N. CHOMSKY and M. P. SCHÜTZENBERGER. *The algebraic theory of context-free languages.* ***Computer programming and formal systems,*** edited by P. Braffort and D. Hirschberg, Studies in logic and the foundations of mathematics, North-Holland Publishing Company, Amsterdam 1963, pp. 118–161.

The first section of this paper sketches the linguistic motivation for studying the properties of context-free languages. I'll say nothing about this section, except that it is quite straightforward, of minimal interest to most readers of this JOURNAL, and of no interest to linguists, who are acquainted with papers which go into this topic in detail.

Section 2 (p. 124) deals with the relationship between the representation of languages in terms of formal power series in a finite number of non-commutative variables (hereafter fps) and their representation in terms of grammars, i.e., generative processes. An fps r is presented as a summation $\Sigma_i(r, f_i)f_i$, where $f_1, f_2, \cdots$ is an enumeration of all strings in the variables, and (r, f_i) is the coefficient of the string f_i in the fps r. Two fps's r and r' are said to be equivalent mod degree n, i.e., $r \equiv r' \pmod{\deg n}$ if $(r, f) = (r', f)$ for every string f of length $\leqq n$. Suppose we have an infinite sequence of fps's $r_1, r_2, \cdots$ such that for each n and each $n' > n$, $r_n \equiv r_{n'} \pmod{\deg n}$. Then the limit of this sequence is $r_\infty = \lim \pi_n r_n$ where for each n, $\pi_n r_n$ is the polynomial formed from r_n by replacing all coefficients of strings of length $> n$ by zero.

A grammar with the initial symbol α_1 and the rules $\alpha_i \to \sigma_{i,j}$ $(1 \leqq i \leqq n, 1 \leqq j \leqq m_i)$ is associated with the set of polynomial expressions

$$\alpha_i = \sum_{j=1}^{m_i} \varphi_{i,j}$$

for $1 \leqq i \leqq n$. Each of these polynomials can be regarded as a mapping ψ_i which carries an n-tuple of fps's $(r_1, \cdots, r_n)$ into the n-tuple obtained by replacing α_k by r_k in

$$\sum_{j=1}^{m_i} \varphi_{i,j}.$$

The generative process itself can be thought of as an infinite sequence of n-tuples of fps's: $(r_{0,1}, \cdots, r_{0,n})$, $(r_{1,1}, \cdots, r_{1,n})$, $(r_{2,1}, \cdots, r_{2,n}), \cdots$ where each $r_{0,i} = 0$, i.e., the fps in which all the coefficients are zero, and where for each i and $j > 0$, $r_{j,i} = \psi_i(r_{j-1,1}, \cdots, r_{j-1,n})$. It is then the case that for each i, j, k, such that $0 < j < k$ and $1 \leqq i \leqq n$, $r_{j,i} \equiv r_{k,i} \pmod{\deg j}$. The solution to the set of polynomials associated with the grammar is defined as the limit of this sequence of n-tuples, i.e., $(r_{\infty,1}, \cdots, r_{\infty,n})$, and any term of a solution to a set of polynomials all of whose coefficients are positive is called a context-free fps. The grammar can now be said to generate the fps $r_{\infty,1}$, and the coefficient of each term in this fps is the number of structural descriptions that the grammar assigns to the string, i.e., the degree of ambiguity of the string.

Both the set of all fps's and the set of context-free fps's form rings closed under the operations: multiplication by an integer, addition, and multiplication. The support of an fps is defined as the set of its strings which have non-zero coefficients. Thus, the support of a context-free fps is the language generated by the corresponding grammar, and the operations addition and multiplication performed on a given pair of fps's result in fps's whose supports are the union and product, respectively, of the supports of the given pair of fps's. (Hereafter the terms "grammar" and "language" are to be taken to mean "context-free grammar" and "context-free language," respectively.)

Finally, an algebraic fps is defined as a term of the solution to a set of polynomials in which there may be negative coefficients. In such an fps, both the set of strings with positive coefficients and the set with negative coefficients are languages; thus the support of an algebraic fps is a language, which can be regarded as the set of strings which are not assigned the same number of structural descriptions by the grammars generating the positive and negative languages.

Section 3 (p. 131) deals further with algebraic fps's. Algebraic fps's, but not context-free fps's, are closed under the operation, Hadamard product of a pair of fps's, and the corresponding operation on their supports is intersection. In addition, relationships between algebraic fps's and classical analysis are discussed.

Section 4 (p. 135) deals with the generative properties of various types of grammars, these types defined by placing restrictions on the form of their rules. The closure properties, inclusion properties, and several other relationships of the sets of languages generated by these several types are also discussed. These topics are discussed again in section 8 (p. 154), which deals with finite transduction and outlines the proof that the intersection of a language of type i and a regular event results in a language of type i.

An alternative characterization of language is given in section 5 (p. 144). The authors first define two subsets of languages: standard regular events and Dyck languages, which constitute subsets of regular events and languages, respectively. The important results of this section are: Any regular event R can be characterized by a homomorphism θ and a standard regular event A, such that $R = \theta A$. Any language L can be characterized by a homomorphism θ, a standard regular event A, and a Dyck language D, such that $L = \theta(A \cap D)$. Various other interesting types of languages including those of the previous section are obtained by placing natural kinds of restrictions on A and D.

Sections 6 (p. 148) and 7 (p. 153) deal with several undecidability theorems of grammars—all based on Post's correspondence problem—and ambiguity. The authors show that there exist inherently ambiguous languages: Every grammar that generates such a language assigns more than one structural description to some of its strings. And it turns out that most interesting questions about grammars are algorithmically unsolvable.

Section 9 (p. 156) brings out several connections between the theory of context-free languages and the theory of automata. G. H. MATTHEWS

DAVID A. HUFFMAN. *Canonical forms for information-lossless finite-state logical machines.* ***Sequential machines, Selected papers,*** edited by Edward F. Moore, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, Palo Alto, and London, 1964, pp. 132–156. (Reprinted from ***IRE transactions on circuit theory,*** vol. CT-6, special supplement (1959), pp. 41–59.)

A combinational circuit is a finite-state circuit with one state only (i.e., without memory). A composition of combinational circuits and delay elements is a sequential circuit. Such a circuit is said to be information-lossless if there exist no two (not necessarily different) states s_i and s_f and no two different input sequences x and x' of equal length and output sequence y such that both x and x' lead from s_i to s_f and yield y.

The author considers several types of information-lossless circuits and tries to give canonical forms (block schemata) into which circuits of these types can be synthesized. He also gives schemata for the inverses of such circuits. Many statements are made in the paper but, in fact, none of them is either formulated or proved in a formal way. The paper seems to be of more interest to engineers than to mathematicians. ANDRZEJ J. BLIKLE

IRVING M. COPI, CALVIN C. ELGOT, and JESSE B. WRIGHT. *Realization of events by logical nets.* Ibid., pp. 175–192. (Reprinted from ***Journal of the Association for Computing Machinery,*** vol. 5 (1958), pp. 181–196.)

In the introduction to the paper under review the authors state the following: "In [XXIII 59], S. C. Kleene obtains a number of interesting results. The most important of these, his analysis and synthesis theorems (theorems 5 and 3), are obscured both by the complexity of his basic concepts and by the nature of the elements used in his nets. In this paper we give a new formulation and new proofs of Kleene's analysis and synthesis theorems, in which we presuppose no acquaintance with Kleene's work. We use simpler basic concepts and construct our nets out of more familiar and convenient elements."

In fact these words characterize strictly the proper aim of the paper under review. The basic notions of this paper are: regular expression (re), regular set (rs), and net. Regular expressions are formulas consisting of letters from the alphabet of $n + 1$ letters $\Lambda, A_1, A_2, \cdots, A_n$ and three operator symbols $\vee$, $\cdot$, *, such that: (1) each single letter is a re, (2) if Ω_1 and Ω_2 are re, then so are $(\Omega_1 \vee \Omega_2)$, $(\Omega_1 \cdot \Omega_2)$, and Ω_1^*, and (3) nothing is a re unless its being so follows from these rules.

The regular sets are all sets of finite strings of $A_1, A_2, \cdots, A_n$ which can be represented by