



---

Soviet-era science, translated into English

# CYBERNETICS AND CONTROL THEORY

1965

SovietRxiv

---

View the original and related papers at <https://sovietrxiv.org/items/ru-196501.56986>

Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.

**Abstract**

**Full Text**

## **CYBERNETICS AND CONTROL THEORY**

**M. A. SPIVAK**

### **DECOMPOSITION OF A REGULAR EXPRESSION WITH RESPECT TO A BASIS AND ITS APPLICATIONS**

*(Presented by Academician V. S. Kulebakin, 3 XII 1964)*

The language of regular expressions <sup>(1)</sup> is at present one of the most widespread languages for recording the operating conditions of an automaton. We shall consider the language of regular expressions supplemented by the symbols  $\cap$  and  $'$ , corresponding to intersection and complementation of events. Such an extension of the language of regular expressions makes it considerably more convenient <sup>(2)</sup>. Unfortunately, the known algorithm of abstract synthesis for the language of regular expressions, described in <sup>(1, 2)</sup>, does not carry over directly to the extended language of regular expressions. The synthesis method for the extended language of regular expressions described at the end of <sup>(2)</sup> is not an algorithm and is associated with considerable difficulties. In the present note a simple algorithm of abstract synthesis for the extended language of regular expressions is indicated. It is based on the decomposition of a regular expression with respect to a certain system of regular expressions, called a basis. An application of decomposition with respect to a basis to the transformation and simplification of regular expressions is also indicated. We shall use the terminology of article <sup>(1)</sup>.

Let  $X = \{x_1, \dots, x_n\}$  be an arbitrary finite set. A regular expression in the alphabet  $X$  is defined inductively:

- 1)  $x_1, \dots, x_n, e, \Phi$  are regular expressions.
- 2) If  $E$  and  $F$  are regular expressions, then  $(E) \cup (F)$ ,  $(E) \cap (F)$ ,  $(E)'$ ,  $(E) \cdot (F)$ ,  $\{E\}$  are also regular expressions.
- 3) There are no other regular expressions.

We shall interpret the symbols  $x_\alpha, e, \Phi$  as elementary events, and the symbols  $\cup, \cap, ', \cdot, \{ \}$  as operations on events. Then every regular expression  $E$  in the alphabet  $X$  defines some event in this alphabet, which we shall denote by the same letter  $E$ . We shall agree that the notation  $E \equiv F$  means graphical identity of the regular expressions  $E$  and  $F$ , while the notation  $E = F$  means equality of the corresponding events.

Introduce the notation

$$\chi(E) = \begin{cases} e, & \text{if } e \in E, \\ \Phi, & \text{if } e \notin E. \end{cases}$$

A finite system of regular expressions  $E_1, \dots, E_m$  in the alphabet  $X$  will be called a **basis** if

$$E_i = x_1 E_{i1} \cup \dots \cup x_n E_{in} \cup \chi(E_i) \quad (i = 1, \dots, m), \quad (1)$$

where each  $E_{i\alpha}$  ( $i = 1, \dots, m$ ;  $\alpha = 1, \dots, n$ ) is one of the regular expressions  $E_j$  ( $j = 1, \dots, m$ ). The equalities (1) will be called the **equations of the given basis**.

We shall indicate an algorithm which, for every regular expression  $E$ , gives a basis including this regular expression and the equations of this basis. The construction of the basis is carried out inductively in accordance with the inductive definition of a regular expression.

- 1) It is easy to see that the regular expression  $\Phi$  itself forms a basis; the system of two regular expressions  $e, \Phi$  is a basis; the system of three regular expressions  $x_a, e, \Phi$  is a basis.
- 2) Let the regular expression  $E \equiv E_1$  be included in the basis  $E_1, \dots, E_m$  with equations (1), and let the regular expression  $F \equiv F_1$  be included in the basis  $F_1, \dots, F_k$  with equations

$$F_j = x_1 F_{j1} \cup \dots \cup x_n F_{jn} \cup \chi(F_j) \quad (j = 1, \dots, k). \quad (2)$$

Then we have:

$$E_i \cup F_j = x_1 (E_{i1} \cup F_{j1}) \cup \dots \cup x_n (E_{in} \cup F_{jn}) \cup \chi(E_i \cup F_j), \quad (3)$$

$$E_i \cap F_j = x_1 (E_{i1} \cap F_{j1}) \cup \dots \cup x_n (E_{in} \cap F_{jn}) \cup \chi(E_i \cap F_j), \quad (4)$$

$$E'_i = x_1 E'_{i1} \cup \dots \cup x'_n E'_{in} \cup \chi(E'_i), \quad (5)$$

where  $i = 1, \dots, m$ ;  $j = 1, \dots, k$ . These equalities show that the regular expressions  $E_i \cup F_j$  form a basis with equations (3), including the regular expression  $E \cup F$ ; the regular expressions  $E_i \cap F_j$  form a basis with equations (4), including the regular expression  $E \cap F$ ; and the regular expressions  $E'_i$  form a basis with equations (5), including the regular expression  $E'$ .

Consider the regular expressions  $G_{i j_1 \dots j_l} = E_{iF} \cup F_{j_1} \cup \dots \cup F_{j_l}$ , where  $i = 1, \dots, m$ , and  $j_1 \dots j_l$  is an arbitrary combination of indices  $1, \dots, k$ . Let  $e \notin E_i$ ; then from (1) and (2) we obtain:

$$G_{i j_1 \dots j_l} = x_1(E_{i1}F \cup F_{j_{11}} \cup \dots \cup F_{j_{l1}}) \cup \dots \\ \dots \cup x_n(E_{in}F \cup F_{j_{1n}} \cup \dots \cup F_{j_{ln}}) \cup \chi(G_{i j_1 \dots j_l}).$$

If, however,  $e \in E_i$ , we find:

$$G_{i j_1 \dots j_l} = x_1(E_{i1}F \cup F_{11} \cup F_{j_{11}} \cup \dots \cup F_{j_{l1}}) \cup \dots \\ \dots \cup x_n(E_{in}F \cup F_{1n} \cup F_{j_{1n}} \cup \dots \cup F_{j_{ln}}) \cup \chi(G_{i j_1 \dots j_l}).$$

These equalities show that the regular expressions  $G_{i j_1 \dots j_l}$  form a basis including the regular expression  $EF$ .

Further, from (1) one can obtain:

$$\{E\} = x_1 E_{11} \{E\} \cup \dots \cup x_n E_{1n} \{E\} \cup e.$$

Consider the regular expressions  $G_{i_1 \dots i_l} = (E_{i_1} \cup \dots \cup E_{i_l}) \{E\}$ , where  $i_1 \dots i_l$  is an arbitrary combination of indices  $1, \dots, m$ . Let  $e \notin E_{i_1} \cup \dots \cup E_{i_l}$ ; then

$$G_{i_1 \dots i_l} = x_1(E_{i_{11}} \cup \dots \cup E_{i_{l1}}) \{E\} \cup \dots \cup x_n(E_{i_{1n}} \cup \dots \cup E_{i_{ln}}) \{E\}.$$

If, however,  $e \in E_{i_1} \cup \dots \cup E_{i_l}$ , then

$$G_{i_1 \dots i_l} = x_1(E_{i_{11}} \cup \dots \cup E_{i_{l1}} \cup E_{11}) \{E\} \cup \dots \\ \dots \cup x_n(E_{i_{1n}} \cup \dots \cup E_{i_{ln}} \cup E_{1n}) \{E\} \cup e.$$

These equalities show that the regular expressions  $\{E\}, G_{i_1 \dots i_l}$  form a basis.

Let  $E_1, \dots, E_m$  be a basis with equations (1). Consider the Moore automaton with input alphabet  $X$ , states  $E_1, \dots, E_m$ , output signals  $e, \Phi$ , and transition and marking functions defined by the formulas:

$$\delta(E_i, x_a) \equiv E_{ia}, \quad \mu(E_i) = \chi(E_i). \quad (6)$$

**Theorem.** *The automaton (6) represents every event  $E_i$  ( $i = 1, \dots, m$ ) from the initial state  $E_i$  with output signal  $e$ .*

**Proof.** Denote by  $\delta(E_i, p)$  the state of automaton (6) into which it passes from the state  $E_i$  under the action of the input word  $p$ . Then we must prove the equivalence of the assertions  $p \in E_i$  and  $e \in \delta(E_i, p)$ . For  $p = e$  the assertion is obvious. Suppose that it is true for all words of length  $k$ , and consider the word  $xp$ , where  $p$  is a word of length  $k$ . From (1) and (6) it follows that  $xp \in E_i$  is equivalent to  $p \in \delta(E_i, x)$ . But, by assumption, this is equivalent to

$$e \in \delta(\delta(E_i, x), p) \equiv \delta(E_i, xp).$$

From the theorem proved it follows that the states  $E_i$  and  $E_j$  in automaton (6) are Moore-equivalent if and only if  $E_i = E_j$ , and simply equivalent if and only if the events  $E_i$  and  $E_j$  differ only by the empty word. Hence there follows a simple way of obtaining an automaton with the least number of states among all Moore or Mealy automata representing the given event.

The decomposition of the regular expression  $E$  with respect to a basis makes it possible to transform this expression so that it contains no symbols of intersection and complementation. To this end consider the system of equations

$$E_i = A_{i1}E_1 \cup \dots \cup A_{im}E_m \cup B_i \quad (i = 1, \dots, m), \quad (7)$$

where  $A_{ij}, B_i, E_i$  are events in the alphabet  $X$ ;  $A_{ij}, B_i$  are given, and  $E_i$  are the unknowns. This system differs only insignificantly from the system of equations considered in (3). It is easy to see that all the results of (3), with the corresponding changes, remain valid for system (7). In particular: 1) if  $e \notin A$ , then the equation  $E = AE \cup B$  has a unique solution, expressed by the formula  $E = \{A\}B$ ; 2) if system (7) has a unique solution, then it can be found by the method of successive elimination of the unknowns.

Let now  $E_1, \dots, E_m$  be a basis with equations (1). These equations may be regarded as a special case of equations (7). By the method (3) it is easily verified that system (1) has a unique solution. Consequently, it can be solved by successive elimination of the unknowns, as a result of which we arrive at regular expressions containing no intersection or complementation.

Saratov State University  
named after N. G. Chernyshevsky

Received  
2 XII 1964

## REFERENCES

1. V. M. Glushkov, UMN, 16, No. 5 (1961).

2. R. F. McNaughton, H. Yamada, IRE Trans. EC-9, No. 1 (1960).
3. V. G. Bodnarchuk, Zhurn. vychislit. matem. i matem. fiz., 3, No. 6 (1963).

*Note: Figure translations are in progress. See original paper for figures.*

*Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.*