A Polish-Type Notation for Chemical Structures

By SYLVAN H. EISMAN
Pitman-Dunn Institute for Research, United States Army,
Frankford Arsenal, Philadelphia, Pennsylvania

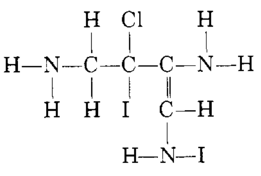
Received January 20, 1964

The notation system described here is an atom-by-atom representation of a chemical compound, rather than a cipher or a fragmentation code. It is designed primarily for use within an information machine (its translation rules are simple enough for a human to learn but quite tedious for him to apply) and particularly to reduce the amount of storage space required for the file of structural formulas. The number of characters required to store the structure is the sum of the number of atoms present, the number of connections which are not single bonds, and a number slightly greater than the number of rings in the representation.

The advantage of this system over the matrix or table forms of notation is the small amount of space required. The disadvantage is that a structure or substructure search is not convenient until this representation is transformed into another. In comparison with the conventional ciphers, the advantage appears to be the few number of rules for transforming between this linear representation and other, more easily searched forms. On the other hand, the ciphers are usually more compact.

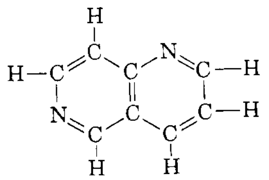
Background of Method.—A common representation of a chemical compound is its structural formula which can be considered as an undirected graph in the topological sense. That is, the atoms correspond with the nodes and the bonds with the edges of a graph.

A *tree* is a graph in which there is no cycle. That is, in passing from node to node along the edges, the same node cannot be reached twice *without* retracing at least one edge. Thus the structural formula



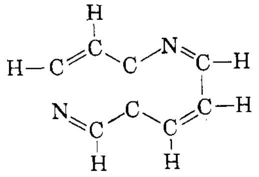
(1)

can be considered a tree whereas



(2)

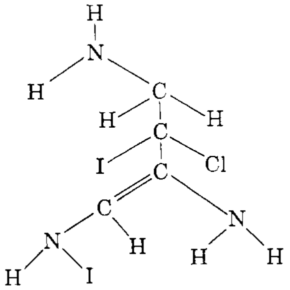
cannot. However, any graph can be converted to a tree by deleting some number of edges. An example of one way in which the graph of (2) can be made into a tree is



(2′)

Note that the nodes remain unaltered. Of course, the removed edges have to be remembered somehow to retrieve the original graph (2) from (2′).

Description of Method.—Several schemes exist for describing trees in a linear fashion rather than in the customary two-dimensional manner. As an example, (1) which looks more like a tree in the following form



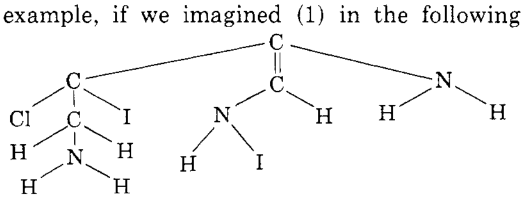
(1′)

can be represented as

H(N(C(HC(ClC(N(HH)C(HN(IH)))I)H)H)) (3′)

where everything within a pair of parentheses is considered bonded to the element just in front of the (symbol defining that parenthetical level. No account is taken, for the moment, of the type of bonding between atoms. Note that position within parenthetical level is in clockwise order starting from the point of entry (some such rule is necessary to fix the order of the elements). For instance, the last C in (3′) has attached to it an H and an N (the IH, at a lower parenthetical level, is attached to the N) while it is itself, along with an N, attached to the preceding C.

For each tree there are several representations of type (3′) depending on which element is chosen as the root. For example, if we imagined (1) in the following form



(1″)

it could have been represented as

C(N(HH)C(HN(IH))C(IC(HN(HH)H)Cl)) (3″)

Form (3″) contains no information that a double bond exists between the first two C's of the string nor does

(3′) reveal that a double bond connects the same two atoms (first two C's following Cl). One method of preserving this information is to precede the second recorded atom of the pair with the symbol d to indicate the double bond. Thus (3′) and (3″) become, respectively

H(N(C(HC(ClC(N(HH)dC(HN(IH)))I)H)H)) (4′)

and

C(N(HH)dC(HN(IH))C(IC(HN(HH)H)Cl)) (4″)

We could extend this idea to other types of bonds such as t and r for triple and resonant bonds, respectively.

The parentheses become redundant if we associate with each type node (atom) a "degree" (valence) which represents the maximum number of other nodes which can be attached to it.

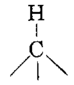
To fix this basic idea consider the string

H C N H H dC H N I H (5)

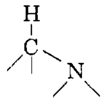
We assume the following table of known valences (multiple valence possibilities will be discussed in the following paragraph).

Atom Type	Valence
C	4
N	3
H	1
I	1

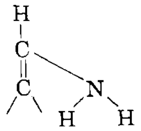
The first H has a valence of 1 and requires something to be connected with it. This is the adjacent C which still is not complete (has three bonds unsatisfied).



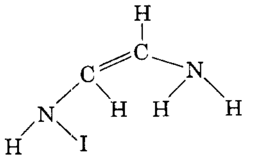
The following N connects with the C through the next clockwise bond from the point of entry reducing to two the number of unsatisfied C bonds, and adding two unsatisfied bonds of its own.



The next two H's connect with the N and have no further bonds themselves to satisfy (they are end points). Since all three bonds of the N are now satisfied, it is complete and the next atom in the string connects with the next clockwise bond of the C. Since that atom C is preceded by a d, the connection is a double bond. Therefore, the first C is complete.



This second C still has two (valence of 4 less 2) bonds to be satisfied. The first of these is taken up by the H which is itself now complete, and the second by N which requires the following I and H for its own completion. We thus have



Since there are no unsatisfied bonds remaining, the process is complete and this tree is a two-dimensional representation of the initial string (5). The entire tree can be considered as a unit with valence of 0 (a compound) and the part of the tree below the initial H a unit with valence 1 (an end point) which can be attached to any other single bond. As a matter of fact, this unit is part of (1′) and except for the first H, the string (5) appears in (4′) as a continuous segment. (Note: the initial H of (5) does *not* correspond with the Cl of (4′) but rather with the preceding C.)

Since the character representating the type of atom must also contain information regarding the valence, several different symbols must be used for an atom which can assume different valences. For example, if a compound is to be encoded by hand, a convention could be adopted which assigns the symbol N to trivalent nitrogen and the symbol Z to pentavalent nitrogen. If the encoding is to be done mechanically (*e.g.*, a chemical typewriter,[1] a a punched card system[2]) from a structure diagram where the single symbol N is used, a computer subroutine can be written which will determine how many bonds are connected to the N and select the preassigned codes for the nitrogen with this particular valence. Similarly, the practice of dropping H's should present no problem to a mechanical system. In this case, the atom to which the H's should have been connected will have a different code. For example, the code for the C's in this representation of acetylene, H—C≡C—H, would have to be different from the code for the C's in the skeletal representation, C≡C. Thus, in decoding, only three bonds would be anticipated instead of the customary four. In the "don't care" case, *e.g.*, —C≡C—, the symbol for the four-bond carbon atom should be assigned and a special symbol, R is a possibility, attached to the unsatisfied bond.

Making use of the preceding ideas, we are now free to eliminate the parentheses since the information they contain is implicit in the order of the nodes and the table of degrees. Appendix I gives the rules for forming a linear representation of a structure and Appendix II, the algorithm for determining the structure from its linear form. Thus, (4′) and (4″) become, respectively

H N C H C Cl C N H H dC H N I H I H H (5′)

C N H H dC H N I H C I C H N H H H Cl (5″)

This form is what we call parenthesis free (Polish-type) notation.

For the case of a cyclic compound (2), we may form a linear form as follows (using our clockwise rule).

H(C(dC(HC(N(dC(HC(HdC(HC(C(HdN(2))d5))))))))) (6)

The digit 2 in the expression HdN(2) indicates the atom to the left of it, N, is connected to the 2nd atom in the string. The symbols d5 in HC(C(HdN(2))d5) indicate that the atom immediately to the left at the next higher parenthetical level, the first C in the fragment, is connected to the 5th atom in the string by a double bond.

In Polish-type notation this would be

H C dC H C N dC H C H dC H C C H dN 2 d5 (7)

Had the last bond, represented by the 5, been a single one a separator (,) would have been used between the integers 2 and 5 to avoid confusion with 25. Notice that in counting position we count only atoms and not other characters.

[1] A. Feldman, D. B. Holland, and D. Jacobus, *J. Chem. Doc.*, 4, 4 (1963).

[2] W. H. Waldo, *ibid.*, 2, 1 (1962).

APPENDIX I

To obtain parenthesis free notation for a compound recorded as a two dimensional graph:

(1) Choose any atom as the first one in the string (in a computer this can be specified, *e.g.*, as the leftmost in the top line) and record its symbol. Go to step 2.

(2) (a) If the last atom recorded in the string has all its valence bonds "satisfied,"[3] back up in the string until the first "unsatisfied" atom is reached. Call this A. Go to step 3. If no such atom exists, the string is complete for this compound and the process halts.

(b) If the valence bonds are not satisfied, call the last element recorded A. Got to step 3.

(3) Consider the next bond of A in the clockwise direction from the last one whose connecting atom has been recorded. (If this is the 1st bond considered for this atom, choose it arbitrarily, or by specific rule if in a machine.) If this bond leads to an already recorded atom append an integer, k, specifying the kth atomic symbol (as opposed to the kth character) in the string to which the atom under consideration is connected. If this leads to a previously unrecorded atom append this new atom to the string. In either case, go next to step 2.

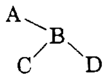
Note that this does not lead to a unique representation of a compound since it depends both on the orientation of the structure as originally drawn and on the choice of starting point.

APPENDIX II

We resort to a flow chart (Fig. 1) to specify the rules for retrieving the interconnections of the atoms from the Polish-type string.

Explanation of Symbols

n—total number of outstanding unsatisfied valence bonds. In this count



the bond leading from A to B is *not* considered satisfied until the atom B has all *its* bonds satisfied. As a corollary, when this algorithm is applied to any connected tree, the first element is the last to have all its bonds satisfied.

i—an index denoting the storage level in use (parenthetical level).

k—represents number of bonds connecting atom at current level to atom at preceding level.

F—represents the Polish-type string being decoded.

n_a—represents the assigned valence number of the symbol

N_i—represents the number of unsatisfied bonds of the atom under consideration at the ith level.

S_i—represents the storage cell holding the uncompleted atom at the ith level and its (so far) associated string.

φ—represents the null (empty) string.

L, R—represent the left and right halves of a register which can be individually addressed but which act as a unit in the shift operation.

σ^{+1}—represents the operation of shifting the string to the left by one character.

C(x)—contents of x.

N(x)—number associated with x.

Explanation of Blocks

1. Initialize indices and place Polish-type string into right register.
2. Clear left register.
3. Common case of single bond.
5. Shift one character from right register into left register.
6. Was there a character?
7. Is this is a first element?
8. Is character a special bond symbol?
9. Number of bonds into k; hold character in L for block 5.
10. Is character an integer?
11. "Valence" of first atom—don't reduce by number of valence bonds leading *into* it. Save as number of bonds to be satisfied for this atom.
12. Reduce "valence" of atom by number of bonds leading in. Save as bonds to be satisfied.
13. Increase total number of valence bonds outstanding.
14. Have all bonds of atom currently at ith level been satisfied?
15. Store contents of L.
16. Go back to previous atom.
17. Close cycle between atom currently at ith level and atom at position C(L) in string. Bond of order k.

18, 19. Reduce valence of atom at ith level and in position C(L) by k.

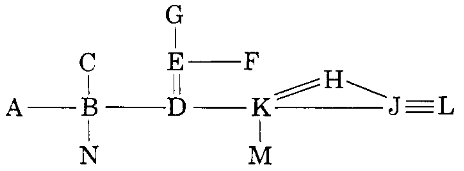
20. Reduce total number of outstanding valences by $2k$.
21. Prepare L register for 15 or 23 and 26.
22. Set i back one if end point of branch reached.
23. Check for multiple bond.
26. Attach last atom to atom above it in tree. This block (along with 17) will cause the difficulty in creating the format if an attempt is made to print out the structural formula.

27, 28. Reduce valence of ith level atom and total number of unsatisfied valence bonds.

29. Has the process returned to the first atom?
30. Check for unsatisfied bonds. "Don't care" conditions have special symbol.

Example of Use of Flow Chart

To demonstrate the sequence of operations in converting from the Polish-type string to the topologically equivalent molecular structure, we provide the following example. We shall use unique characters for each atom and assume the table of symbols *vs.* associated number of bonds is known. The structure is



(8)

with representation determined according to rules of Appendix I.

$$\text{A B C D dE G F K dH J tL 8 M N} \qquad (9)$$

Table I takes the Polish-type string (9) and traces its conversion, step by step, through the flow chart showing all changes in counters and indices. The entries in the columns labeled L and S_i merely show what connections have already been determined; the parentheses have been added in the sense of (4′) and (4″) for clarification. It is important to recognize that this algorithm does not produce a two-dimensional diagram but only the ordered

[3] In the case where bonds are left open to indicate radicals or "don't care conditions," a dummy symbol, *e.g.*, R, should be supplied.

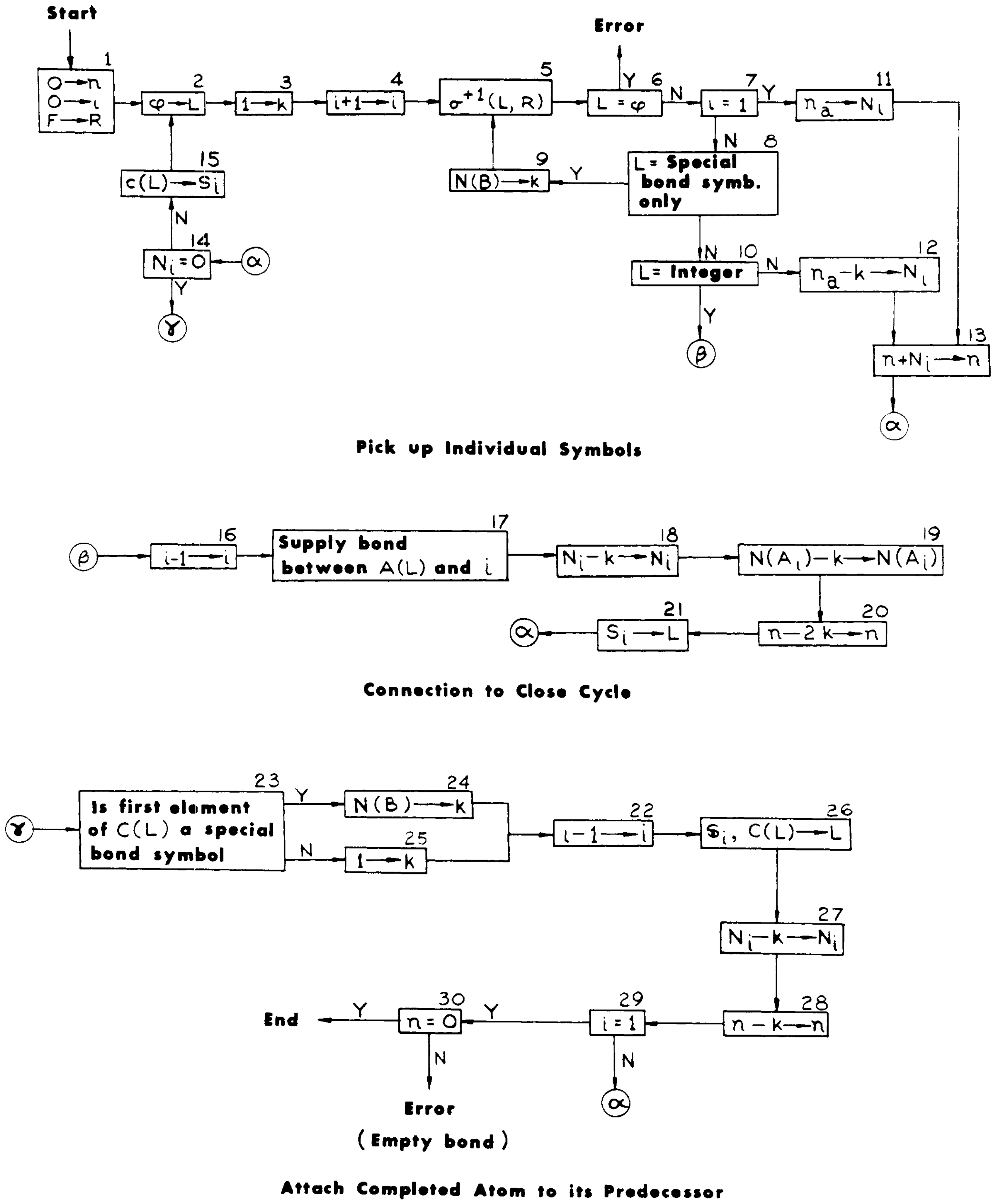


Fig. 1.—Flow chart for converting Polish string to structural formula.

Table I[a]

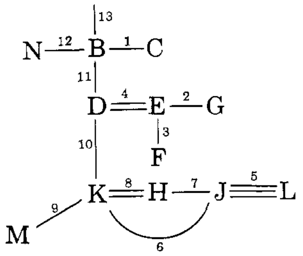
	k	i	L	N1	N2	N3	N4	N5	N6	N7	..	n	S1	S2	S3	S4	S5	S6
Start		0	Empty									0						
1	1	1	A	1								1	A					
2	1	2	ϕ B		3							4		B				
2	1	3	ϕ C			0						4						
4	1	2	B(C)		2							3		B(C)				
5	1	3	ϕ D			3						6			D			
6	1	4	ϕ d															
7	2		dE				2					8				dE		
8	1	5	ϕ G					0				8						
9	1	4	dE(G)				1					7				dE(G)		
10	1	5	ϕ F					0				7						
11	1	4	dE(GF)				0					6						
12	2	3	D(E(GF))			1						4			D(E(GF))			
13	1	4	ϕ K				4					8				K		
14	1	5	ϕ d															
15	2		dH					1				9					dH	
16	1	6	ϕ J						4			13						J
17	1	7	ϕ t															
18	3		tL							0		13						
19	3	6	J(L)						1			10						J(L)
20	1	7	ϕ 8															
21		6	J(L)[b]				3		0			8						
22	1	5	dH(J(L))					0				7						
23	2	4	K(H(J(L)))				1					5				K(H(H(L)))		
24	1	5	ϕ M						0			5						
25	1	4	K(H(′)M)				0					4						
26	1	3	D(E(′)K(′))			0						3						
27	1	2	B(CD(′))		1							2		B(CD(′))				
28	1	3	ϕ N			0						2						
29	1	2	B(CD(′)N)		0							1						
30	1	1	A(B(′))	0								0						

[a] Contents of any column (storage cell) remain unchanged until something else is read in. ϕ in the L-column indicates that the register has been cleared at beginning of this step. [b] In step 21; eighth atom is K; bond supplied between it and contents of $S_i = S_6$ which is J.

connections among the atoms which still have to be arranged in some geometric pattern.[4]

Beginning with step 25, the apostrophe within the parentheses indicates that the contents have been deleted from the table to conserve space.

The following diagram, produced manually with the aid of Table I, shows the order in which the connections were made.



ACKNOWLEDGEMENTS

The author gratefully acknowledges the stimulating discussions with Mrs. Marilyn Belson during which the ideas of this paper were developed. Thanks are also due the referees who made several suggestions for clarifying the presentation.

While this paper was in the final typing stage, the work of Prof. Hiz of the University of Pennsylvania, "A Linearization of Chemical Graphs,"[5] was brought to the author's attention. The basic ideas of the two papers are similiar but some of the details are quite different.

[4] If the decoding is done manually, the procedure will follow the brief description given for the conversion of (5) with the decoder choosing the geometric layout which he can rearrange at will. Use of the flow chart was implicit in that example. If the decoding is done mechanically, special provisions must be made for positioning the symbols once the connections are determined (boxes 17 and 26 of the flow chart). A preliminary program exists for the IBM 1401 which will take a Polish-type string notation of a non-ring structure and print out a two-dimensional picture on a line printer.

[5] H. Hiz, *J. Chem. Doc.*, 4, 135 (1964).