SYNTAX-DIRECTED TRANSLATION OF ORGANIC CHEMICAL FORMULAS INTO THEIR TWO-DIMENSIONAL REPRESENTATION

NEAL CARPENTER
Feldberg Computer Center, Brandeis University, Waltham, MA 02154, U.S.A.

(Received 21 May 1975)

Abstract—This paper describes a computer program which translates standard organic chemistry formulas into their two-dimensional representation. The syntactic rules describing the input language as well as the semantic actions which perform the translation when a rule is recognized are described in the paper. The translator is operational on a PDP-10 computer and displays the formulas using a plotter.

(1) INTRODUCTION

In 1957 the International Union of Pure and Applied Chemistry (IUPAC) standardized a nomenclature for organic compounds which has been widely accepted among chemists (Hendrickson et al., 1970; RCNOC, 1960). This nomenclature provides the organic chemist with a basis for identification of numerous compounds. A strict nomenclature system is obviously required to eliminate ambiguities in characterizing a chemical formula. The IUPAC system satisfies this requirement.

Syntax-directed translation (Gries, 1971) can be advantageously used to transform formulas in the IUPAC nomenclature into their standard two-dimensional representation. The purpose of this paper is to describe how this translation can be accomplished. The availability of translation schemes of this kind constitute the first step towards the automatic formulation of organic reactions by computer.

(2) TRANSLATION

The syntactic rules for defining organic compounds according to the to the IUPAC nomenclature are presented in Table 1. According to these rules, one can define compound names such as:

(1) 3-CHLORO-3-AMINO-3-PENT-4-ENONE
(2) 2,4-DIBR0MO-CYCLO HEXANE
(3) 3-HEPTYNE.

The above strings (implicitly) contain the following information necessary for the translation:

(a) the length of the carbon chain (e.g. PENT indicating a carbon chain of length 5 in (1), HEX meaning a chain length of 6 in (2));
(b) one or more unit prefixes, indicating the number of the carbon to which any functional group is attached, followed by the functional group itself (e.g. a chlorine atom and amino group are attached to carbon 3 in (1));
(c) the carbon number and type of unsaturation\(^*\), which composes the primary suffix (e.g. a double bond at carbon 4 in (1); a triple bond at carbon 3 in (3));
(d) a secondary suffix again providing points of attachment of functional groups (e.g. carbon 2 is a carbonyl in (1));
(e) singular or multiple functionality (e.g. 2 bromine atoms are present in (2)).

The translation process involves the execution of semantic actions once the elements of a rule are recognized while parsing (Gries, 1971). These semantic actions are informally described in Table 2.

The data structure utilized by the translator consists of a 2-dimensional table having four columns. Each time a new piece of information is translated, the corresponding entry is made in the next row of the table. The first column contains the position of attachment of any functional group. Column 2 is used to store the type of multiple bond (double or triple bond) if any exist in the given molecule. Column 3 contains the functional group itself. Column 4 stores secondary suffixes which, when present, would disturb the conventional numbering of carbon atoms in this particular method of display (e.g. COOH, CONH2, etc.). Besides the information stored in the table, the additional item needed to draw the formula is the length of the carbon chain. This length is determined by the semantic action attached to rule 7 (see Tables 1 and 2). The actual drawing proceeds in three steps. The carbon skeleton is first drawn followed by the attachment of the functional groups (if any exist) on the specified carbon atom. Lastly, multiple bonds are drawn into the molecule.

In order to illustrate the translation process, consider the input string: 2-HYDROXY-3,5-DI CHLORO-1,4-OCT ADI-EN-6-YN E. Notice that the blanks placed between basic symbols in this formula are automatically introduced as separators by the lexical scanner which is called by the syntactical analyzer (Gries, 1971). This string is parsed as shown in Fig. 1 and its translated 2-dimensional representation is displayed in Fig. 2. The corresponding data structure is given in Table 3. Notice that according to the semantic rules of Table 2 syntactically correct but semantically incorrect formulas are detected by the translator.

The described translation scheme has been implemented on a PDP-10 computer using a compiler generator (Cohen et al., 1972), the semantic actions of Table 2 having been coded into FORTRAN. The reader will find in Fig. 3 several examples which illustrate the current capabilities of the translator.

(3) FINAL REMARKS

Along with the translation of IUPAC names to their two-dimensional representation, the reverse translation—the parsing of structural formulas to their corresponding
The variable integer denotes an unsigned integer, defined as in most programming languages. This variable is recognized by a lexical scanner.

3. The above grammar does not contain self-embedding metavariables and it is therefore representable in a finite-state form. Thus the semantic actions could have been driven by a finite-state recognizer; however, the availability of a compiler generator was a major factor in the decision to implement the translator based on the above BNF rules.
Table 2

<table>
<thead>
<tr>
<th>Rule Number</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>No action is performed but notice that although cyclic compounds are not graphically displayed, they are accepted sentences in the language.</td>
</tr>
<tr>
<td>7</td>
<td>The length of the carbon chain is determined.</td>
</tr>
<tr>
<td>36</td>
<td>Make new entry in column 1.</td>
</tr>
<tr>
<td>30</td>
<td>Specific functionality is entered in column 3. If there is no entry in column 1 for that row, enter a '11' by default (the first carbon is conventionally used if no others are specified).</td>
</tr>
<tr>
<td>8, 9</td>
<td>Checks are performed to ensure previous entries in column 3 when multiple functionality is recognized.</td>
</tr>
<tr>
<td>25</td>
<td>Make corresponding entries in column 2 of data structure.</td>
</tr>
<tr>
<td>26</td>
<td>Flag either column 3 or 4 depending upon the type of secondary suffix.</td>
</tr>
<tr>
<td>1, 2</td>
<td>The information contained in the matrix is processed and graphically displayed using a plotter. Processing involves rearranging carbon attachment numbers, the length of the carbon chain and checking to ensure that there are 8 bonds to each carbon atom. If these requirements are not met, an error message is given. (The input string is syntactically correct but semantically incorrect.)</td>
</tr>
</tbody>
</table>

Fig. 1.

chemical names was also implemented. Examples of this inverse translation are given in Fig. 4. Checking is again performed to ensure the detection of syntactically correct but semantically incorrect strings (e.g. \( \text{CH}_2=\text{CH}(_4)\text{CH}(_3)=\text{CH}_3 \)).

It should be mentioned that both translation schemes have their limitations. The IUPAC nomenclature system cannot be used to describe numerous compounds due to the abundance of common names such as benzene, naphthalene, etc., but these names can be stored in a data
base with their appropriate intermediate matrices which enable the drawing of their two-dimensional representation. Similarly, the grammar which describes the reverse translation does not span the wide range of structural formulas—only those formulas which are systematically convertible to IUPAC names are acceptable. However, this grammar could easily be modified to accept and transform any structural formula into its corresponding data structure.

It is felt that the described translation schemes can be easily implemented and conveniently, used as input-output parts of general organic chemistry programs designed to simulate chemical reactions. To this effect, it should be mentioned that the internal data structure utilized by the translator can be readily transformed into an equivalent form (described in Hendrickson, 1971) and which is suitable for specifying reactions. Besides its usage in that context, the current programs can also be used as an educational tool in introductory chemistry courses.

Acknowledgements—The author wishes to thank Prof. Jacques Cohen for his guidance in the preparation of this paper and to Prof. James B. Hendrickson whose course in Organic Chemistry inspired me to do the work described in this paper.

REFERENCES