PATHFINDER II. A Computer Program That Generates Wiswesser Line Notations for Complex Polycyclic Structures†

ANTONIO ZAMORA* and TOMMY EBE

Chemical Abstracts Service, The Ohio State University, Columbus, Ohio 43210

Received October 16, 1975

A program to generate canonical WLNs for polycyclic structures, originally written at the Dow Chemical Company, has been modified to significantly reduce execution time and to provide additional data for the encoder. This program is currently being used as a support tool for creating a WLN data base at Chemical Abstracts Service.

INTRODUCTION

Chemical Abstracts Service (CAS) is building a ring system data base that currently contains about 40,000 Wiswesser Line Notations[1] (WLNs). In an effort to make this data base as accurate as possible, mechanized procedures have been implemented to assist the manual encoding and editing of WLNs.[2] One of these procedures involves the use of a computer program (PATHFINDER II) to generate canonical WLNs for complex polycyclic structures. This program was derived from the original version described by Bowman, Landee, Lee, and Reslock.[3]

PATHFINDER II, while retaining the basic algorithm, differs from the original version in several respects. Equivalent starting points and equivalent paths are now identified to aid in manually encoding heterocyclic structures. Additional alternate notations are generated when necessary for manual selection. Input conventions have been altered to minimize the risk of human error. Edits have been added to check program input and output data for accuracy. The program was streamlined, and it now runs ten times faster than the original version. A mathematical relationship between structural features and processing time has been derived and confirmed. In addition, a program timer has been installed to monitor and limit processing time per structure.

A preferred path is the basis for a unique WLN for a given ring system. Finding the preferred path is the most time-consuming task for complex ring systems because of the large number of paths that must be examined. This task is ideally suited for a computer, and it is the major task performed by PATHFINDER II.

SELECTION OF PREFERRED PATH

WLNs for polycyclic structures are generated according to notation Rules 30, 31, 32, and 43. Rule 30 describes the hierarchical procedures for selecting the preferred path; Rules 31, 32, and 43 indicate the order in which the notation symbols must be cited for the preferred path. PATHFINDER II determines the preferred path for a polycyclic structure by comparing all possible paths with respect to the first seven hierarchial requirements of Rule 30, which require that the best path must (a) cite the smallest rings present; (b) cite the fewest rings necessary to define the structure completely; (c) have the fewest branch locants, all of which must be citable in the final notation; (d) have the lowest sum of fusion locants (a fusion locant is the earliest alphabetic locant occurring in a ring; the sum of the fusion locants is obtained by adding the ranks of the corresponding alphabetic character, e.g., $a = 1$, $b = 2$, etc.); (e) have the earliest alphabetic set of fusion locants in the order of their appearance in the notation; (f) have the earliest set of notation symbols for denoting bridges and nonconsecutive locants; (g) have the earliest sequence of ring numerals (e.g., 556 is preferred over 565).

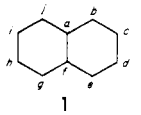
Rule 30 also requires that the preferred path must not cross a fused ring junction, i.e., that connection between two ring atoms each of which is (a) jointly shared by only two cited rings, and (b) connected to only three ring atoms.

The program constructs paths by assigning alphabetic locants to the ring atoms to build the longest possible chain of consecutive locants. Any atoms that are not in the longest chain are assigned branch locants, i.e., locants with hyphens. The path constructed by the program is compared with the stored "best" path, which initially is the path input by the encoder. A better path replaces the previously stored "best" path and the process is repeated. When all possible paths have been compared, the stored "best" path is the preferred path.

PATHFINDER II uses the preferred path to generate the WLN for a saturated carbocyclic system. The program identifies the starting point for the preferred path in terms of its input locant and also prints supplementary locant data for that preferred path. The encoder can then readily transfer the preferred path locants to the structure diagram and modify the generated WLN to describe any heteroatoms and unsaturation present in the ring system. To assist the encoder in finding the correct path for heteroatoms and unsaturation (Rule 30h to 30n), PATHFINDER II also identifies (a) any points equivalent to the starting point for the preferred path, and (b) any paths identical with the preferred path from the same starting point. The following paragraphs describe some of the features of PATHFINDER II that aid the encoder in his manual effort.

EQUIVALENT STARTING POINTS

When PATHFINDER II discovers a path identical with the preferred path from a *different* starting point, it prints a message stating that those starting points are equivalent. Paths are identical if they produce the same set of nonconsecutive locant pairs. The notational equivalence of the points at locants *a* and *f* in **1** is visually obvious.



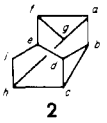
1

For complex ring systems, however, the existence of equivalent starting points is often not visually obvious. Hence the message identifying such points is a valuable encoding aid.

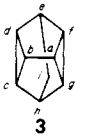
† Portions presented to the Division of Chemical Information, 170th National Meeting of the American Chemical Society, Chicago, Ill., Aug 25, 1975.

* To whom correspondence should be addressed.

The notational equivalence of the starting points at locants *a* and *b* in **2**, for example, would not be easily recognized by an encoder. By transforming the structure into **3**, the equivalence becomes evident.



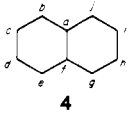
2



3

EQUIVALENT PATHS

When PATHFINDER II discovers a path identical with the preferred path from the *same* starting point, it prints a message stating that equivalent paths exist. In **1** an equivalent path exists from *a*, as shown in **4**. The same is true for *f*, since it is equivalent to *a*.



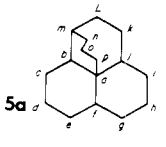
4

ALTERNATE NOTATIONS

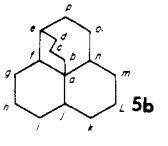
The WLN rules require that the earliest set of notation symbols for bridges and nonconsecutive locants (Rule 30f) must be obtained before considering the set of ring numerals (Rule 30g). Since PATHFINDER II does not generate the notation until it finishes processing all the paths, it cannot conveniently look ahead to determine if Rule 30f is satisfied. It therefore ignores Rule 30f when comparing paths, makes its choice by Rule 30g, but saves all the paths that are otherwise not differentiated through Rule 30e. When the path processing is completed, these saved paths are used to print alternate notations for selection by the encoder.

ALTERNATE PATH NOTATIONS

When the program *cannot* choose between two different paths from the same starting points because both paths have the same sequence of ring numerals, it generates alternate path notations for manual selection. Such alternate path notations are illustrated in **5a** and **5b**. The encoder can easily choose the preferred notation (**5a**), since /BM is earlier than /EP (Rule 30f).



5a



5b

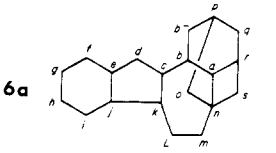
L6666/BM 3AAB P AXTJ

Preferred

L6666/EP 3AAF P AXTJ

ALTERNATE PATH (30F) NOTATIONS

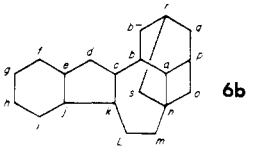
When PATHFINDER II *can* choose between two different paths by their sequences of ring numerals (Rule 30g), it still generates alternate path notations for the encoder as shown in **6a** and **6b**. Both paths produce notations that have the same alphabetic set of cited fusion locants (*ecaaaa*, Rule 30e) but differ in the sequence of ring numerals (657664 vs. 657466, Rule 30g). By applying Rule 30f, however, the encoder can determine that **6a** has the preferred notation, since /B-P is earlier than /B-R.



6a

L E6 C57664/B-P/NS B- 4AABN S NXTJ

Preferred

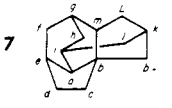


6b

L E6 C57466/B-R/NS B- 4AABN S NXTJ

ALTERNATE RING NOTATIONS

When generating a WLN for the preferred path, PATHFINDER II must occasionally choose between rings that are simultaneously closed by the same nonconsecutive locant pair, are of equal size, and have the same fusion locant. Since such a choice also requires looking ahead, the program instead generates alternate ring notations. For **7**, the fifth ring cited may be either *gfeabm* or *ghiabm*. The encoder can easily choose the preferred notation by Rule 30f (F vs.H).



7

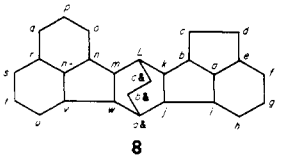
Preferred: L566 B56/B-K/GM B- F 5AABBE M BXTJ

Alternate Ring: L566 B56/B-K/GM B- H 5AABBI M BXTJ

INPUT CONVENTIONS

The description of a ring system to be processed by PATHFINDER II is input as a set of nonconsecutive locant pairs in the order in which the rings are closed by the path. For example, the five locant pairs *ae*, *ai*, *b-k*, *bm*, and *gm* are sufficient to describe the topology of the ring system illustrated in **7**. The set of nonconsecutive locant pairs for input to the program is derived from any arbitrary path chosen by the encoder. It is not necessary for this path to be the longest unbranched path or a path describing the smallest rings of the structure. However, if the input path closes the smallest rings or has few branch locants, the program processes the ring system faster. PATHFINDER II converts the input set of nonconsecutive locant pairs into a connection matrix that describes the ring system. It is this connection matrix that is traversed by the program in its search for the preferred path.

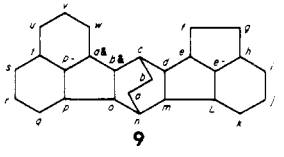
PATHFINDER II also accepts as input a list of fused ring junctions that must not be crossed. During routine processing it is not necessary to provide this list to the program, because the definition of a fused ring junction is dependent on the path chosen rather than on the topology of the ring system. In **8** is shown an example where the nonconsecutive locant pair *mw* represents a ring fusion junction whereas the pair *jk* does not. This is due to the fact that, for the path illustrated, the locant pair *mw* is contained in only two rings, but *jk* is contained in three rings.



8

On the rare occasions when PATHFINDER II determines that the preferred path for a structure crosses a fused ring junction, it prints a diagnostic message that alerts the encoder to input a selected list of fused ring junctions and to reprocess the structure. For symmetric structures, the encoder may also need to restrict the number of points from which a path may originate, in order to prevent the program from crossing an equivalent fused ring junction for a path originating from a different starting point.

This apparently capricious procedure is followed to ensure that human error will not prevent the program from finding the preferred path. Since the path supplied to PATHFINDER II by the encoder is arbitrary, it is conceivable that a path might be input such as that illustrated in **9**. For this path the locant pairs *dm* and *ob&* are both fused ring junctions. If the encoder were to input both of these fused ring junctions, the program would not cross them and therefore would not find the preferred path.



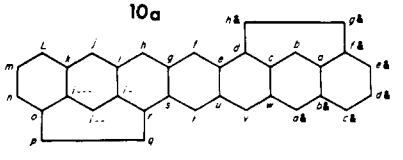
9

EDITS

A variety of edits, which make the program highly reliable, have been incorporated in PATHFINDER II to check the input data as well as the accuracy of the generated notation. Whenever errors are found, diagnostic messages are printed and processing is terminated. The input nonconsecutive locant pairs are checked to ensure that they are indeed nonconsecutive and that they are listed in the proper sequence. Input fused ring junctions are checked to ensure that they are a subset of the input nonconsecutive locant pairs.

The generated WLN is checked to detect uncited branch locants. This uncommon error occurs as a result of choosing a path with fewer branch locants, not all of which are citable, rather than a path with more branch locants, but all citable (Rule 30c). This is illustrated in **10a** and **10b**. The message alerts the encoder that the notation must be generated manually.

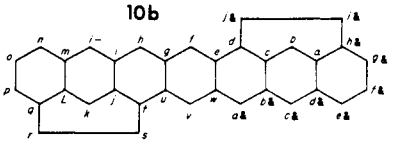
10a



L I6 I---6 I-7 G6 E6 C6667 B I-- 4ACI-I--- H&-TJ

Program Input

10b

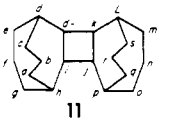


L I6 L6 J7 G6 E6 C6667 B K 4ACJL J&-TJ

ERROR LOCANT I- IS NOT CITED

Program Output

PATHFINDER II uses a subroutine, which finds the smallest set of smallest rings,[4] to detect cases where it is impossible by current WLN rules to describe a path that closes the smallest rings. In **11** is shown a case where the first ring closed by the best path that can be found has eight atoms, but the ring analysis (4,7,7,7,7) does not have any eight-atom rings. The program prints error messages for such cases.



11

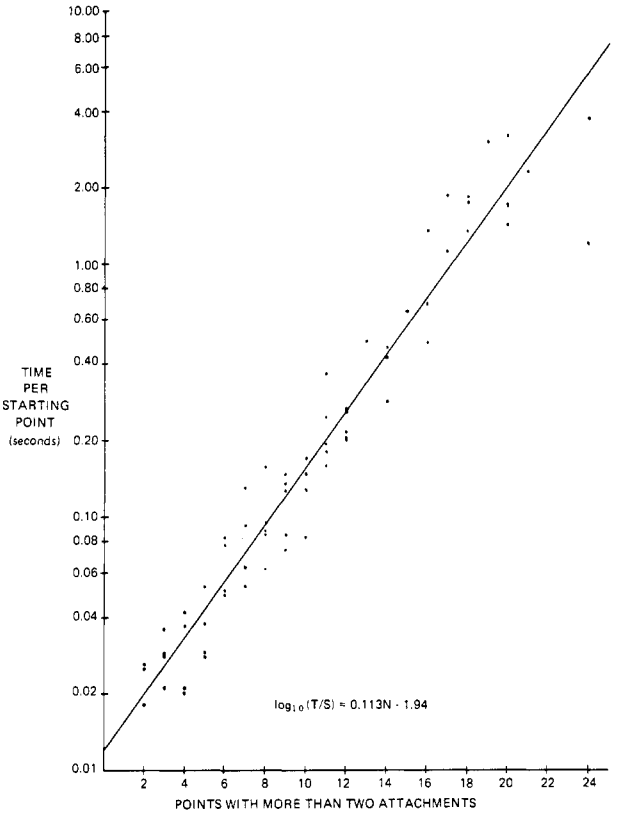


Figure 1. Dependence of execution time on structural features.

PROGRAM PERFORMANCE

A significant reduction in program execution time was achieved by streamlining the algorithm through the use of a software monitor and heuristic programming techniques. The software monitor, PROGLOOK,[5] was used to identify the sections of the program that were most time-consuming. Those sections were then analyzed and recoded to obtain the same results with fewer computations. The greatest gains in efficiency were obtained by using partial results from previous iterations to reduce the number of unsuccessful attempts at finding the preferred path. In addition, a frequently used table look-up subroutine was recoded in assembler language. Except for the program timer subroutines, all other portions of PATHFINDER II are coded in PL/1.

One of the heuristic programming techniques that helps reduce execution time is the early recognition of equivalent starting points. The program therefore processes only one such point completely. PATHFINDER II also automatically eliminates any points from which it would be impossible to start the preferred path. The only atoms tried as starting points are those that have three or more ring attachments, or those that are attached to an atom having three or more ring attachments. However, simple bridge heads, i.e., atoms attached to three ring atoms, all of which have only two ring attachments, are also eliminated. Timing studies on PL/1 versions of both the original and current programs demonstrated that these modifications reduced program execution time by more than 90%.

The time per starting point (T/S), required by PATHFINDER II to examine all the paths of a ring system and generate the WLN, is exponentially proportional to the number of ring atoms (N) having more than two ring attachments. Figure 1 shows the predicted linear relationship between log (T/S) and N. This relationship makes it possible to easily estimate an upper bound for the time PATHFINDER II would require to process a particular structure. Processing time can also be estimated using the number of rings (R) in the structure, since for most structures, $N = 2R - 2$. Since equivalent starting points are quickly identified by the program, the processing time for highly symmetrical polycyclic structures is less than estimated.

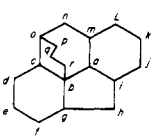
PROGRAM TIMER

One of the features of PATHFINDER II that helps to control the cost during routine use of the program is a program timer. A limit of one minute is established as the maximum amount of computer time to be spent per ring system. If the time required to examine all the paths exceeds the limit, the "best" path found during the analysis at that time is printed along with diagnostic messages. A new time limit may be established through a parameter card for a special computer run to complete the analysis of these structures. To avoid unnecessary repetitive processing, all points that have already been tried as starting points during the original processing are identified to the program through parameter cards.

EXAMPLES

The following examples further illustrate the notations and messages generated by the program. They also give the time required by PATHFINDER II, compiled with the PL/1 optimizing compiler, to obtain the results on an IBM 370/168 computer. Times previously reported[3] (Burroughs B-5500 computer programmed in Extended ALGOL-60) for examples 1 and 2 are given in parentheses.

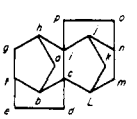
Example 1



WLN: L B6566 B6/CO 4ABBC R BXTJ

TIME: 0.93 seconds (90 seconds)

Example 2

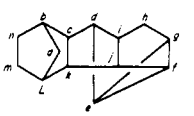


WLN: L B555 C5 J5 I5 A K 4BCIJ P CX IXTJ

MESSAGE: Locant a is equivalent to locant k

TIME: 0.96 seconds (124 seconds)

Example 3

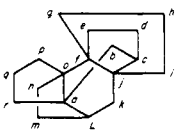


WLN: L E3 D5 D5 C555/FJ/BN A 3DEI NTJ

MESSAGE: Locant a starts 2 equivalent preferred paths

TIME: 1.15 seconds

Example 4

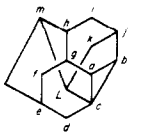


WLN: L F5 C56565/FJ/FO B 7AACFJJO R AX FX JX OXTJ

MESSAGE: Locant a is equivalent to locant j

TIME: 1.23 seconds

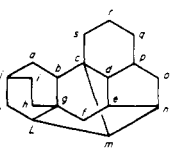
Example 5



WLN: L366 B56 C6/HM/EN D 8AABCCCGL N CXTJ

TIME: 1.75 seconds

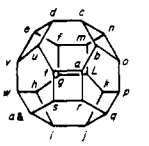
Example 6



WLN: L B666 B5 C5 D5 C6/GL/EN A 9BBCCDDEGG S CX GXTJ

TIME: 2.51 seconds

Example 7



WLN: L G6 F4 C6 B4 K6 J4646 D4 E6 H4 I6
-18-ABCDEFGHIJKLMNOPQR A&-TJ

MESSAGE: Locant a starts 2 equivalent preferred paths
Locant a is equivalent to all other locants

TIME: 29.12 seconds

ACKNOWLEDGMENT

The authors would like to thank Dr. Carlos M. Bowman and Mrs. Frances M. Voci for their helpful discussions and the Dow Chemical Company for providing the original PL/1 version of the program.

REFERENCES AND NOTES

(1) E. G. Smith, "The Wiswesser Line-Formula Chemical Notation", McGraw-Hill, New York, N.Y., 1968.
(2) T. Ebe and A. Zamora, "Wiswesser Line Notation Processing at Chemical Abstracts Service", *J. Chem. Inf. Compt. Sci.*, preceding paper in this issue.
(3) C. M. Bowman, F. A. Landee, N. W. Lee, and M. H. Reslock, "A Chemically Oriented Information Storage and Retrieval System. II. Computer Generation of the Wiswesser Notations of Complex Polycyclic Structures", *J. Chem. Doc.*, **8**, 133–8 (1968).
(4) A. Zamora, "An Algorithm for Finding the Smallest Set of Smallest Rings", *J. Chem. Inf. Compt. Sci.*, following paper in this issue.
(5) R. Johnson, and T. Johnston, "PROGLOOK Users Guide", COS-02250 Document No. SCC 007, COSMIC, University of Georgia, Athens, Ga., Oct 29, 1971, Rev. 3.