ON OPERATORS REALIZABLE IN LOGICAL NETWORKS

1957

SovietRxiv

Abstract

Full Text

MATHEMATICS

B. A. TRAKHTENBROT

ON OPERATORS REALIZABLE IN LOGICAL NETWORKS

(Presented by Academician M. V. Keldysh, 26 IX 1955)

1. Let X, Z, and Q be three finite sets whose elements we shall call input, output, and auxiliary letters, respectively. The sets themselves will be called the input, output, and auxiliary alphabets; let m, n, k be the corresponding numbers of letters in them.

By X^t we shall denote the set of all **input sequences** of the form:

$$x(1),\ x(2), \dots,\ x(t), \dots, \tag{1}$$

whose elements are input letters. The notation Z^t, Q^t has the analogous meaning. For each natural number μ, by X^μ we denote the set of all **input words** of length μ:

$$x(1),\ x(2), \dots,\ x(\mu).$$

The notation Z^μ, Q^μ has the analogous meaning.

The objects of the subsequent discussion are operators mapping X^t into Z^t (or X^μ into Z^μ) **without anticipation**. This means that the i-th letter of the output sequence (or of the output word) is determined uniquely by the first i letters of the input sequence to which it corresponds, and does not depend on the subsequent letters. Obviously, an operator without anticipation mapping X^t into Z^t induces, for any natural μ, an operator without anticipation mapping X^μ into Z^μ, namely: the initial word of length μ of the input sequence is transformed into the initial word of length μ of the output sequence. Among such operators are the operators realizable in logical networks in the sense of Burks and Wright ([4]) (see Fig. 1).

At each moment of time t ($t = 1, 2, \dots$)* some input letter $x(t)$ enters the truth block L through the input channel I, and through the feedback channel II from the delay cell Zd there enters an auxiliary letter $q(t-1)$, issued at the preceding moment of time through the output channel IV; the pair of letters $x(t)$ and $q(t-1)$ is transformed by the truth block L into the pair of letters $z(t)$ and $q(t)$, which are issued through the output channels III and IV. It is

Fig. 1

Figure 1: Fig. 1

assumed that at $t = 1$ the delay cell Zd already contains some initial auxiliary letter.

The corresponding operator can be specified by means of a system of equations of the form

$$\begin{aligned} z(t) &= \Phi[x(t), q(t-1)], \\ q(t) &= \Psi[x(t), q(t-1)], \\ q(0) &\text{ —an initial constant,} \end{aligned} \tag{2}$$

where $\Phi(\xi, \eta)$, $\Psi(\xi, \eta)$ are two-place functions defined for all $\xi \in X,\ \eta \in Q$.

* In the case of a mapping X^μ into Z^μ, t assumes values only up to μ.

If an operator admits a representation of the form (2) with an auxiliary alphabet Q of k letters and does not admit such a representation with an auxiliary alphabet of smaller cardinality, then we shall say that it has weight k. If an operator without anticipation is defined on X^μ, then it has, as is easy to see, finite weight; if, however, it is defined on X^t, then, generally speaking, it is not representable in the form (2) and cannot be assigned a finite weight.

2. Let us note that if letters are represented in binary code, then $\log_2 m$ and $\log_2 n$ binary channels, respectively, are required to input an input letter into the truth block and to output an output letter from it. In addition, $\log_2 k$ binary feedback channels will be required, and hence also $\log_2 k$ delay cells, each of which delays one binary digit (one unit of information) by one beat (one unit of time); for this reason it is natural to take $\log_2$ of the weight of an operator as the measure of the operator' s memory.

Fig. 1

Let us call the volume of a word from X^μ the product $\mu \cdot \log_2 m$; obviously, the volume of a word characterizes its quantity of information. If an operator of weight k is defined on X^μ, then the ratio $\log_2 k/(\mu \cdot \log_2 m)$ of the measure of its memory to the volume of the words being processed is naturally called the specific memory of this operator. It is easy to see that for any operator without anticipation defined on X^μ, the inequality

$$\log_2 k \leq \mu \cdot \log_2 m, \tag{3}$$

always holds, having the simple meaning that the measure of the memory of an operator does not exceed the volume of the word processed by it, i.e. the specific memory of the operator does not exceed 1; moreover, it may be arbitrarily close to zero or may even be equal to zero. (The latter, obviously, occurs in the case when the i-th letter of the output word depends only on the i-th letter of the input word and does not depend on the letters preceding it.)

The following theorem shows, however, that in processing long words, in the overwhelming majority of cases one encounters operators with high specific memory.

Theorem 1. *For any $\varepsilon > 0$, among the operators without anticipation defined on X^μ, the fraction of those for which the specific memory is less than $1 - \varepsilon$ tends to zero as $\mu \to \infty$.*

3. Operators realizable in logical networks possess certain periodicity properties that have already been noted in the literature (1^{-4}). The theorem of this section and its corollary refine these known results.

For a periodic sequence

$$x(1),\ x(2), \dots,\ x(p)\ (x(p+1), \dots,\ x(p+r)) \tag{4}$$

we shall say that it has period r and reduced length $p + r$.

Theorem 2. *An operator of weight k has the following property: it transforms any periodic sequence with period r and reduced length $p + r$ into a periodic sequence with period $\leq k \cdot r$ and with reduced length $\leq p + k \cdot r$.*

The estimate formulated in the theorem is exact in the sense that, for any k and r, one can easily construct an operator of weight k that transforms some sequence with period r into a sequence with period $k \cdot r$.

An operator is called constant if it maps every sequence from X^t (or word from X^μ) into one and the same output sequence.

···activity (into one and the same word) (i.e., it produces a fixed sequence).

From Theorem 2 and the remark made above there follows a corollary.

Corollary. A constant operator of weight k produces an output periodic sequence whose reduced length is exactly equal to k.

It is also easy to show that every periodic sequence with reduced length k can be produced by some constant operator of weight k. At the same time, one can construct an operator without anticipation which has the property indicated in Theorem 1 (for $k = 2$), but has no finite weight, i.e., is not representable by equations of the form (2).

4. The relation between operators defined on X^t and on X^μ is clarified in the following proposition.

Theorem 3. *In order that two operators of weight $\leq k$, defined on X^t, coincide, it is necessary and sufficient that the operators induced by them on X^{2k-1} coincide.*

This estimate cannot be improved; for any k one can construct a pair of distinct operators of weight k which coincide on all initial input words of length $2k-2$.

5. The theorems of the present note were established by the author in the course of his joint work with Prof. N. E. Kobrinskii in the area of analysis and synthesis of logical networks. In particular, Theorem 3 proves useful in the synthesis of logical networks (for example, multicycle relay-contact circuits), since it makes it possible, from a certain rough estimate of the weight of the operator being synthesized, to determine its exact weight and thereby makes it possible to minimize the amount of memory in the logical network.

Penza State Pedagogical Institute
named after V. G. Belinsky,

Received
22 IX 1956

REFERENCES

1. V. I. Shestakov, *Avtomat. i telemekh.*, **15**, No. 2, 107 (1954).
2. V. I. Shestakov, *Avtomat. i telemekh.*, **15**, No. 4, 310 (1954).
3. V. I. Shestakov, DAN, **99**, No. 6 (1954).
4. A. Burks, J. Wright, *Proc. Inst. Rad. Eng.*, **41**, 10 (1953).

* The author has learned that the article by E. F. Moore, “Gedanken-experiments on sequential machines,”included in the volume *Automata* (a Russian translation is in press), contains a result analogous to Theorem 3.

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

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