ERROR CORRECTION IN AUTOMATA WHOSE COMBINATORIAL PARTS ARE REALIZED WITH THE HELP OF ORTHOGONAL SERIES EXPANSIONS

UDC 62-50

M. G. Karpovskii

A method is proposed for realizing an automaton by representing its excitation function in the form of finite orthogonal series, as is an error-correction method for this type of automata, based on the use of a system of arithmetic codes.

1. A finite automaton whose states and input signals are coded by binary strings of respective lengths m and k is described by a system of Boolean functions (excitation functions)

$$b^{(i)} = f^{(i)}(x^{(0)}, \ldots, x^{(k-1)}, a^{(0)}, \ldots, a^{(m-1)}) \quad (i = 0, 1, \ldots, m-1). \tag{1}$$

Here, $x^{(i)}$, $a^{(j)}$, and $b^{(s)}$ are the binary components of the code strings of the input signal and the states in the previous and present cycle.

System (1) can be realized by presenting it in the form of an orthogonal series [1-3]. For this, we let

$$z = \sum_{j=0}^{m-1} a^{(j)} 2^{m-1-j} + \sum_{j=0}^{k-1} x^{(j)} 2^{m+k-1-j}, \quad b = \sum_{j=0}^{m-1} b^{(j)} 2^{m-1-j}. \tag{2}$$

Then, system (1) can be represented by the function

$$b = f(z). \tag{3}$$

We extend $f(z)$ to the piecewise constant $\Phi(z)$:

$$\Phi(z) = f(\delta) \text{ when } z \in [\delta, \delta + 1) \tag{4}$$

and represent $\Phi(z)$ in the form of a series

$$\Phi(z) = \sum_{j=0}^{\infty} c_j \psi_j(z), \tag{5}$$

where $\{\psi_j(z)\}$ is a complete system of basic orthogonal functions.

It was shown in [1, 3] that it is desirable to choose, as the $\{\psi_j(z)\}$, a system of Walsh or of Haar functions, so that series (5) would contain no more than 2^{m+k} initial terms, and the basis functions would assume the values 0 and 1, or 0, +1, −1.

With this, expression (5) defines a method of realizing the combinatorial part of the automaton whose basic advantages are as follows:

1) the structure of the combinatorial part does not depend on the automaton, and is specified beforehand, the minimization of complexity of the combinatorial part reduces to the minimization of the number of nonzero coefficients of series (5), and can be solved by analytic methods, thus almost completely eliminating exhaustive trials [1];

Leningrad. Translated from Avtomatika i Telemekhanika, No. 9, pp. 204-207, September, 1971. Original article submitted October 27, 1970.

TABLE 1

a	x			
	x_0	x_1	x_2	x_3
a_0	a_0	a_3	a_3	a_0
a_1	a_5	a_3	a_7	a_1
a_2	a_5	a_3	a_7	a_1
a_3	a_0	a_5	a_3	a_5
a_4	a_1	a_6	a_7	a_0
a_5	a_5	a_4	a_3	a_5
a_6	a_0	a_4	a_7	a_6
a_7	a_1	a_3	a_7	a_0

TABLE 2

a	Code for a		
	$a^{(0)}$	$a^{(1)}$	$a^{(2)}$
a_0	0	0	0
a_1	0	1	1
a_2	0	1	0
a_3	0	0	1
a_4	1	0	0
a_5	1	1	0
a_6	1	1	1
a_7	1	0	1

TABLE 3

x	Code
	$x^{(0)}$
x_0	0
x_1	0
x_2	1
x_3	1

2) it is comparatively simple to solve the problem of seeking an optimal redefinition of the excitation function for partially determined automata [3];

3) the block diagram of the device realizing the automaton contains standard blocks for the computing devices, which simplifies its realization, increases its level of engineering sophistication and, in particular, increases the possibilities of microminaturizing the device and of realizing it on large-scale integrated devices or by homogeneous structures;

4) arithmetic codes can be effectively used for error correction, making it possible to organize constant modular inspection in the system containing the computer's arithmetic device.

2. We now consider the question of detecting and correcting errors in automata synthesized by the method described above. As in [4], we shall assume that the input signal is not subjected to the influence of errors. Since the computation of the value of the excitation function is performed on a summer (which computes the sum of series (5)), we shall use, for error-correcting in the automaton, an arithmetic (AN + i)-code [5, 6].

Let the automaton A_0 to be synthesized by specified by a table (graph) of transitions, and let $\lambda = \{\lambda_0, \lambda_1, \ldots, \lambda_{n_\lambda - 1}\}$ be some partition of the set of its input signals. We construct an automaton A_λ, equivalent to A_0, by decomposing each state of A_0 into equivalent states such that A_λ is found in each of its states only when the signals at its input fall into one and the same block of partition λ. The states in which automaton A_λ can be found when the input signals belong to block λ_s will be called λ_s-attainable. The process of constructing redundant automaton A_λ from initial automaton A_0 is described in detail in [4]. For correcting errors in A_λ, it suffices to code the λ_s-attainable ($s = 0, 1, \ldots, n_\lambda - 1$) states of A_λ by the elements of the corresponding (AN + s)-code for which the modulus A provides the required correction capability. With this, the requisite redundancy for the given modulus A depends on the concrete features of the mapping realized by automaton A_0 and, for any multiplicity of errors to be corrected, there exists a class of automata for which correction of errors of the given multiplicity does not require redundant memory elements (see the example below).

A formal description of our method of correcting errors; the methods of choosing modulus A and of coding the states of the original automaton A_0; the methods of finding an optimal partition λ of the set of input signals; necessary and sufficient conditions for the existence of automata correcting, by the method described, errors of a given multiplicity, realizing a given automaton mapping and having a given memory size; as well as methods of constructing the decoding device – all these can be obtained analogously to what one does in realizing the combinatory parts of automata by the usual methods of Boolean algebra [4, 7].

Example. Consider automaton A_0 with the transitions shown in Table 1 (here, a_i ($i = 0, 1, \ldots, 7$) and x_l ($l = 0, 1, 2, 3$) are the internal states and input signals for A_0). We assume that A_0 is realized by binary elements and that it is necessary to detect single errors. Optimal for the given automaton is, for example, the partition for which $\lambda = \{\lambda_0, \lambda_1, \lambda_2\}$ and $\lambda_0 = \{x_0, x_3\}$, $\lambda_1 = \{x_1\}$, $\lambda_2 = \{x_2\}$, ($n_\lambda = 3$).

Automaton A_λ coincides with A_0. For the detection of a single error in the given case we can set A = 3; the coding of the states is given in Table 2. In the given case the detection of a single error would not require increasing the memory size. The coding of the input signals is given in Table 3. The values of function $f(z)$, which is constructed by Eqs. (2)-(4) and which corresponds to the system of excitation functions for the given coding, is given in Table 4. Here are given the binary codes of the coefficients in the expansion $c_p^{(q)}$ ($p = 0, 1, \ldots, m + k - 1$; $q = 1, 2, \ldots, 2^p$) ($m = 3$, $k = 2$) corresponding to the piecewise constant $\Phi(z)$ [cf., (4)] with a Haar basis [1] (with $c_p^{(q)}$ being represented by function $c(r)$, where $r = 2^p + q - 1$). As is obvious from Table 4, for the computation of the values of the automaton's excitation function with single errors being detected, it is necessary, in the given case, to store 11 nonzero coefficients of which only 6 are pairwise distinct.

TABLE 4

z, r	$f(z)$	$c(r)$ 2^{-2}	2^{-1}	2^0	2^1	s*	z, r	$f(z)$	$c(r)$ 2^{-2}	2^{-1}	2^0	2^1	s	z, r	$f(z)$	$c(r)$ 2^{-2}	2^{-1}	2^0	2^1	s	z, r	$f(z)$	$c(r)$ 2^{-2}	2^{-1}	2^0	2^1	s
0	0	0	0	1	1	0	8	1	0	0	1	1	1	16	2	0	0	0	0	0	24	0	0	0	0	0	0
1	0	1	0	0	0	1	9	1	0	0	0	0	0	17	2	0	0	0	0	0	25	6	0	0	0	0	0
2	6	1	0	0	0	0	10	1	0	0	0	0	0	18	5	0	0	0	0	0	26	3	0	0	0	0	0
3	6	1	0	0	0	0	11	1	0	0	0	0	0	19	5	0	0	1	1	0	27	3	0	0	0	0	0
4	3	0	0	0	0	0	12	7	0	1	1	0	1	20	5	0	0	0	0	0	28	0	0	0	1	1	1
5	3	0	1	1	0	1	13	1	0	1	1	0	0	21	5	0	0	0	0	0	29	0	0	0	0	0	0
6	6	0	0	0	0	0	14	4	0	0	0	0	0	22	2	0	0	1	1	0	30	6	0	0	0	0	0
7	0	0	0	0	0	0	15	4	0	0	1	1	1	23	5	0	0	0	0	0	31	6	0	0	0	0	0

* In the columns headed by the letter "s" are given the values of the sign bits of the coefficient codes.

3. The block schematic of the realization of the automaton whose excitation functions are synthesized in accordance with (5) contains a generator of the basis functions $\{\psi_i(z)\}$, a block for the storage of the coefficients c_i, and an adder (for computing the products $c_i\psi_i(z)$ a multiplying device is not required since, for any $z \in [0, 2^{m+k})$ and any i, $\psi_i(z) \in \{0, 1\}$ or $\psi_i(z) \in \{0, \pm 1\}$ for the bases in question). The automaton's memory elements can be combined with an adder of the accumulating type if summation of terms of the series is performed serially in time.

With this, the complexity of realization of an automaton is basically determined by the complexity of the block storing the coefficients, which it is natural to treat as a monotonically increasing function of the number of nonzero coefficients. In connection with this, in order to decrease the number of nonzero coefficients, one can use the methods described in [1] for completely specified automata, and the methods described in [3] for partially specified automata.

The use of arithmetic codes makes it possible to correct errors both in the adder itself and in the block of coefficient storage.

Fusion of the automaton memory with the adder on which one computes the sum of the series makes it possible to employ, for error-correction in the automaton, methods which are not based on the use of arithmetic codes.

The coefficients c_i of series (5) are computed by the formula

$$c_i = \frac{1}{2^{m+k}} \sum_{z=0}^{2^{k+m}-1} \Phi(z)\psi_i(z). \tag{6}$$

The binary codes of the coefficients and the numbers occurring in the summer have integral and fractional parts, and after completion of the computation of the excitation function the fractional part of the number in the adder must equal zero. This makes it possible to detect errors, comparatively simply and effectively, in the corresponding $\Delta(\Delta \leq m + k)$ bits of the summer, or to correct them by rounding up to the nearest integer the magnitude of the sum after the end of the computation. Because of this, it is desirable to use the arithmetic code only for correcting errors in the digits of the integral parts of the coefficient codes and in the corresponding digits of the summer. However, to detect errors in these digits one can use the overflow signals of the adder and the values of the sign bit after completion of the computation of the excitation functions.

LITERATURE CITED

1. M. G. Karpovskii and É. S. Moskalev, "Realization of systems of logical functions by means of expansion in orthogonal series," Avtomat. i Telemekhan., No. 12 (1967).
2. M. G. Karpovskii and É. S. Moskalev, "The use of autocorrelation characteristics for realization systems of functions of Boolean algebra," Avtomat. i Telemekhan., No. 2 (1970).
3. M. G. Karpovskii and É. S. Moskalev, "Realization of partially specified functions of Boolean algebra by means of expansions in orthogonal series," Avtomat. i Telemekhan., No. 8 (1970).