match no. 19 pp. 277-286 1986

A LINGUISTIC VIEW OF LINEAR POLYMERS*

by

P.P. Thankachan
Physical Chemistry Division
National Chemical Laboratory, Pune 411 008, India

(Received: May 1985)

ABSTRACT

The theory of formal languages is used to model linear polymeric systems. The structure of such systems is modelled using generative grammars, and the possible use of semantic methods in studying the information content of such chains is suggested.

*NCL Communication No.3826.

INTRODUCTION

Mathematical descriptions of the physical world have only relatively recently begun to use the methods of "discrete" mathematics like graph theory, lattice theory etc. One of the more recent tools to have entered the arsenal of the physical scientist is the theory of automata, which has recently been used to study the statistical mechanics of self-organising phenomena[1]. An alternative approach complementary to the theory of automata is the theory of formal languages[2]. In the present work we outline the relevant theory and indicate its applicability to the theory of linear polymers.

OVERVIEW OF FORMAL LANGUAGE THEORY

We view a language as a set of sentences, each of which is a string of symbols. The (finite) set of symbols by whose concatenation the sentences of the language are formed will be called the terminal alphabet, and denoted by T. Now let us consider the set T^*, consisting of all possible strings formed by concatenating a finite number of symbols of T, and the null or empty string λ. We immediately see that

i) For all $a,b \in T^*$, $ab \in T^*$

ii) For all $a,b,c \in T^*$, $(ab)c=a(bc)$

iii) For all $a \in T^*$, $a\lambda = \lambda a = a$.

This shows that T^* is a monoid under the operation of concatenation; infact, it is the free monoid generated by T.

A language, then, is a subeet of the free monoid generated by an alphabet T. To define the set of strings of T^* that forms the language we use a grammar. Grammars could be of two types: descriptive or generative. A descriptive grammar describes the sentences of the language using rules of (descriptive) grammar, whereas a generative grammar defines the language by rules by which all the sentences (and no other strings) can be generated. While for natural languages descriptive grammars are in common use, for formal languages generative grammars are more frequently used.

A generative grammar is defined to be a quadruple

$$G = (N,T,P,S)$$

where N is a set of syntactic categories or a non-terminal alphabet, P is a set of rules ("production rules") and S is a distinguished element of N, called the start or sentence symbol; T as above is the terminal alphabet.

Each element of P, that is, each production rule, is an ordered pair (α,β) where α,β are elements of $(NUT)^*$, and contains at least one nonterminal. The production rule (α,β) is more commonly written in the form $\alpha\rightarrow\beta$ to make it transparent that the string α may be substituted by the string β.

Each derivation starts with S, and the production rules are applied in turn; a substring α is replaced by a substring β at a given stage only if the rule $\alpha\rightarrow\beta$ is in P. Note, however, that a replacement which is permissible is not

mandatory, and we may choose freely from all the permissible rules at a given stage. Given a string $u\alpha v$, if $(\alpha \to \beta) \in P$ then we say that $u\alpha v$ directly derives $u\beta v$, or $u\alpha v \Rightarrow u\beta v$; note that by the application of the rule $\alpha \to \beta$ the substring α has been replaced by β. If by the application of one or more production rules a string u is changed to a string v, we write

$$u \overset{+}{\Rightarrow} v \quad \text{(u derives v in one or more steps)}$$

We may extend this to include the case of zero productions by using

$$u \overset{*}{\Rightarrow} v$$

to denote "u derives v in zero or more steps". Clearly, $u \overset{*}{\Rightarrow} u$.

If $S \overset{*}{\Rightarrow} u$, we say that u is a sentential form of the language. In particular if u is a string of terminals alone, then there are no more rules which can be applied, and the sentential form u is called a sentence.

We now define the language generated by the grammar G to be

$$\mathcal{L}(G) = \{u \in T^* \mid S \overset{+}{\Rightarrow} u\}$$

that is, the set of all strings of terminals that can be derived from S in one or more steps.

As an illustrative example let us consider the grammar G_1, where

$$G_1 = (\{S\}, \{a,b\}, \{S \to Sa, S \to Sb, S \to a, S \to b\}, S)$$

It is easily seen that $\mathcal{L}(G_1)$ is the set of all non-empty strings over $\{a,b\}$, that is, $\{a,b\}^* - \{\lambda\}$. If on the other

hand we were to consider the grammar

$$G_2 = (\{S\}, \{a,b\}, \{S \rightarrow aSb,\ S \rightarrow ab\}, S)$$

then we can see that

$$\mathcal{L}(G_2) = \{ a^n b^n \mid n \geq 1 \}$$

Based on the types of productions present, grammars are classified into four broad classes:

Type 0 : No restrictions.

Type 1 : Productions are of the form $\alpha \rightarrow \beta$ where the number of symbols in β is at least as great as the number of symbols in α.

Type 2 : Productions are of the form $A \rightarrow \alpha$, where A is a single non-terminal

Type 3 : Productions are of the form $A \rightarrow Ba$ or $A \rightarrow a$ where A, B are non-terminals and a is a terminal.

Languages generated by type i grammars are called type i languages. These language types also correspond to sets of strings that can be recognised by different classes of automata, but we shall not dwell upon this aspect in the sequal except to point out that these automata could have been a starting point for an alternative formulation.

We note in passing that the example grammar G_1 was of type 3 and G_2 of type 2.

APPLICATION TO LINEAR POLYMERS

The simplest of the polymer structures is the linear structure, where monomeric units are linked together

in a chain without branchings. We may consider the monomeric units as being concatenated to form the polymer. Each polymeric species is an element of the free monoid M^* generated by the set M of monomers under concatenation. In view of the transparent analogy with the situation in formal language theory, we now explore the possibility of "generating" polymers of suitable classes by "Grammars" as discussed earlier. There are two cases to be distinguished, namely addition polymerisation and condensation polymerisation.

The case of addition polymerisation is illustrated by polyethylene, where C_2H_4 units add end-on to form polyethylene:

$$n\ (CH_2{=}CH_2) \longrightarrow -\ (CH_2\ -\ CH_2)_n-$$

Condensation polymerisation is exemplified by the polymerisation of terephthallic acid with ethylene glycol:

$$n\ HOOC.C_6H_4.COOH + nHO.CH_2.CH_2OH \xrightarrow{-2nH_2O} -[OCC_6H_4.CO\ OCH_2CH_2O]_n-$$

Here the repeat units do not have the same empirical formula as the set of monomers involved, since elimination of water molecules has occurred. Nevertheless, we may regard the chain as being formed by alternating $OC\ C_6H_4CO$ and OCH_2CH_2O units, which we take as the basic units ('monomers') in our description. It will be noticed that even in the addition polymerisation case we did not take note of the structural change involved in the C_2H_4 units, where the

double bonds of the monomer have given place to single bonds in the polymer. This is a sacrifice in structural terms that we make in the hope of ease of description.

Let us consider the set M = {a,b} of monomers (that is, only two types of monomeric units, denoted as a and b, are involved in our discussion), and investigate some of the types of polymeric structures that can be generated by suitably chosen grammars employing M as the terminal alphabet. We establish a convention that the start symbol will always be denoted by S, and that in what follows the terminals are the set M= {a,b} unless explicitly stated otherwise. With these conventions the grammar is specified when the non-terminals N and the rules of production P are specified.

(1) The set of all possible polymers. The grammer with a single nonterminal S and the rules of production

$$S \rightarrow Sa,\ S \rightarrow Sb,\ S \rightarrow a,\ S \rightarrow b$$

generates this set, for it is easily seen that

$$\mathscr{L}(G) = M^* - \{\lambda\}$$

It will be noted that this grammar is nothing but a paraphrase of the grammar G_1 of the earlier section.

(2) The set of all chains in which a's and b's alternate. We use the set of nonterminals S,A,B and the set of production rules

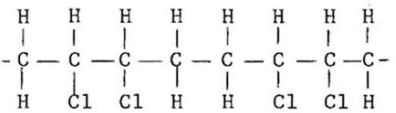
$$P = S \rightarrow Ab, S \rightarrow Ba,\ B \rightarrow Ab, A \rightarrow Ba,\ B \rightarrow b, A \rightarrow a$$

clearly the set of strings generated by this grammar is

$$\{b^r(ab)^n a^s \mid n \geq 1,\ r, s \in \{0,1\}\}$$

$$\cup\ \{a^r(ba)^n b^s \mid n \geq 1,\ r, s \in \{0,1\}\}$$

In addition to describing polymers like terylene formed from OCC_6H_4CO and OCH_2CH_2O units, we can use this model to describe the head-to-head linking of vinyl chloride units:



whereas the head-to-tail linking that is present in polyvinyl chloride is described by the grammar with rules $S \rightarrow Sa$, $S \rightarrow a$ and only S as a non-terminal.

This model abstracts the initiation step into the start symbol, the propagation steps into the non-terminals, and the termination steps into rules which do not involve non-terminals on the right hand side. These correspondences though indirect are obvious. The reactants themselves are suppressed from explicit appearance except through the production rules which 'pick and add' them to the growing chain. The production rules embody the chemical affinities between the various units. By taking such a 'chunked view' which suppresses the details of the chemical reactions, we are able to see the emerging patterns more clearly and with less obfuscating detail.

This linguistic view comes to be of additional significance when systems such as the nucleic acids which bear information in their base sequences are considered. In the DNA double chain the base pairs cytosine and guanine, and adenine and thymine, always occur paired; for simplicity let us denote the phosphate-sugar-base unit of the DNA strand by the symbols c,g, a and t depending on the base involved. Then the'grammar'

$$G = (N,T,P,S)$$

where $N = \{S\}$, $T = \{c,g,a,t\}$, and

$$P = \{ S \rightarrow gSc,\quad S \rightarrow cSg,$$
$$S \rightarrow aSt,\quad S \rightarrow tSa,$$
$$S \rightarrow gc,\quad S \rightarrow cg$$
$$S \rightarrow at,\quad S \rightarrow ta \}.$$

generates all possible DNA double strands. Our grammar provides the form of DNA, but not the'meaning' or information contained therein. Now it is well-known that linguistic studies have syntactic as well as semantic aspects, where the syntax describes the form, and semantics, the meaning. It is to be hoped that application of semantic techniques will lead, for example, to considerations regarding the genetic code. Attempts in this direction are in progress.

It will be noticed that only the type 3 and type 2 grammars have been used for our modelling purposes so far. This situation corresponds to the case with programming

languages, where again the type 3 and type 2 grammars have the most utility for specification purposes.

ACKNOWLEDGEMENT

The author wishes to thank Dr. A.P.B. Sinha for his interest and encouragement in the present work.

BIBLIOGRAPHY

1. S. Wolfram, Rev. Mod. Phys. 55, 601-644 (1983) and references therein.
2. J.E. Hopcroft and J.D. Ullmann, 'Formal languages and their relation to Automata', (1969), Addison-Wesley, Reading, Mass.