviews and demonstrates work toward grammars for the third category of chemical language, namely, structure diagrams.

LANGUAGES AND GRAMMARS

Natural and Artificial Languages. Languages can be broadly classified as natural or artificial in the manner of Figure 1. The natural languages are those developed by humanity over the millennia, which tend to be both ill-defined in any formal sense and subject to uncontrolled evolution in the introduction of fresh vocabulary, dialects, and so on. In contrast, artificial languages are those introduced or developed by interested parties to fulfil specific needs. Although in some cases they too may be not rigorously defined, and also subject to change, nevertheless, examples exist of artificial languages that are both well-defined and static. In Figure 1, the languages named are arranged in an approximate order of rigorous definition, with the fully natural languages on the left and the fully artificial on the right. Thus, we consider chemical notations such as Wiswesser line notation (WLN)[26] and chemical formulas to be more formally specified than mathematical notations whose evolution has been more "natural". The chemical nomenclatures are considered to be artificial languages, though at the natural extreme. They are by no means static and may be subject to changes of definition from time to time.

Phrase Structure Grammars. A language may be defined in terms of a grammar that is seen intuitively to have two components: a vocabulary of words from which sentences can be built and a set of rules that govern the juxtaposition of words in each sentence. The rules thus restrict syntactically valid sentences to a subset of all possible combinations of vocabulary words, but they have no control over the semantics of such sentences. A chemical name may be regarded as a sentence in one of the languages that are chemical nomenclatures.

It is a simple, though lengthy and tedious, task to draw up a vocabulary for any known language, but to express the rules of a grammar some additional material is necessary. Grammar rules are written not only in terms of the vocabulary words, or rather classifications of these called *terminal symbols*, but also in terms of the names given to particular phrases or constructions in the language, *the nonterminal symbols*. One particular phrase name, *the distinguished symbol*, denotes the overall unit, or sentence, of the language.

A *production* or *grammar rule* consists of a left part and a right part separated by a production symbol, which is commonly written as a right arrow (→), an equal sign (=), or two colons followed by an equal sign (::=). In the most general case both the left and right parts may contain terminal and nonterminal symbols. For certain types of grammar, the left part consists of just one nonterminal, which names the phrase, with on the right a list of terminals and nonterminals that define the components of the phrase. In this case, a sentence of the language defined by such a grammar may be generated by application of a sequence of production rules starting from the distinguished nonterminal symbol.

Chomsky Classification. Chomsky[27] defined four types of phrase structure grammars that may be used in the analysis of natural and artificial languages. Each type of grammar described a class of languages that is a proper subset of the class of languages defined by a lower numbered grammar type. That is, type 0 grammars are the most general and can define all the languages of type 1, type 2, and type 3 grammars, the last being the most restrictive. The type of a Chomsky phrase structure grammar is defined by the restrictions placed on the format of the production rules, which in turn restrict the language that can be described.

Chomsky type 3 grammars generate languages called finite state or regular languages. All type 3 grammars have production rules of just two forms. Either a single nonterminal symbol is rewritten as a single terminal symbol, or a single nonterminal symbol is rewritten as a single nonterminal symbol followed by a single terminal symbol, or it is rewritten as a single terminal symbol followed by a single nonterminal symbol. An example of a language described by a type 3 grammar would be where one or more a's can be followed by one or more b's, and the numbers of each may be different, for example

aabbb, aaaab, ...

Chomsky type 2 grammars are called context-free grammars, and they have production rules where a single nonterminal symbol is rewritten by a string of terminal and nonterminal symbols. There may be any number of symbols in the right part, with no restriction on the arrangement of the terminal and nonterminal symbols. An example language that may be described by a context-free grammar but not by a regular grammar is

ab, aabb, aaabbb, aaaabbbb, ...

It is not possible to describe this language with a regular grammar because of the symmetry around the junction point between the string of a's and b's. As indicated above, a regular grammar would only be able to describe a language where any number of a's are followed by an equal or different number of b's. The context-free grammar has a production rule whose right part has a nonterminal symbol sandwiched between two terminal symbols, which allows the grammar to define languages of this symmetric type.

Chomsky type 1 grammars are called context-sensitive grammars and have still weaker restrictions on the form the production rules may take. Both the left and right part of the production rules may have one or more symbols, but the left part must have at least one nonterminal symbol. An example of a context-sensitive language is

abc, aabbcc, aaabbbccc, ...

whose grammar is given by Cleaveland and Uzgalis.[28] This is not a context-free grammar, since we can describe a language with an equal number of a's and b's, but the additional restriction that there be the same number of c's cannot be described with a context-free grammar.

Chomsky type 0 grammars are called recursively enumerable and provide the most general description.

The relationship between the Chomsky classification of grammars and programming language definitions is well documented.[28,29] The context-free grammars are the most common grammars used in this connection. There is a type of recognizer algorithm for each type of phrase structure grammar, and recognizers for types 2 and 3 have been widely developed to compile computer programs.

Phrase Structure Grammars and Chemical Nomenclature. Chemical nomenclature is a language whose syntax has been described by context-free phrase structure grammars.[19,30,31] A recognizer may thus be written to recognize chemical names in the same way that a compiler recognizes valid computer programs. However, a pure recognizer that is able to show the conformity of a given input string to a given grammar is not sufficient, since it is also necessary to identify the precise detailed meaning of the name to produce an alternative, translated representation. Thus, the development of the grammar must also take into account not only the nature of the source language itself but the manner in which practical translation will occur during later stages of parsing and semantic processing. Assumptions and apparent restrictions introduced during grammar construction can often be checked and relaxed during these later stages of the overall translation process.

The vocabulary of chemical nomenclature consists of a number of punctuation symbols (commas, hyphens, and enclosing marks such as parentheses), integers, and strings of alphabetic characters. Much of the language of organic chemical nomenclature can be described by a regular grammar, but there are constructs that are not regular. For example, the normal nesting order of enclosing marks is { [( )] }, where parentheses are used first as the innermost marks and then brackets and braces, repeating the sequence if further nesting is needed.

The choice of enclosing mark is thus dependent on those used before, so the language is not even context free. However, if enclosing marks are restricted to just parentheses, their use is similar to the context-free language {ab, aabb, aaabbb, ...} described earlier, with the a's representing opening parentheses and b's closing.

Given this restriction, the language of organic chemical nomenclature can be represented by a context-free grammar. The grammar does not describe a fully artificial language, but rather a quasi-natural one that has developed by custom and practice over a period of time and is still developing. Thus, there is a need for the grammar to be easily modified to cope with future changes in nomenclature definitions. Significantly, chemical nomenclature differs in two ways from high-level programming languages, whose grammars generally are also context free.

First, it is readily apparent that chemical names (e.g., iododecane) have few if any clearly defined delimiting characters between the significant parts of the name (the "words" of the sentence), whereas in most programming languages the space is frequently used as a separator. This lack of delimiters, other than self-delimiting symbols such as hyphens, commas, and parentheses—as in 5,6-bis(2-iodopropyl)decane—requires a more complex method for recognizing name fragments and finding their terminal symbols, since it is generally necessary to establish the extent of the next fragment to be processed. For example, iododecane must be split into iodo - decane, not into io - dodecane.

Second, the semantically most significant and style-determining part of a chemical name is typically toward the right-hand end, whereas in most programming languages significant terms occur to the left of subordinates. For example, compare the substituted decane name given above with the Pascal statement below, where the **while** on the left determines the interpretation of what follows:

```
while ch <> '.' do read(ch);
```

The mapping of an essentially right-rooted nomenclature onto techniques devised for left-rooted programming languages is eased by the shortness of chemical names relative to the typical program. This allows the entire text of a name to be absorbed and processed from right to left, presenting an effectively left-rooted language to the translator.

DISCUSSION

IUPAC systematic chemical nomenclature is a language to which a grammar-based approach may be applied for the automatic recognition and translation of names. Structure diagrams are the primary means of communication of chemical structure information between chemists and a target into which other representations can usefully be translated. Since the structural meaning of the syllables that are used to form a systematic name and the rules that govern the assembly of the structural information into a complete structural representation are both known, then a structure diagram can be produced automatically from a systematic name. The structure diagram for a trivial name cannot be derived in this way, but can only be obtained by knowing the structure corresponding to each trivial name. The structure diagram of a semisystematic name can only be derived if the structure of the trivial part is known.

Subsequent papers in this series include the discussion of the development of a formal grammar for the IUPAC nomenclature;[32] the parsing, syntactic, and semantic analysis of systematic chemical names;[33] the generation of connection tables and display of structure diagrams; techniques for handling semisystematic and specialist nomenclature; and the correction of errors in systematic nomenclature.

ACKNOWLEDGMENT

We acknowledge the support provided for this project by the Laboratory of the Government Chemist and the particular help and advice provided by R. Lees, E. W. Godly, and other staff of its Chemical Nomenclature Advisory Service.

REFERENCES AND NOTES

(1) Rush, J. E. Computer Hardware and Software in Chemical Information Processing. *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 140–149.

(2) Gray, N. A. B. *Computer Assisted Structure Elucidation*; Wiley: New York, 1986; p 214.

(3) Silk, J. A. Realistic vs. Systematic Nomenclature. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 146–148.

(4) Egan, H. What's in a Name? *Chem. Br.* **1984**, *20*, 126–129.

(5) Egan, H.; Godly, E. W. Organisation and Chaos in the World of Nomenclature. *Chem. Br.* **1980**, *16*, 16–25.

(6) Lees, R, Smith, A. F., Eds. *Chemical Nomenclature Usage*; Ellis Horwood: Chichester, England, 1983.

(7) Cooke-Fox, D. I.; Kirby, G. H.; Rayner, J. D From Names to Diagrams—by Computer. *Chem. Br.* **1985**, *21*, 467–471.

(8) Rayner, J. D. A Concise Connection Table Based on Systematic Nomenclatural Terms. *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 108–111.

(9) Cahn, R. S.; Dermer, O. C. *Introduction to Chemical Nomenclature*, 5th Ed.; Butterworths: London, 1979.

(10) International Union of Pure and Applied Chemistry. *Nomenclature of Organic Chemistry, Sections A–F and H*; Pergamon: Oxford, U.K., 1979.

(11) Coyle, J. D.; Godly, E. W. *Chemical Nomenclature*; Open University: Milton Keynes, England, 1984; Chapter 7.

(12) Garfield, E. An Algorithm for Translating Chemical Names to Molecular Formulas. In *The Awards of Science and Other Essays*; ISI Press: Philadelphia, 1985; p 453.

(13) Lozac'h, N.; Goodson, A. L.; Powell, W. H. Nodel Nomenclature—General Principles. *Angew. Chem., Int. Ed. Engl.* **1979**, *18*, 887–889.

(14) Hirayama, K. *The HIRN System. Nomenclature of Organic Chemistry, Principles*; Maruzen: Tokyo, 1984.

(15) Garfield, E. An Algorithm for Translating Chemical Names to Molecular Formulas. *J. Chem. Doc.* **1962**, *2*, 177–179.

(16) Vander Stouw, G. G.; Naznitsky, I.; Rush, J. E. Procedures for Converting Systematic Chemical Names of Organic Compounds to Atom Bond Connection Tables. *J. Chem. Doc.* **1967**, *7*, 165–169.

(17) Vander Stouw, G. G.; Elliott, P. M.; Isenberg, A. C. Automatic Conversion of Chemical Substance Names to Atom Bond Connection Tables. *J. Chem. Doc.* **1974**, *14*, 185–193.

(18) Vander Stouw, G. G. Computer Programs for Editing and Validation of Chemical Names. *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 232–236.

(19) Elliott, P. M. Translation of Chemical Nomenclature by Syntax Controlled Techniques. M.Sc. Thesis, The Ohio State University, Columbus, OH, 1969.

(20) Stillwell, R. W. Computer Translation of Systematic Chemical Names to Structural Formulas—Steroids. *J. Chem. Doc.* **1973**, *13*, 107–109.

(21) Carpenter, N. Syntax Directed Translation of Organic Chemical Formulae into their 2-D Representation. *Comput. Chem.* **1975**, *1*, 25–28.

(22) Barker, P. G. Syntactic Definition and Parsing of Molecular Formulae: Part 1. Initial Syntax Definition and Parser Implementation. *Comput. J.* **1975**, *18*, 355–359.

(23) Kirby, G. H.; Milward, S. Syntax to Facilitate the Word Processing of Chemical Formulas. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 57–60.

(24) Barnard, J. M.; Lynch, M. F.; Welford, S. M. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 2. GENSAL, a Formal Language for the Description of Generic Chemical Structures. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 151–161.

(25) Welford, S. M.; Lynch, M. F.; Barnard, J. M. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 3. Chemical Grammars and Their Role in the Manipulation of Chemical Structures. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 161–168.

(26) Smith, E. J. *Wiswesser Line Formula Chemical Notation*; McGraw-Hill: New York, 1968.

(27) Chomsky, N. Three Models for the Description of Language. *IRE Trans. Inf. Theory.* **1956**, *2*, 113–124.

(28) Cleaveland, J. C.; Uzgalis, R. C. *Grammars for Programming Languages*; Elsevier: New York, 1977; p 18.

(29) Hopcroft, J. E.; Ullman, J. D. *Introduction to Automata Theory, Languages and Computation*; Addison-Wesley: Reading, MA, 1979.

(30) Rayner, J. D. Grammar Based Analysis by Computer of the IUPAC Systematic Chemical Nomenclature. Ph.D. Thesis, University of Hull, Hull, England, 1983.

(31) Cooke-Fox, D. I. Computer Translation of IUPAC Organic Chemical Nomenclature to Structure Diagrams. Ph.D. Thesis, University of Hull, Hull, England, 1987.

(32) Cooke-Fox, D. I.; Kirby, G. H.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Chemical Nomenclature. 2. Development of a Formal Grammar. *J. Chem. Inf. Comput. Sci.* (second of three papers in this issue).

(33) Cooke-Fox, D. I.; Kirby, G. H.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Chemical Nomenclature. 3. Syntax Analysis and Semantic Processing. *J. Chem. Inf. Comput. Sci.* (third of three papers in this issue).