match no. 2 pp. 45-50 1976

AN ATTEMPT TO FORMALIZE THE CHEMICAL NOMENCLATURE

Ödön Mestyanek[*]

Budapest Technical University,
Department of Inorganic Chemistry

and

Péter Réti

Computer Application R and D Centre of
Chemical Industry, Budapest, Hungary

(received: June 1976)

1. Introduction

This paper is attempted to be a step towards formalizing the chemical nomenclature. The tool helping us in this direction is mathematical linguistics. It is found to be useful handling the various topics of chemistry and the sequel of this paper will be dealing with related themes.

2. Some notations and definitions

These notations can be found in the standard textbooks of mathematical linguistics, therefore they are listed here for reference only. (for more details, see [1]).

Alphabet: a non-empty finite set V.

V-word: a finite sequence of the elements of V,

Length of a word: the number of symbols contained in it. Let $\xi = x_1 x_2 \dots x_k$ be a word then its length will be notated by $|\xi| = k$.

V^* is the set of all possible V-words. (including the empty word Λ).

[*] to whom all correspondence should be addressed.

Grammar a 4 -tuple: $G = (V, \Sigma, S, R)$ where
V is a finite alphabet,
$\Sigma \subset V$,
$S \in V \setminus \Sigma$,
$R = \{ r: (u,v) \mid u,v \in V^* \}$ is the finite set of the rewriting rules, its elements are often written in the form $u \rightarrow v$.

Derivation: given G and $u,v \in V^*$. V is called derivable from u (notation $u \Rightarrow v$) if there exist $z_1, z_2, w, y \in V^*$ such that $u = z_1 y z_2$, $v = z_1 w z_2$ and $(y,w) \in R$ hold.
$u \overset{*}{\Rightarrow} v$ if either u = v or there exist such $u_0, u_1, \dots, u_r$, that $u = u_0$, $u_r = v$ and $u_i \Rightarrow u_{i+1}$ for all i.

Language: let G be a grammar then

$$L(G) = \{ x \in \Sigma^* \mid S \overset{*}{\Rightarrow} x \}$$ is called language.

From this definition one can see that a calculus is given by G. In the applications there are such definitions which are near to the notion of the normal algorithm (Markov-type). In this paper we restrict ourselves to the given definitions. Types of languages and corresponding grammars (according to N.Chomsky).

Type of grammar	The form of rules
3:	$\xi \rightarrow u,\ \xi \rightarrow u\vartheta \quad u \in \Sigma^*, \vartheta \in V \setminus \Sigma$
2: (it is called context-free,CF)	$\xi \rightarrow v,\ \xi \in V \setminus \Sigma,\ v \in \Sigma^*$
1: (it is called context-sensitive, CS)	$u\xi v \rightarrow uyv \quad u,v \in V^*,\ \xi \in V \setminus \Sigma,\ y \in V^* \setminus \{\Lambda\}$
0:	arbitrary

3. The chemical noménclature

The chemical nomenclature contains the definitive rules which are necessary to name a compound. The names, fixed by the nomenclature to the compounds, are a subset of the English language. Henceforth in this paper "nomenclature" always means the IUPAC 1957 rules. It is supposed that the reader is accustomed to the naming of chemical compounds and knows the basic rules (see [2]).

The nomenclature is a non-formal system. It would be desirable to have a one-to-one mapping from the set of compounds to the set of names (it is a pity that the nomenclature is not such a mapping).

The rules of the nomenclature remind one to the grammatical rules. Hence it is a natural thing to use the apparatus of the mathematical linguistics.

4. Generation of the names of unbranched alkanes

Alkane is the generic name of saturated acyclic hydrocarbons (branched or unbranched). In this section alkane always means unbranched and henceforth it will not be indicated.

The first four saturated unbranched acyclic hydrocarbons are called methane, ethane, propane and butane. Names of the higher members of this series consist of a numerical prefix and the termination "-ane". (Examples: pentane, hexane and so on).

Let $G_a^n = (V^n, \Sigma^n, S, R^n)$ be a grammar, where

$\Sigma^n = \{$ meth, eth, prop, but, pent, ..., ane $\}$ (the dots indicate that the numerical prefixes are given up to n, i.e. the set Σ^n contains n+1 elements),

$$V^n = \Sigma^n \cup \{\text{Alk, B, S}\},$$

$$R^n = \left\{ \begin{array}{lll} r_1: & \text{S} & \longrightarrow \text{Alk B} \\ r_2: & \text{Alk} & \longrightarrow \text{meth} \\ r_3: & \text{Alk} & \longrightarrow \text{eth} \\ r_4: & \text{Alk} & \longrightarrow \text{prop} \\ \vdots & \vdots & \quad\ \vdots \\ r_{n+2}: & \text{B} & \longrightarrow \text{ane} \end{array} \right.$$

Proposition 1.

$L(G_a^n)$ equals to the set of the names of all alkanes up to n C atoms.

Proof.

Σ^n contains the whole set of necessary morphemes and R^n provides for generating all names. It may be easily seen that the rules can be used only in order of their indices.

Corollary.

The set of the names of the alkanes is a type 2 language (context-free).

The result stated above can be formulated in the notions of the theory of automata. A push-down stack, which accepts the elements of $L(G_a^n)$ and only them, can be construated (see [1]). Changing G_a^n by writing yl instead of ane in one can see that the modified grammar generates the names of all monovalent normal hydrocarbon radicals.

5. Branched alkanes.

It was shown in the previous section that the names of alkanes and their radicals can be generated by context-free rules. In this section it will be seen that there are such subsets of the nomenclature which can not be generated by context-free rules.

A saturated branched acyclic hydrocarbon is named by prefixing the designators of the side chains to the name of the longest chain present in the formula.

Let $G_n^n = (\Sigma^n \cup \{S,A,B,C\}, \Sigma^n, S, R)$ be a grammar where
$\Sigma^n = \Delta^n \cup \Gamma^n \cup \Theta^n$,
$\Delta^n = \{2-, 3-, 4-, \ldots, n-\}$ (the set of designators),
$\Gamma^n = \{\text{methyl}, \text{ethyl}, \text{propyl}, \ldots.\}$ where the dots again denote that the set contains exactly n elements,
$\Theta^n = \{\text{methane}, \text{ethane}, \text{propane} \ldots.\}$,

$$R = \begin{cases} r_1\text{: } S \rightarrow ABC & \\ r_2\text{: } ABC \rightarrow AB\vartheta & \text{where } \vartheta \in \Theta^n \\ r_3\text{: } AB\vartheta \rightarrow \delta B\vartheta & \text{where } \delta \in \Delta^n \\ r_4\text{: } \delta B\vartheta \rightarrow \delta\gamma\vartheta & \text{where } \gamma \in \Gamma^n \end{cases}$$

Here r_3 is an abbreviation for some rules: should ϑ contain k carbon atoms r_3 contains k-2 rules where the number in δ can be at most equal to k-2. The analogous remark is true for r_4.

The reader must take into consideration that the number of rules contained in R is finite; they can apply unconditionally hence this grammar is not a matrix or program type one.

Proposition 2. The names of saturated branched acyclic hydrocarbons can be generated by a CS grammar.

Proof. G_b^n is a context-sensitive grammar. $L(G_b^n)$ is equal to the set of names dealt in propsition 2. The names are all valid ones.

Proposition 2 might arise the question whether there are context-free rules to generate the names of branched alkanes. The answer is no, because one can easily see that any context-free grammar generates besides valid names wrong ones as well (for instance 2-ethylmethane) and will always violate the rule of the longest chain.

We have not stated that a "weaker" than a context-sensitive grammar is not sufficient for generating the names of branched alkanes, but it must be emphasized that it has to be between CF and CS grammars.

6. Conclusions

We have seen that mathematical linguistics is a useful tool for investigating the chemical nomenclature. The subject of forthcoming research can be the total formalization of chemical nomenclature applying the results of mathematical linguistics or/and theory of automata.

7. Bibliography

[1] S.Ginsburg: The Mathematical Theory of Context-Free Languages. McGraw-Hill, London, 1966.

[2] Definitive Rules for Nomenclature of Organic Chemistry. In Handbook of Chemistry and Physics. 54th edition 1973-1974. editor Robert C.Weast. CRC Press.