Computer Generation of Wiswesser Line Notation*

C. D. FARRELL,[1] A. R. CHAUVENET,[2] AND D. A. KONIVER[3]
Division of Computer Research and Technology, National Institutes of Health, Public Health Service, HEW, Bethesda, Md. 20014

Received June 1, 1970

Computer programs developed at the National Institutes of Health (NIH) produce uncontracted though otherwise canonical Wiswesser Line Notation (WLN) for a fairly broad class of compounds. An associated front end allows a chemist to communicate with the programs by drawing structures on a Rand Tablet. The WLN generation programs accept connection table input, either from a previously existing file or generated from the Rand Tablet drawing. The programs recognize situations which they cannot handle—the output is thus either correct WLN or a message by which the programs acknowledge their limitations. In general, correct WLN will be produced for any compound containing not more than one nonbenzene ring. Work is under way to extend this to polycyclic fused ring systems. The philosophy and concepts behind these programs are explained along with the more interesting algorithmic results. The role of the WLN-generation programs in a developing NIH chemical information system is briefly discussed. The WLN programs are written in Fortran IV and have been developed on a PDP-10 computer.

Every representation of a chemical compound is an abstraction whose value is a function of its accuracy and usefulness for a particular application. The most widely used chemical representation is the structural diagram, which might well be termed the natural language of chemists. Many representations are useful for special purposes and among these is the Wiswesser Line Notation (WLN), which has been found to have advantages in information retrieval applications because of its compact linear nature.[1,2]

Most chemists would prefer to interact with a computer in their natural language. One of the main objectives of this project has been to allow the chemist to communicate with computer-based systems in a manner that is convenient to both—to let the chemist speak structural diagrams while the computer speaks WLN, with the computer program acting as interpreter. WLN was chosen because it is a well-developed[10] and widely-used[11] linear chemical notation.

From a practical standpoint, such an interpreter must operate in the time frame of the user. This has been achieved. A computation time of less than one second has been accomplished for moderately complex structures such as that shown in Figure 1. WLN generated by the interpreter may be used to enter a WLN-processing program, as described later in this paper. One of the interesting sidelights of this work has been an extensive investigation of the WLN rules. This represents an attempt to fully specify the rules in a form other than English sentences. Some initial work in this regard was reported by Davis and Straka[3] on the automatic encoding of compounds. They generate a noncanonical WLN, whereas the system reported in this paper was designed to produce a canonical notation.

We have emphasized flexibility and natural language input so that a chemist requires little or no special training to use the system. The program through which the chemist communicates with the rest of the system is vital in this respect if the programs are to be used rather than observed.

The system may be viewed as consisting of three modules. The first, described below as the Input Program, runs the Rand Tablet/display system and allows the chemist to communicate with the other programs via structural diagrams. Upon completion of a structure, this program arranges the connection table in a form that is convenient for the other programs and turns control over to a Ring-Handling Program which tears the structure apart, identifying atoms as ring or nonring atoms and making lists of the connections of all rings. Benzene rings are identified and replaced by the WLN symbol "R." Assuming there are no nonbenzene rings, the Basic WLN Program attacks the transformed connection table and produces correct WLN for arbitrarily complex molecules. In cases where one nonbenzene ring exists, the Ring-Handling Program generates the WLN for the ring structure and interacts with the Basic WLN Program to derive correct locants and produce notation for substituents on the ring. When the complete WLN has been generated, it is typed on the user's teletype, and then he may proceed to draw another structure.

The design of each of the three modules mentioned

* Presented before the Division of Chemical Literature, 160th Meeting, ACS, Chicago, Ill., September 15, 1970.

[1] Present address, Computer Science Department, Stanford University, Stanford, Calif. 94305.
[2] Present address, Department of Biochemistry, Duke University, Durham, N. C. 27706.
[3] To whom inquiries should be addressed.

above, the manner of their interaction, and the role the entire system can play in a chemical information network, are described in the remainder of this paper. The basic flow-diagrams of the modules appear in the appendix.

THE INPUT PROGRAM

The chemist should be able to communicate with computer programs in his natural language of structural diagrams regardless of the representation used by or the purpose of the programs. The generalized structure input program allows the user to "draw" a molecule on the Rand Tablet[4] and examine the structure on the display as he creates it. The program generates a redundant connection table while the molecule is being drawn. Our program is similar to those developed elsewhere[2, 6] for the same purpose and is described below.

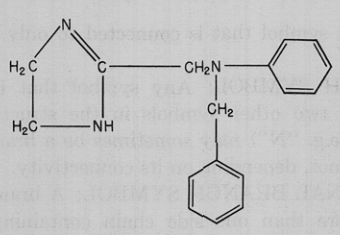
The Rand Tablet is a graphical input device, consisting of a 10-inch square "writing" surface and pen-like stylus. Under the hard surface of the tablet is a grid of 1024 by 1024 wires, so that there are 2^{20} distinguishable positions on the tablet surface. In addition, the stylus has a movable tip containing a microswitch. This switch is normally open but is closed when the stylus is pressed down on the surface.

A Cathode Ray Tube (CRT) display is used in conjunction with the Rand Tablet, since the stylus leaves no trace on the tablet but is sensed electronically. The user thus draws on the tablet and observes his drawing on the display. To orient the user, a dot is displayed on the CRT corresponding to the position of the stylus as it is moved over the tablet surface. The computer senses when the stylus is pressed down on the tablet and then lifted. This action is the means used to communicate with the program while drawing a molecule.

A "menu" of labeled boxes is displayed on the CRT, each box corresponding to an area on the tablet. The user indicates that he is choosing a particular box by pressing and then lifting the stylus within the area enclosed by the box. The program displays an arrow next to the latest box selected, both as a reminder to the user and as a signal that the action of pressing and releasing the stylus was correctly received.

Figure 2 shows the display as it appears when the program is ready for input. The menu is along the right border. The upper set of boxes contain the standard atoms used in constructing a molecule, while the lower set includes various editing features that have been found useful.

The user proceeds by selecting an operation from the menu and executing that operation until he wishes to



WLN: T5M CN BUTJ B1NR&1R

Figure 1. The computation time to produce the WLN for this compound was under one second

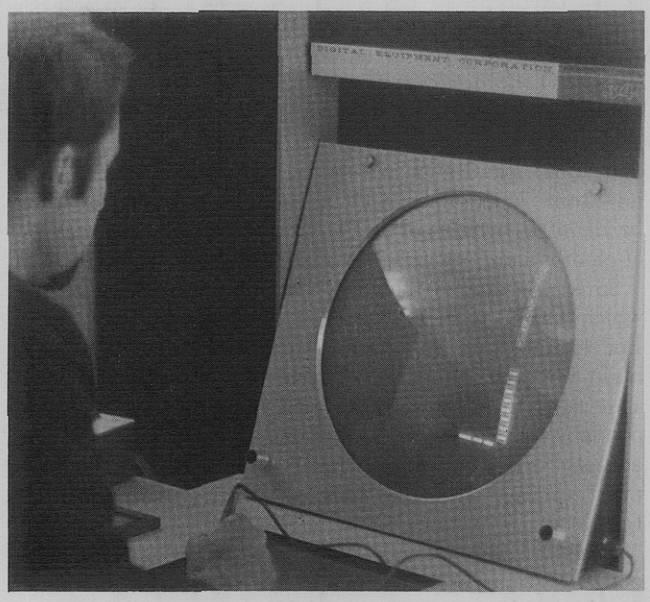


Figure 2. The user is going to input a structure

make another menu choice. Thus, after choosing carbon, one may place any number of carbon atoms in the drawing area, each automatically being bonded to the last, until another menu choice is made. At all times the program indicates the current "atom of interest" and will automatically bond any newly entered atom to the atom of interest. The atom of interest is generally the last nonmonovalent atom entered by the user. It may be changed by use of the SET editing command. Other editing commands allow insertion of additional BoNDs (BND) for multiple bonding and closing rings, DEletion of Bonds (DEB) and Deletion of AToms (DAT) for error-correcting, CANceling (CAN) the present menu choice without executing it, CLeaRing (CLR) the drawing area, ENDing (END) input of the current structure, or SAVing (SAV) the current structure. The SAV feature is quite useful if one wishes to enter several homologous structures, as it is only necessary to draw the basic molecule once, after which the SAVed file may be retrieved (redisplayed) and modified.

The atoms provided in the standard menu (except the special "atom" X) are those that have special WLN symbols and simultaneously are those that occur most frequently in organic compounds. A feature that is being added to the Input Program allows the user to establish new atom boxes containing whatever unusual atoms he desires. The X atom is used to input molecular fragments and serves the role of the organic chemists' "R group." It is used as an initiation point in WLN generation but is not represented in the resultant WLN. The molecule X—COOH would thus produce the WLN fragment VQ.

It has proven simple and convenient to enter a variety of chemical structures via the Input Program with no knowledge of the program and only limited training with the Rand Tablet. The Input Program contains some minimal chemical knowledge and will prevent the user from making obvious errors such as creating a 5-connected carbon. The program also knows the lowest valence of all the elements and will assume as many unspecified hydro-

gen atoms as are necessary to satisfy this minimal valence for each atom in a structure. Thus if the user draws C—C—O the program will interpret this as ethyl alcohol. Hydrogen atoms may, of course, be specified wherever they make it easier for the user to perceive the structure and must be drawn whenever the minimal valence of an atom is exceeded. Thus the phosphine group may be denoted by —P or —PHH, whereas all connectivities of pentavalent phosphorus must be explicitly indicated.

This program is a convenient communication device, which has uses in addition to the generation of WLN. For example, it has been used by another project to build a library of connection tables for mass spectral reconstruction. Furthermore, the rest of the system is independent of the Rand Tablet/display routines and can accept connection table input from previously stored magnetic-tape files, disc files, or directly from the user's teletype.

When a complete connection table has been input, various subroutines order the bonds in a way that is convenient for the WLN generation programs and indicate all hydrogens (both implicit and explicit) by flags. No attempt is made to construct a canonical connection table. Connection table comparisons are not performed. The existing programs operate equally well on any of the possible representations of a given molecule.

Figure 3 shows a completed drawing of acetylsalicylic acid, often a necessary medicant for the program development! In this example, all hydrogens have been explicitly entered and the "SET" box is on the methyl group carbon. The time required for an experienced Rand Tablet user to enter this structure is somewhat less than 30 seconds. The program makes no attempt to normalize the drawing. The structural diagram exists for the convenience of the user, and he is free to draw in any manner that pleases him. There is presently no means for entering stereochemical information into the connection tables, and therefore this facility is not provided in the current drawing routines.

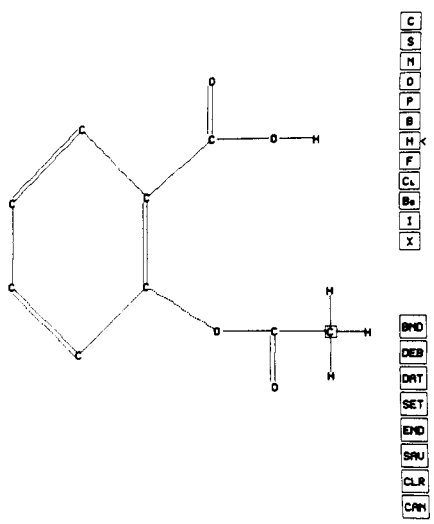
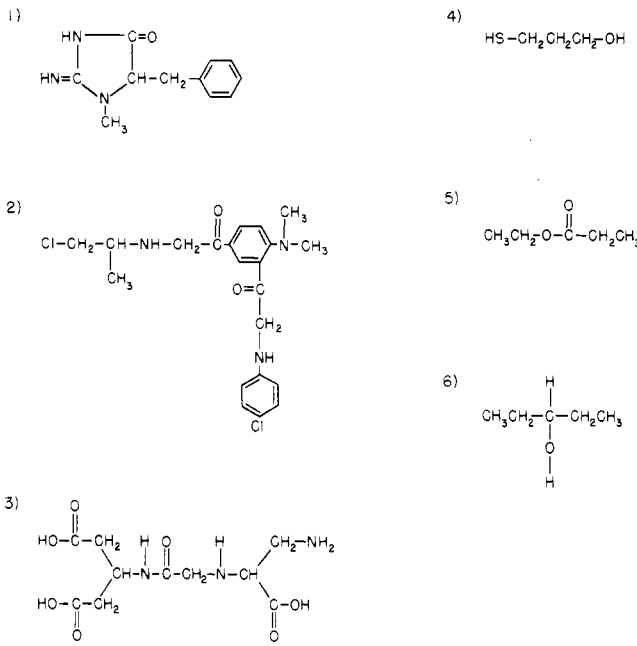


Figure 3. Acetylsalicylic acid has been drawn



	Wiswesser Line Notation	Time (Seconds)
1)	T5NYMV EHJ A1 BUM E1R	Ø.57
2)	G1Y1&M1VR CV1MR DG& DN1&1	1.22
3)	Z1YVQM1VMY1V-1VQ	.65
4)	SH3Q	.2
5)	2VO2	.2
6)	QY2&2	.18

Figure 4. Computation times

WLN GENERATION

The development of the WLN generation program has been influenced by several basic objectives. The program has the ability to recognize its present shortcomings (situations it cannot handle) and neither loop indefinitely nor produce incorrect WLN. A central feature of the system design is that the program produces either correct WLN or a message indicating the reason for its inability to generate the WLN. In general, uncontracted though otherwise canonical WLN will be produced for any structure accepted by the Input Program which contains not more than one nonbenzene ring. A second requirement is speed. The program has good execution time in a time-shared environment, as illustrated by the results shown in Figure 4.

In the discussion of implementation techniques and algorithms, the following terms and definitions pertaining to structures of atoms or WLN symbols are used:

END: A symbol that is connected to only one other symbol.

BRANCH SYMBOL: Any symbol that is connected to more than two other symbols in the structure. The same character (e.g. "N") may sometimes be a branch symbol and sometimes not, depending on its connectivity.

TERMINAL BRANCH SYMBOL: A branch symbol that has no more than one side chain containing other branch symbols.

LINEAR CHAIN: Any continuous chain of symbols, each of which is connected to no more than two other symbols in the chain. Branch symbols may be included in a linear

chain, but no more than two associated side chains may be included with any branch symbol.

EXTREMITY: A linear chain containing no branch symbols.

WLN GENERATION FOR NONCYCLIC STRUCTURES

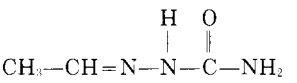
Once it is known that a certain structure or substructure has no rings, a program is entered that handles arbitrarily branched acyclic structures. Benzene is not considered a ring structure at this point because the Ring-Handling Program has created WLN "R atoms" to replace the six carbon atoms of benzene rings. The Basic WLN Program has two essential phases. The first phase steps through all the atoms of the structure, converting each atom or atom group into the correct WLN symbol(s) and creating a connection table for these symbols. A Subchain Array is also created during this phase which contains information about branch symbols and symbol lengths of all unbranched parts of the structure. The second phase makes use of these two tables to find the correct starting point for creating the final WLN and to construct a few smaller tables needed for handling complex branched structures. This phase also performs the final construction of the correct WLN notation when possible or prints out an appropriate message telling why it couldn't complete the translation.

Phase One. Figures 5 and 6 show two examples of the arrays created in the first phase of the routine. The Subchain Array has a variable number of rows and always four columns. Each entry in the first column contains the number of a WLN symbol (the row number from the WLN connection table) to which the linear chain being described is known to be connected. The first row of this array will always have a zero in column one since no connections are known at the start (the table is created dynamically as the structure is analyzed). The second and third columns contain the WLN symbol number for the beginning and end of a subchain of the structure. Each subchain starts with the next WLN symbol found in stepping through the structure and continues until either a branch symbol or an end is encountered. The fourth column is zero if the subchain terminated with an end; otherwise it contains the row number from the original connection table corresponding to the branched atom that terminated the subchain.

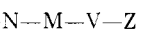
In Figure 6 the WLN connection table is not complete. Row two shows only two of the three connections for the Y symbol because the connectivity information for branch symbols is also contained in the Subchain Array and is actually not needed at all in the WLN connection table. It is partially represented in the WLN connection table simply for efficiency in the computer program.

Phase Two. Most of the complexity in the handling of noncyclic structures is involved with the analysis of branching. Other than the symbol translation performed in phase one, the WLN rules basically require that one determine the linear chains that (a) contain the greatest or smallest number of branch symbols, (b) contain the greatest or smallest number of WLN symbols, or (c) have the earliest or latest alphabetic value. Case (c) was handled by writing a subroutine that would iteratively perform alphabetic comparisons along two linear chains until a difference was discovered and reorder the two input arguments if necessary so that the first one always had the later value. This method allows many possibilities to be compared two at a time, without having to save more than the best one at any given time.

Structural Diagram



WLN Graphic Formula



Original Connection Table

1) C 2
2) C 1 4 4
3) C 5 6 7 7
4) N 2 2 5
5) N 4 3
6) N 3
7) O 3 3

WLM Connection Table

1) 2 2
2) U 1 3
3) N 2 4
4) M 3 5
5) V 4 6
6) Z 5

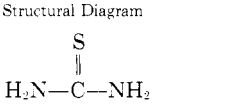
Subchain Array

0 1 6 Ø

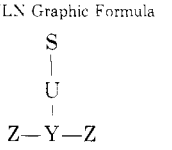
Final WLN
ZVMNU2

Figure 5

Structural Diagram



WLN Graphic Formula



Original Connection Table

1) N 2
2) C 1 4 3 3
3) S 2 2
4) N 2

WLN Connection Table

1) Z 2
2) Y 1 3
3) U 2
4) S 3
5) Z 2

Subchain Array

Ø 1 2 2
2 3 4 Ø
2 5 5 Ø

Final WLN
ZYZUS

Figure 6

It is fairly clear that as the degree of branching increases linearly the number of possibilities for cases (a) and (b) rises quadratically. This would greatly hamper the effectiveness of automatic translation in this domain if no means were developed for analyzing structures without having to consider all the possibilities. Such a reduction is possible by making use of the fact that most of the possibilities involve redundant evaluations of the same substructures. By carefully saving a small amount of information when it is first discovered, a procedure can be constructed which requires each part of the structure to be evaluated only once. This procedure therefore only increases linearly with the degree of branching.

In case (a), concerned with counting branch symbols, the first step is to create a connection table for all the branch symbols. Then the compression procedure shown below is followed for all cases except those having exactly zero or one branch symbols in the structure, which are handled separately.

1. A terminal branch symbol is found and given a value of zero.
2. The unused connection (there must initially be exactly one by definition) is then followed to another branch. This new branch symbol has an entry in the connection table of branch symbols which points back to the previous one. A value one greater than the largest value associated with the previous branch symbol is stored with this backward link in the connection table.
3. If all the connections associated with this new branch symbol have now been evaluated, the compression is complete and the two largest (smallest) values point outward from this branch symbol along the desired linear chain. Otherwise proceed to step 4.
4. If more than one connection associated with this new branch symbol is still unevaluated or if the one remaining connection is to a terminal branch symbol, then repeat steps 1 through 4.
5. Otherwise repeat steps 2 through 4 starting from this branch symbol.

This same procedure can also determine the linear chains with the greatest (smallest) number of WLN symbols [case (b)] by using the symbol count information implicit in the Subchain Array rather than constant values of zero in step 1 and one in step 2 for creating values for branch symbol connections. One other variation of this general algorithmic procedure is used to determine the shortest (either measurement) side chain off of some particular branch symbol. This is done by adding one more condition to step 4: If this new branch symbol is the one to be measured from, then repeat steps 1 through 4.

The compression algorithm described above creates and saves all the information for efficiently determining the longest or shortest linear chains through a molecule. All that remains is working backwards (selective uncompression) to find the actual symbols included in the desired chain(s). If ties exist, it is generally required that all tied chains should be saved for further analysis to break the tie. The uncompression technique described below will do exactly that:

1. Mark the final node of compression that had all its connections evaluated and choose the two connections with the greatest (smallest) value associated with this node. If there are ties for the second best, set a flag that indicates ties exist and choose all the tied connections.
2. For each of the connections chosen in the previous step, follow the indicated connection to the next node and mark this new node. If this new node is a terminal branch symbol, then this chain is completely uncompressed, all its internal components are marked, and the proper extremities can be determined from the Subchain Array and by alphabetization. Otherwise progress to step 3.
3. Choose the connection associated with the new node that has the greatest (smallest) value. Once again if ties exist, choose all the tied connections and set the flag indicating ties exist.
4. Return to step 2 with this new choice (choices).

When the uncompression is entirely completed all relevant atoms have been marked, and a flag will be set if ties exist anywhere. If there are ties, it is quite easy to reduce the table of branched connections to include only the branch symbols that were marked, thus creating a smaller (or equal) connection table with only the relevant chains needed to break ties.

The basic compression and uncompression algorithms are quite similar to techniques used by Moore[8] and Elithorn[5] for finding the shortest path through a maze. These other studies applied their techniques to the routing of telephone calls and railroads and to human psychological testing. The primary difference in the techniques lies in the fact that Moore assumed fixed beginning and ending points, and Elithorn established a fixed beginning point and a binary tree structure. The algorithms described in this paper were designed for the demands of creating WLN for arbitrarily branched molecules, and they represent a simple extension of Moore and Elithorn's techniques to arbitrary noncyclic graphs with no ends given. As mentioned earlier, this compression-uncompression technique only increases linearly with the complexity of the structure being analyzed, as opposed to the quadratic increase resulting from more obvious exhaustive methods.

Following the analysis of branching, the WLN rules require that terminal symbols be inserted into the notation in some cases, and that locants be created for di- or multisubstituted benzene rings. These tasks are handled by the Basic WLN Program at the time the final WLN string is created.

WLN GENERATION FOR CYCLIC STRUCTURES

There are three distinct components of the Ring-Handling Program: finding the rings, analyzing each ring system to produce a correct locant path, and generation of canonical WLN. A program for determining locant paths has been reported by Bowman *et al.*[1]

The ring-finding routines search the connection table for rings. The number of rings to be found is obtained by evaluating

$$R = 1 + C - A$$

where R is the number of rings, A the number of atoms, and C the number of connections. There is one connection if a is bonded to atom b, regardless of bond type. The reader can easily prove this formula by induction and note its equivalence to other ring-number formulas[13] which were found somewhat more cumbersome to state or employ.

The initial action taken by the ring-finder is to eliminate (flag) all external atoms. An external atom lies outside the ring structure and is flagged by the following simple procedure:

1. Flag each end atom in the connection table.
2. Go through the as yet unflagged atoms in the connection table and flag each atom which is connected to only one other nonflagged atom.
3. If no atoms were flagged during the last pass through the connection table, or if all atoms are flagged, procedure is complete. Otherwise, go to step 2.

This procedure has the effect of flagging atoms that are definitely not part of ring systems. Any atom left unflagged is either in a ring system or in a chain connecting ring systems. The flags are set in the original connection table while a new potential ring table is created which includes only the nonflagged atoms. If there is only one ring to be found, the procedure is completed, as the ring must include exactly those atoms in the potential ring table.

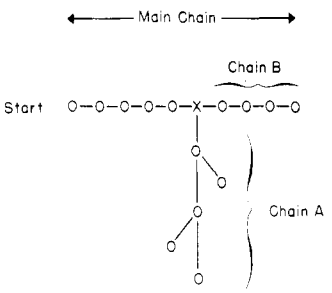


Figure 7. A graphic structure of branch points

It is possible to generate all cycles in the graph represented by the potential ring table. This is a time-consuming process which leaves the problem of deciding whether or not each cycle is a ring. The term ring has its usual chemical meaning, whereas a cycle is any path through a graph that returns to its starting point without visiting any other node more than once. For example, napthalene has two rings and three cycles. In general there are many more cycles than rings, and it is quite difficult to identify the rings.

Instead of this approach, a heuristic search has been implemented that will rapidly analyze arbitrary fused and simple bridged ring systems. The basic organization of this search is as follows:

1. Select as a starting point any 3-connected atom in the potential ring table that has not yet been a starting point and preferably is not part of any discovered ring. If not possible, select a higher-connected atom by the same criteria. If still not possible go to step 5.
2. From this point, search all chains (to a fixed depth), setting temporary flags on the processed connections. If no cycle is closed, turn off the flags and go to step 1. When a cycle is closed go to step 3.
3. Check the new cycle to see if it is a ring rather than a cycle composed of two or more rings. If the latter is true, split the cycle into its component rings. Then do step 4 for the ring(s).
4. Check the ring list to determine whether this ring has previously been found. If it has, set permanent flags on all edges between 2-connected atoms in the ring. If this is a new ring, add it to the ring list and leave temporary flags set between 2-connected atoms in the ring. If all rings have been found, the procedure is complete. Otherwise, select as a starting point either of two neighboring 3- (or higher) connected atoms in the new ring and go to step 2. If this is not possible go to step 1.
5. Erase all temporary flags and try to find a high probability starting point that has not yet been tried—e.g., a 3-connected atom that may be the center of a perifused system, where two of the three rings have been found. If this is not possible, terminate the search.

Once all rings have been found, they are classified into systems. "A ring system is defined as either a single ring or a group of rings joined to each other in such a way that each ring shares at least one ring atom in common with another ring in the group."[10]

Any one-membered ring system that is found to be a benzene ring is replaced in the original connection table (i.e., the six carbon atoms are deleted) by the WLN "atom" R whose nonhydrogen connections are indicated in a sequential path around the ring. If there are no nonbenzene rings, the Basic WLN Program is entered.

If a nonbenzene ring exists, a preliminary ring notation is created which includes the WLN for cited ring segment symbols and markers for uncited carbon atoms. The locant path is determined by comparing all of the reasonable possibilities—e.g., a heterocyclic monocycle must have a heteroatom in the "a" position. When required, the program interacts with the Basic WLN Program to determine alphabetic order of substituents on the ring. When all necessary locants have been determined, the canonical ring notation is generated, including any necessary H, U, or T symbols. Locants for substituents are provided by the Ring-Handling Program while the Basic WLN Program generates notation for the substituents themselves. Any ampersands required to "back over" benzene rings in substituents are inserted by the Ring-Handling Program.

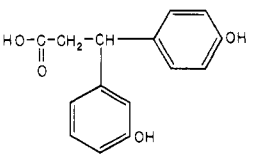
At this time work has been completed for single monocycles with substituents. The program is being extended to include polycyclic fused ring systems.

ANALYSIS OF WLN

Concurrent with the programming efforts, an extensive investigation of the WLN rules was required. Results of this study illustrate the need for a formal analysis of the WLN language. This can be demonstrated by several examples:

1. Figure 7 shows the branching structure of a hypothetical molecule, with the WLN starting point indicated. When the notation reaches branch symbol X, WLN rules 6 and 7 require citing first the symbol chain with the fewest branch symbols. It is not clear which has the fewest branch symbols—chain A, which has a total of five branch symbols with at most three in a linear chain, or chain B, which has four branch symbols in a linear chain. The algorithm for handling branched structures can be expressed more simply when chain A is to be cited first.
2. Figure 8 helps illustrate an omission in the rules. Rule 7 refers to rule 23 to determine which side chain to cite first, but rule 23 does not include this situation. We have patched rule 23 to include this case and produce the notation QV1YR CQ&R DQ to uniquely represent the meta-hydroxy substituent in a hierarchial order before the para-hydroxy substituent (locant C before locant D).

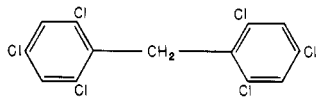
In addition to the main WLN rules there are special rules for methyl contractions and multipliers. The methyl contraction rules are fairly easily expressed as a string grammar operation on WLN strings and could be added to our present programs. This procedure is employed by many WLN users and produces a more concise notation.



Q V 1 Y R Q
R
Q

QV1YR CQ&R DQ

Figure 8.



Uncontracted WLN: GR CG EG B1R BG DG FG
Contracted WLN: GR CG EG B- 21

Figure 9.

However, contracted WLN strings are less desirable for creating structural diagrams and for some information retrieval applications.

Multipliers are very difficult to handle and undesirable in almost every respect except for conciseness of stored records. The major problem is that the applicability of multipliers is based on the structural diagram and has no well-defined relation to the WLN string. Thus the compound shown in Figure 9 may employ a multiplier even though the uncontracted WLN string, GR CG EG B1R BG DG FG, shows no repetitions of length greater than two! Furthermore, once it is determined that the notation for a compound might be contractable by multipliers, the contracted form must be constructed simply to determine whether enough marks are saved as compared to the uncontracted WLN. An examination of a 17,000-entry WLN file has shown that about 11% of the compounds are expressed with multipliers. The slight total saving in symbols does not appear to justify the use of multipliers in view of the problems they present in computer-based systems.

WLN AND INFORMATION SYSTEMS

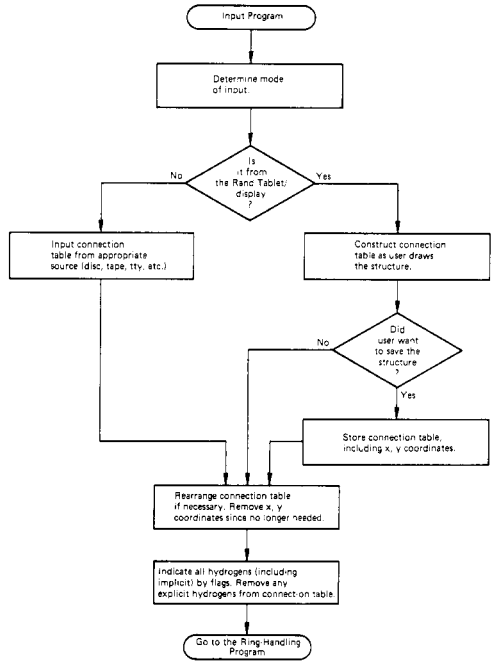
The Input Program and the WLN generation programs we have described may be used as components of a larger WLN-based chemical information system. While investigating the potential of such a system, a bit-screen search program has been implemented which employs an extension of the algorithm reported by Granito et al.[7] When matches are found, they should ideally be presented to the chemist in his structural diagram language. Accordingly, a program has been developed that generates diagrams from WLN, much in the spirit of CROSSBOW.[9] The existence of these components allows a chemist to have both his input and his output displayed as structural diagrams, while using WLN for information retrieval.

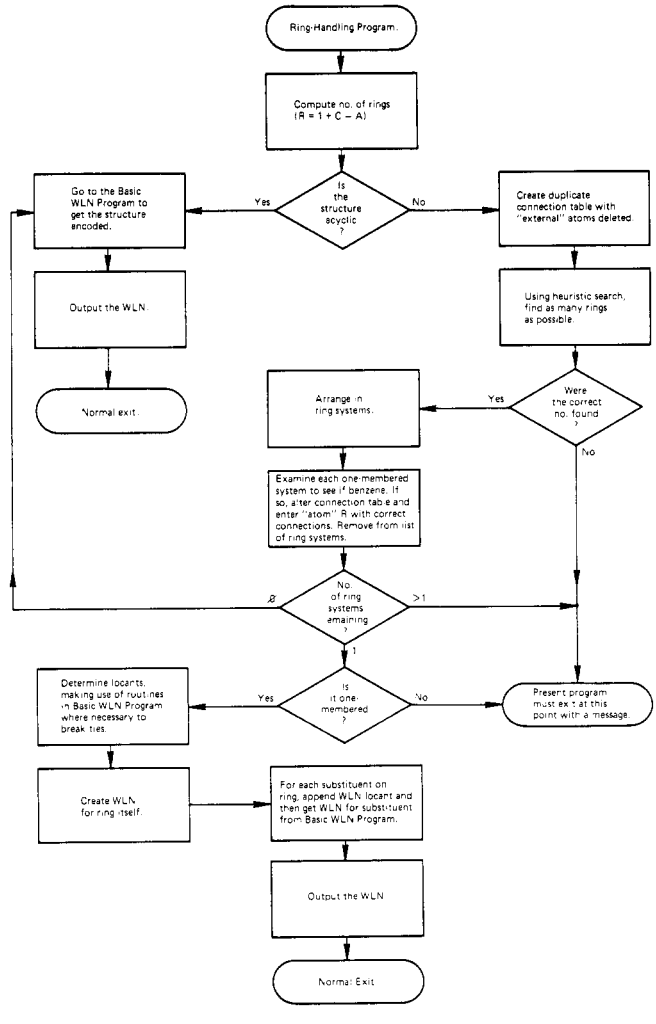
CONCLUSION

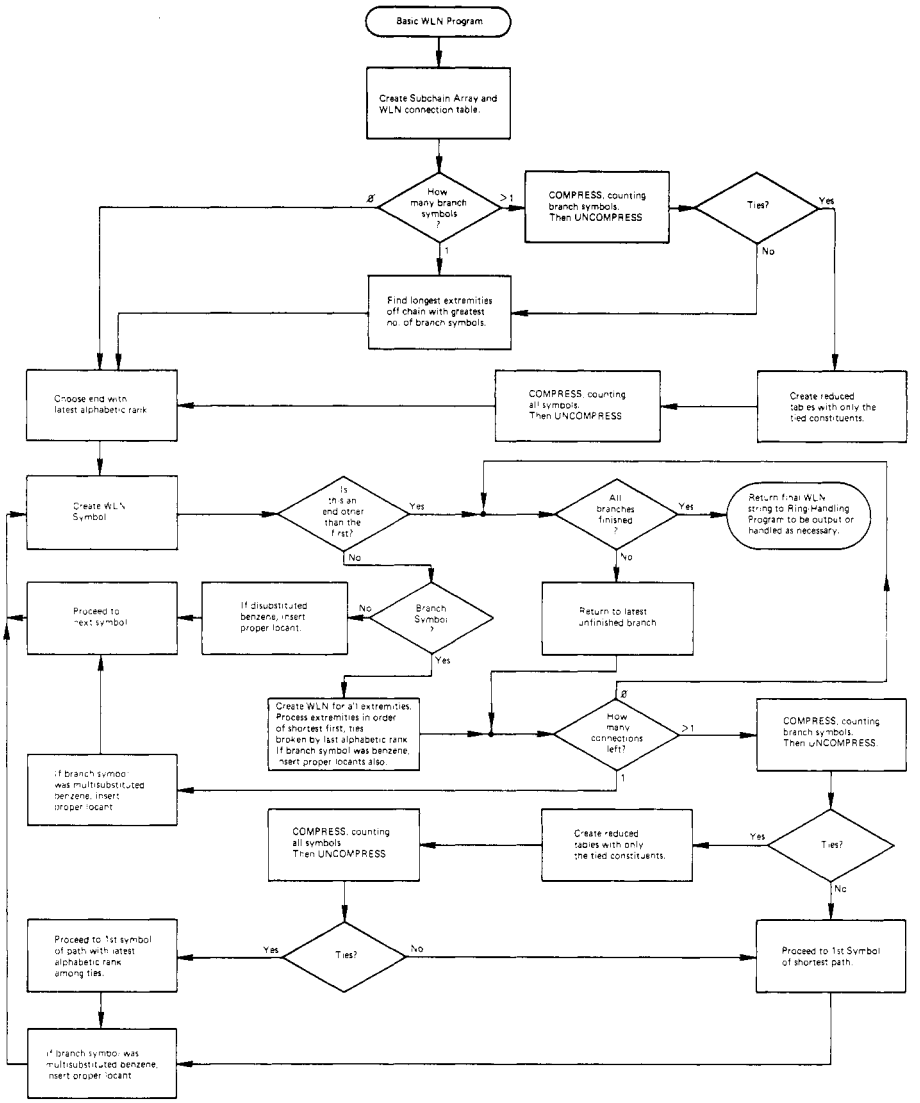
The automatic encoding of compounds into WLN will enable a chemist to communicate easily and efficiently with computer-based chemical systems. The programs described in this paper generate uncontracted canonical notation for monocyclic, benzene-containing, or acyclic compounds. One advantage of the programs is the certainty that only correct WLN will be produced. The system is presently being extended to create WLN for polycyclic fused-ring systems.

APPENDIX

The following flow-diagrams represent the basic logic of the three modules of the system: the Input Program, the Ring-Handling Program, and the Basic WLN Program.







ACKNOWLEDGMENT

The authors gratefully acknowledge the encouragement of and consultations with Alan Gelberg of the Food and Drug Administration and William J. Wiswesser of Fort Detrick.

LITERATURE CITED

(1) Bowman, C. M., F. A. Landee, N. W. Lee, and M. H. Reslock, "A Chemically Oriented Information Storage and Retrieval System (II): Computer Generation of the Wiswesser Notation of Complex Polycyclic Structures," J. CHEM. Doc. **8** (3), 133-8 (1968).

(2) Corey, E. J., and W. T. Wipke, "Computer-Assisted Design of Complex Organic Syntheses," *Science* **166**, 178-92 (1969).

(3) Davis, J. E., and L. E. Straka, "Automated Coding Via a Cathode Ray Tube," paper presented at the 158th Natl. Mtg., ACS, New York, N. Y., September 1969.

(4) Davis, M. R., and T. O. Ellis, "The Rand Tablet: A Man-Machine Graphical Communication Device," Proceedings of the Amer. Fed. Inform. Processing Soc. Fall Joint Computer Conference **26**, 325 (1964).

(5) Elithorn, A., and A. Telford, "Computer Analysis of Intellectual Skills," Int. J. Man-Machine Studies 1, 189-209 (1969).

(6) Gagnoud, A. M., "A Graphical Interactive System for the CRT Input-Output of Chemical Structures," DECUS Proceedings, 23-29, Spring 1969. Available from the DECUS office, Maynard, Mass. 01754.

(7) Granito, C. E., G. T. Becker, W. J. Wiswesser, and K. J. Windlinx, "Computer-Generated Superimposed Codes for Searching Chemical Substructure Files," paper presented before the Chem. Doc. Section, 5th Middle Atlantic Regional Mtg., ACS, Newark, Del., April 1970.

(8) Moore, E. F., "The Shortest Path Through a Maze," Proceedings of an International Symposium on the Theory of Switching, Ann. Computn. Lab. Harv., 285-92, 1959.

(9) Rogers, M. A. T., "CROSSBOW," paper presented at the 158th Natl. Mtg., ACS, New York, N. Y., September 1969.

(10) Smith, E. G., "The Wiswesser Line-Formula Chemical Notation," McGraw-Hill, New York, N. Y., 1968.

(11) Smith, E. G., "The Wiswesser Notation: State of the Art in 1969," a report to the Chemical Notation Association, September 1969. Available from Prof. E. G. Smith, Chem. Dept., Mills College, Oakland, Calif. 94613.

(12) Wiswesser, W. J., "107 Years of Line-Formula Notations (1871-1968)," J. CHEM. Doc. **8**, 146-50 (1968).

(13) Wiswesser, W. J., private communication.