Valid Structure Diagrams and Chemical Gibberish*

STEPHEN J. TAUBER**
Informatics Inc.
6000 Executive Blvd.
Rockville, Maryland 20852

KIRK RANKIN
National Bureau of Standards
Washington, D. C. 20234

Received December 3, 1971

Chemical structure diagrams are considered as utterances in a written language. Two types of grammars are considered for this language: topological grammars which emphasize the connectivity of the diagrams and geometric grammars which emphasize the arrangement of the diagrams in a plane. The vocabulary associated with each grammar consists of entities like atomic symbols, numeric subscripts, and bond symbols. Examples are given of initialization, propagation, and terminalization rules from each type of grammar and of the application of such rules. The value of grammars for the validation of computer input and for generation of computer output are indicated. The hypothesis is presented that compact computer storage may become accessible via grammars.

It has become well recognized that chemical structure diagrams are the natural language of chemists.[1-3] In an earlier paper,[4] we indicated that this language is susceptible to formal linguistic analysis analogous to a methodology applicable to languages such as English, Fortran, and algebraic notation. The key concepts are that utterances in a written language consist of strings of symbols from a given vocabulary, that a distinction can be made formally between grammatical and ungrammatical utterances, and that a successful grammar highlights natural groups within grammatical utterances—e.g., subject and predicate in English, and alkyl group and carbonyl group in chemical notation.

The objective of the linguistic study of chemical structure diagrams is to discover the regularities which are inherent in them and to construct a set of formal rules which will characterize these regularities. Possible applications of such a study and related activities are also discussed in this paper.

GRAMMARS FOR CHEMICAL STRUCTURE DIAGRAMS

We present two distinct approaches to constructing grammars which can account for chemical structure diagrams. One approach views the diagram as a topological entity with the characteristics of a graph, the other deals directly with the geometric arrangement of the diagram in the plane.

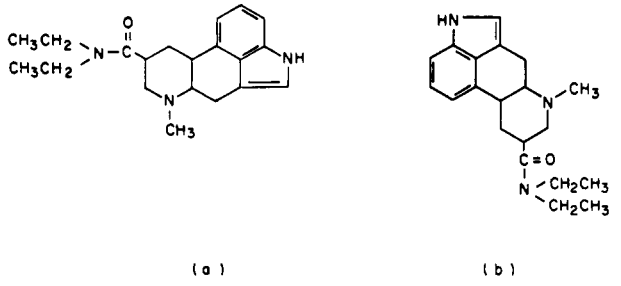
Topological Grammars. The usual structure diagram used for identifying a chemical species ignores the actual physical dimensions such as bond lengths and angles. Yet when the chemical structure diagram is considered as a graph, it is viewed as an abstraction from sets of atoms connected by bonds, and thus chemistry is to that extent emphasized by the diagrams.

By dealing strictly with the graph underlying a structure diagram a topological grammar leaves for separate treatment the actual display of the diagrams. The strength of this approach lies in the fact that it ignores pictorial considerations as it emphasizes structure. It thus accommodates the chemist's notion that—e.g., representations (a) and (b) in Figure 1 are entirely equivalent despite the pictorial facts that the ring systems are oriented differently by 120°, that the angles in the pentagon representing the five-membered ring differ, and that symbols for the amide group are differently laid out. The topological approach is powerful in that it groups equivalent structure diagrams such as (a) and (b) in Figure 1 together. To complete the transition from abstract graph to displayed structure diagram, another grammar component, is required, as discussed in the section on Structure Diagram Display.

A grammar consists of vocabulary symbols plus a set of replacement rules each of the form

$$L \rightarrow R$$

which states that if L occurs, then it may be replaced by R.



(a) (b)

Figure 1. Two distinct structure diagrams which are chemically equivalent

*Presented at the Symposium on Input-Output Interaction of the Chemist with the Computer, 161st Meeting, ACS, Los Angeles, Calif., March 31-April 1, 1971.

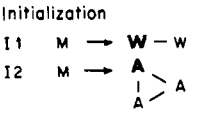
**To whom correspondence should be addressed.

Table I. Terminal Vocabulary for Chemical Structure Diagrams

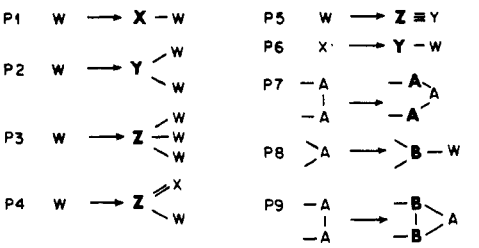
Atomic Symbols	H, He, Li, . . . , No, Lw
Numeric Subscripts	2,3,4,5, . . .
Bond Symbols	—, =, ≡

Table II. Examples of Replacement Rules for a Topological Grammar

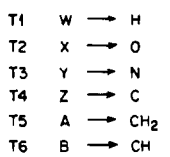
Initialization



Propagation



Terminalization



For our present purposes, we may take the vocabulary as consisting of two kinds of symbols—terminal and nonterminal. The terminal symbols are the atomic symbols, numeric subscripts, and bond symbols (see Table I). A more complete terminal vocabulary would have to include symbols for charge, an unpaired electron, implied ring carbon, etc.

The replacement rules are of three types: initialization rules to begin from a common origin, propagation rules to generate the underlying grammatical structure, and terminalization rules to convert everything to the terminal symbols of the language. (Some terminal vocabulary symbols may also be introduced by propagation rules.) Examples of replacement rules in a topological grammar are presented in Table II. In the notation used, the characters shown in boldface on the right-hand side are to be placed so as to be incident upon the same bond symbol or symbols the characters on the left-hand side were incident upon; boldface characters are otherwise entirely equivalent to ordinary characters. The initialization rules declare as it were, "Let there be a molecule." Rule I1 causes the generation of an acyclic structure diagram to be initiated; rule I2 leads to a cyclic structure diagram. The nonterminal symbols, W, X, Y, Z, A, B, permit distinction among different types of nodes. They may be thought of roughly as representing valence classes of atoms and moieties and cyclic or acylic units. Thus, the various rules in which Y appears represent the various possible situations for trivalent atoms and moieties. In these propagation rules, the terminal bond symbols are introduced directly, without subsequent bond terminalization rules required. It is important to note that no orientation is implied in any of the bond symbols. Thus, rule P2 could equally well have been written in any of the forms in Table III.

Such a grammar operates to generate a derivation of a chemical graph by successive applications of the rules. A simple illustration of a derivation is given in Table IV. On successive lines are shown the stages of the derivation on the left, and on the right, the specific rule which is applied in order to reach that stage from the preceding one.

Geometric Grammars. An alternative approach to the linguistic analysis of chemical structure diagrams is to consider them as entities embedded in the Euclidean plane, each symbol having a specific position. The plane is considered as segmented into polygonal regions. We have here chosen to use segmentation into rectangles, but the discussion would be very similar if hexagons or triangles were used. Indeed, not all of the polygons need be congruent, as long as an infinite segmentation is established. Fuller discussion of this point is left for elsewhere.[5] By considering the actual graphic layout, the chemical equivalence of alternative representations as in Figure 1 is ignored. However, this approach includes within itself an accounting for the display properties of the structure diagrams.

A grammar which is based on the geometric approach is comparable to one based on the topological approach in the sense that there are initiation, propagation, and termination rules and that a rule applies if the left-hand side matches a subpattern which occurs, and that that subpattern may then be replaced by the right-hand subpattern. The specific rules are however very different, as shown in Table V.

Explicit recognition is made of the segmentation of the plane; the "windows" on the left side and on the right side of each rule always have precisely the same arrangement of "panes." The empty panes are as much part of the subpatterns as are the labeled panes. These are particularly significant in the technique for avoiding overlap of distinct pieces of the structure diagram. Since this grammar ex-

Table III. Alternate Presentations of a Single Topological Replacement Rule

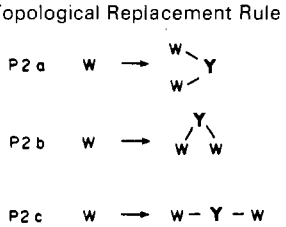
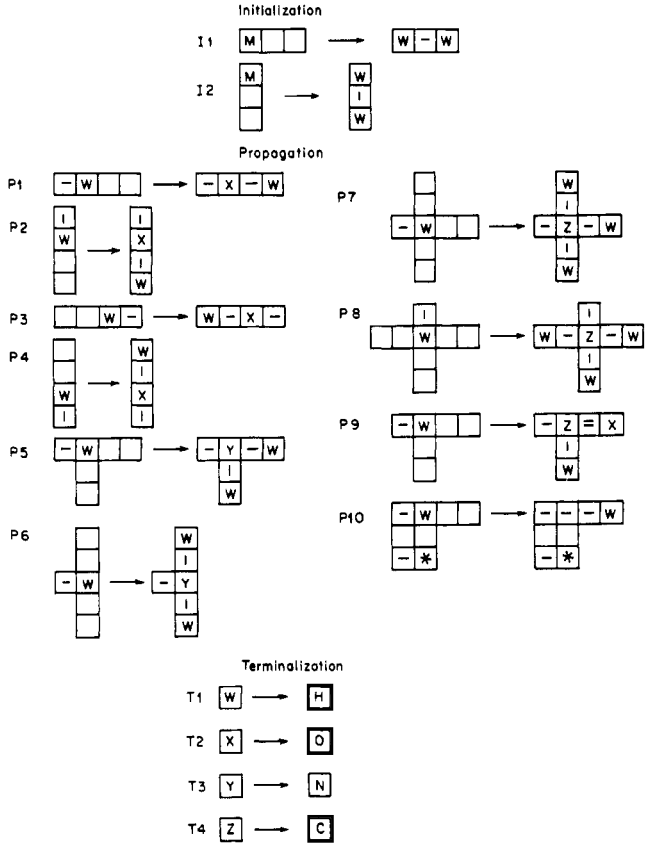


Table IV. A Sample Derivation via a Topological Grammar

Stage of Derivation	Rule Applied
M	(initial symbol)
W—W	I1
W—Z=X | W	P4
W—Z=Y—W | W	P6
W—Z=Y—X—W | W	P1
W—Z=Y—X—H | W	T1
H—C=N—O—H | H	T2,T3,T4,T1,T1

Table V. Examples of Replacement Rules for a Geometric Grammar



plicitly deals with arrangement in the plane, distinct rules are necessary for the several possible orientations, as for example rules P1 to P4 in Table V, which all deal with what chemically might be considered as elaboration of a univalent moiety in the same fashion.

In this grammar, control of overlap is achieved by the device of bond lengthening, similar to that described by Zimmerman and Lefkovitz.[6] Rule P10 is a bond lengthening rule. The symbol * represents any nonnull label in the pane. There is an aspect to such rules which is troubling to the chemist: to wit, the fact that the lengthened bond symbol actually consists of three separate symbols in three distinct panes although the chemical structure diagram as normally drawn by the chemist would contain a single stroke. We could overcome this objection by postprocessing the output patterns of this geometric grammar with another (and trivial) grammar component which converts sequences of three panes with short strokes into oversized panes with continuous long strokes.

An example of how this grammar operates to generate a structure diagram display by successive application of the rules is given in Figure 2. Note how a bond lengthening rule is invoked in such a way that overlap is avoided; rule P5 could not have been applied at that stage because that stage did not have all of the empty panes called for by the left side. The end result of this particular generation is one particular representation of 2-propanol. The alternative representations in Figure 3 would necessarily require a different sequence of rule applications. This contrasts with the topological grammar which recognizes only one labeled graph representing 2-propanol.

APPLICATIONS AND RELATED WORK

In the preceding section the susceptibility of chemical structure diagrams to linguistic analysis was shown, and two distinct approaches to grammars were presented. We now discuss three areas of application where linguistic research might lead to pragmatic results.

Input Validation. One of the possible applications of

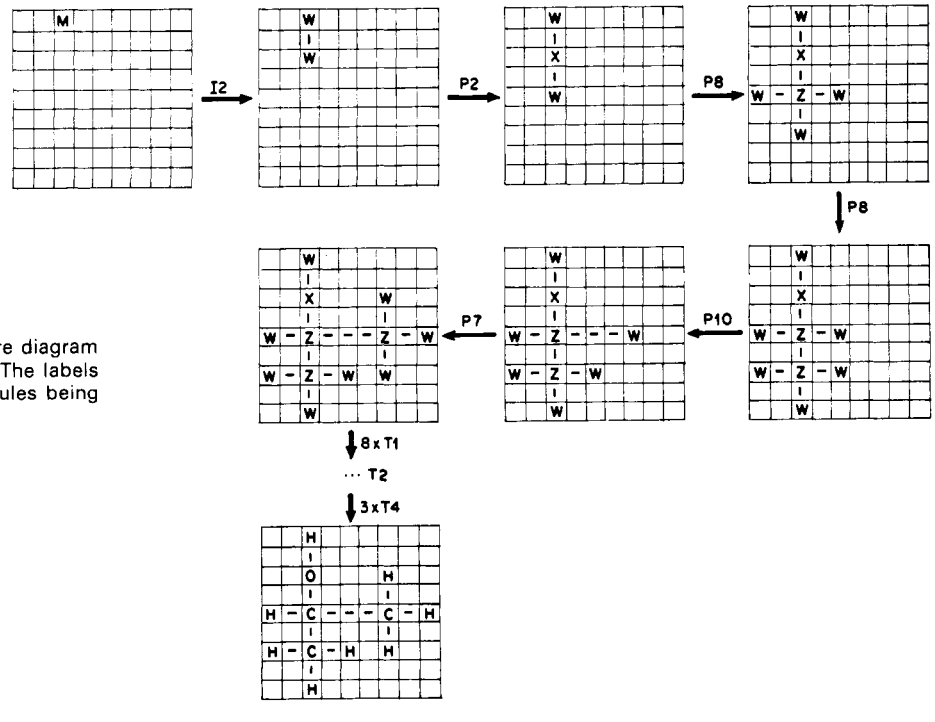


Figure 2. Sample derivation of a structure diagram for 2-propanol in a geometric grammar. The labels on the arrows indicate the replacement rules being applied

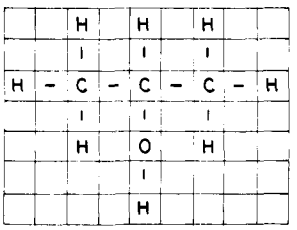


Figure 3. Alternative structure diagram for 2-propanol

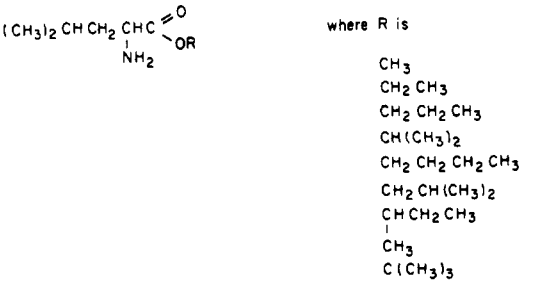


Figure 4. Nest of structure diagrams for a set of leucine esters

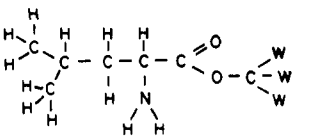


Figure 5. Intermediate stage of derivation common to leucine esters

linguistic research such as reported on here would be the construction of analytic algorithms for validation of chemical structure diagrams comparable to validation subroutines which now exist as preprocessors to computer language compilers.[7-10] Such an approach would be an alternative to the heuristic validation checks currently existing.[11, 12]

Validation could be done with either of the two types of grammars we have discussed. In both cases, validation is a procedure which is the inverse of generation. The problem is to determine whether a putative structure diagram is a valid structure diagram, where validity is defined by the grammar. The procedure would begin by scanning the input diagram for occurrences of subpatterns which occur on the right side of the replacement rules. The procedure continues by successively creating stages in a derivation "from the bottom up." The procedure terminates when either (1) the initial symbol is reached, or (2) no right-hand subpattern is found and the initial symbol has not been reached. In case (1), we have a valid structure diagram; in case (2) we have an invalid diagram (presumably one mistakenly drawn).

Acceptance of a structure diagram offered to the validation routine would provide an analysis of the diagram in terms of the subpatterns featured by the grammar. By the choice of the natural groupings[4] which a given grammar emphasizes, the validation routine could be made to support an algorithm for classifying structure diagrams according to chemical type. To the extent that this capability were utilized, the validation routine would function as a preprocessor in somewhat the same sense that a syntax validator is a preprocessor to a compiler.

Structure Diagram Display. The distinction is made above between topological and geometric grammars in the capability of the latter to generate a display directly whereas the former requires a second component to convert abstract graphs into two-dimensional displays.

Pieces of this problem have been tackled—e.g., the algorithmic generation from linear notations for acyclic graphs of representations which will drive a plotter,[13] the plotting of both cyclic and acyclic graphs from algorithmically derived and hand-coded representations,[13] and the development of a geometric grammar with considerable attention to the overlap problem.[14] These pieces are now discussed in the order mentioned.

First, there is a programmed system which will accept, as input, topological representations of acrylic graphs; will generate an intermediate representation; and will produce, as output, displayed diagrams. These topological representations are in the form of "vernacular" DENDRAL strings[15] (strings which conform to the notational formats but which do not necessarily obey the seniority criteria) and are at the same time possible equivalents to the abstract graphs which are the outputs of a topological grammar (see the section above). The system allows for experimentation with various methods for orienting the pieces of an acyclic graph. The problem of accidental overlap of the pieces is dealt with directly, but no general solution to that problem has been programmed.

Second, there is a program which will accept, as input, hand-coded representations of both cyclic and acyclic graphs and will produce, as output, displayed diagrams.

Third, there is a geometric grammar which generates patterns corresponding to a major subset of acyclic structure diagrams. This grammar incorporates a solution to the problem of accidental overlap. The solution essentially involves a search for blank space surrounding a nonterminal symbol so as to apply, when possible, a rule which involves elaboration of the chemical structure. If not enough blank space is found to permit applying one of these rules, then one of the bond-lengthening rules is applied.

Compact Storage. Work on string languages, specifically with selective computer generation of strings from grammars,[13] suggests the conjecture that a method more compact than storage of direct structure representations may be available. Grammar rules can be stored and pointers can invoke appropriate rules in proper sequence to generate the structure diagrams on call. The compactness would derive from applying labels to arbitrary sequences of pointers and from the possibility of nesting such sequences.

If we consider the set of leucine esters shown in Figure 4, we can cover the common structural features by applying, from the grammar in Table II, the sequence of rules I1(T1)-P3(T1)(T1)(T4)P3(T1)(P3(T1)(T1)(T1)T4)(T4)P3(T1)(T1)(T4)P3(T1)(P2(T1)(T1)T3)(T4)P4(T2)(T4)P1(T2)P3(T4) to generate the graph shown in Figure 5. Each rule is applied to a symbol introduced by the right-hand side of the previous rule or the rule preceding the previous parenthesized rules. If a rule introduces n nonterminal symbols, then it is followed by n rules of which n-1 are parenthesized. In the example just given, the first P3 applies to one of the W's generated by I1, and the T1, T1, and T4 immediately following generate two H's and a C from two of the W's and the Z generated by this P3, while the second P3 acts on the third W. If we give the label LE to this sequence of rules, then the complete sequences for the several esters are those shown in Table VI. The introduction of a multiplier notation for successive application of a rule or rule sequence[13] would of course improve the compactness of such a notation.

The choice of which sequences for applying rules deserve labels is a heuristic one, and it would probably differ according to how various files of compounds might be organized (see Input Validation). The rule sequences need not begin with an initialization rule, and thus labeled sequences could perfectly well be strung one after another. Thus the rules for generating the graph of leucine *tert*-butyl ester might be rendered as LE MC, where MC represents (P3-(T1)(T1)(T1)T4) thrice in succession. It seems reasonable

Table VI. A Set of Leucine Esters Defined by Topological Grammar Rules

Leucine Ester	Definition by Rule
CH_3	LE(T1)(T1)T1
CH_2CH_3	LE(T1(T1)P3(T1)(T1)(T1)T4
$CH_2CH_2CH_3$	LE(T1)(T1)P3(T1)(T1)(T4)P3-(T1)(T1)(T1)T4
$CH(CH_3)_2$	LE(T1)(P3(T1)(T1)(T1)T4)P3-(T1)(T1)(T1)T4
$CH_2CH_2CH_2CH_3$	LE(T1)(T1)P3(T1)(T1)(T4)P3-(T1)(T1)(T4)P3(T1)(T1)(T1)T4
$CH_2CH(CH_3)_2$	LE(T1)(T1)P3(T1)(T4)(P3(T1)-(T1)(T1)T4)P3(T1)(T1)(T1)T4
$CHCH_2CH_3$ $\vert$ CH_3	LE(T1)(P3(T1)(T1)(T1)T4)P3-(T1)(T1)(T4)P3(T1)(T1)(T1)T4
$C(CH_3)_3$	LE(P3(T1)(T1)(T1)T4)(P3(T1)-(T1)(T1)T4)P3(T1)(T1)(T1)T4

to expect that the "natural groups" recognized by the chemist[4] should be the most likely objects of such rule sequences. Thus there would be modules for "leucyl," "*tert*-butyl," "cholestane nucleus," etc.

OPEN QUESTIONS

We have presented what we believe to be a promising approach to dealing with chemical structure diagrams by linguistic techniques. By no means do we claim a solution to the problems of validation, display, or storage of such diagrams.

To date, a grammar exists for only a small sublanguage within the domain of chemical structure diagrams.[14] Any grammar would have to be tested for adequacy by comparing the patterns which it generates against those actually used by chemists. The borderline distinction between the unacceptable and the barely acceptable will be particularly difficult to make in a rigorous fashion. It remains to be seen how much of the natural language of chemical structure diagrams is accessible to grammars of the types discussed and whether the practical domains of topological grammars and of geometric grammars differ. Certainly considerable experimentation would be needed to test the truth of our conjecture that grammars provide a tool for compact storage of structure diagrams.

LITERATURE CITED

(1) Corey, E. J., and Wipke, W. T., "Computer-Assisted Design of Complex Organic Synthesis," *Science* **166,** 178 (1969).

(2) Farrell, C. D., Chauvenet, A. R., and Koniver, D. A., "Computer Generation of Wiswesser Line Notation," *J. Chem. Doc.* **11,** 52 (1971).

(3) Vasta, B. M., Spann, M., and Gelberg, A., "Chemical Notation Systems: Current State of the Art and Future Trends," presented at 161st Meeting, ACS, Los Angeles, March 29, 1971.

(4) Rankin, K., and Tauber, S. J., "Linguistics as a Basis for Analyzing Chemical Structure Diagrams," *J. Chem. Doc.* **11,** 139 (1971).

(5) Walker, J. C., Rankin, K., and Tauber, S. J., National Bureau of Standards, Washington, D. C., unpublished results, 1971.

(6) Zimmerman, B., and Lefkovitz, D., "Computer-Generated Structural Formulas with Standard Ring Orientations," presented at 6th Middle Atlantic Regional Meeting, ACS, Baltimore, Feb. 5, 1971.

(7) Morris, D., "The Use of Syntactic Analysis in Compilers" in "Introduction to System Programming," P. Wegner, Ed., Chap. 17, Academic Press, London, 1964.

(8) Lee, J. A. N., "The Anatomy of a Compiler," Chap. 3, Reinhold Publishing Co., New York, 1967.

(9) Galler, B. A., and Perlis, A. J., "A View of Programming Languages," Sect. 2.5, Addison-Wesley Publishing Co., Reading, Mass., 1970.

(10) Gries, D., "Compiler Construction for Digital Computers," Sect. 5.3, Wiley, New York, 1971.

(11) Cossum, W. E., Hardenbrook, M. E., and Wolfe, R. N., "Computer Generation of Atom-Bond Connection Tables from Hand-Drawn Chemical Structures" in "Parameters of Information Science," A. W. Elias, Ed., Proc. Am. Doc. Inst., Vol. **1,** pp. 269 ff., Spartan Books, Washington, 1964.

(12) Jacobus, D. P., Davidson, D. E., Feldman, A. P., and Schafer, J. E., "Experience with the Mechanized Chemical and Biological Information Retrieval System," *J. Chem. Doc.* **10,** 135 (1970).

(13) Orser, D., National Bureau of Standards, Washington, D.C., unpublished results, 1970.

(14) Currie, B., and Rankin, K., "Generative Grammars and Chemical Structure Diagrams," 160th Meeting, ACS, Chicago, Sept. 14-18, 1970.

(15) Lederberg, J., "DENDRAL-64, A System for Computer Construction, Enumeration and Notation of Organic Molecules as Tree Structures and Cyclic Graphs," NASA Rept. No. **CR 57029,** Accession No. **N65-16328** 1964.