



$$q_i(t) = f[Q_1(t), Q_2(t), \dots, Q_n(t), x_1(t), x_2(t), \dots, x_m(t)]$$

$$z_j(t) = [Q_1(t), Q_2(t), \dots, Q_n(t), x_1(t), x_2(t), \dots, x_m(t)]$$

2.

$$k \geq \log_2 n$$

$$Q_1, Q_2, \dots, Q_k,$$

0 1.  
k -

$$2^k \quad n.$$

[1]:

- 1.
- 2.

$Q_i$

$$(0, \dots, 1)$$

(0,4)

1.  
 $Q_i$ ,  
0

(1,2,3)

1.

Табл.1

$$r(Q_1) = \{0,4; \overline{1,2,3}\}$$

k

$$2^{k-1}$$

$$2^n/2-1$$

$$2^{k-1}$$

Состояние элемент авт	$Q_1$	$Q_2$	$Q_3$
состояние автомата			
0	0	0	1
1	1	0	1
2	1	1	0
3	1	0	0
4	0	1	1

$Q_i$   $r_i$   $Q_i$

Табл.2

Состояния авт. \ ВХОДНОЙ СИГНАЛ	0	1	2	3	4
0	0	0	3	4	0
1	1	2	2	2	2

Табл.3

Состояния авт. \ ВХОДНОЙ СИГНАЛ	0	1	2	3	4
0	0	0	1	1	0
1	1	0	0	0	0

$$r_1 = \{0, 2; 1, 3, 4\},$$

$$r_1 = \{0, 2, 3; 1, 4\} \quad (3).$$

$$r_1 = \{0, 2, 3; 1, 4\} \quad \{0, 2; 1, 3, 4\}.$$

$(x_{a_i})$   $(x_{a_j})$   $a_i$   $a_j$

г.

$k$  :

$Q_i$

$r_i$

$Q_1, Q_2, Q_3$

$r_1, r_2, r_3$

$$\begin{aligned} Q_1: r_1 &= \{0, 2, 3; 1, 4\} \quad \{0, 2; 1, 3, 4\} \\ Q_2: r_2 &= \{0, 1, 2, 3, 4\} \quad \{1, 2; 0, 3, 4\} \\ Q_3: r_3 &= \{0, 2; 1, 3, 4\} \quad \{2, 3; 0, 1, 4\} \end{aligned}$$

$Q_1, Q_2, \dots, Q_k$

$k$

$k$

1.

$i, j$

$i, j$

2.

$i, j$

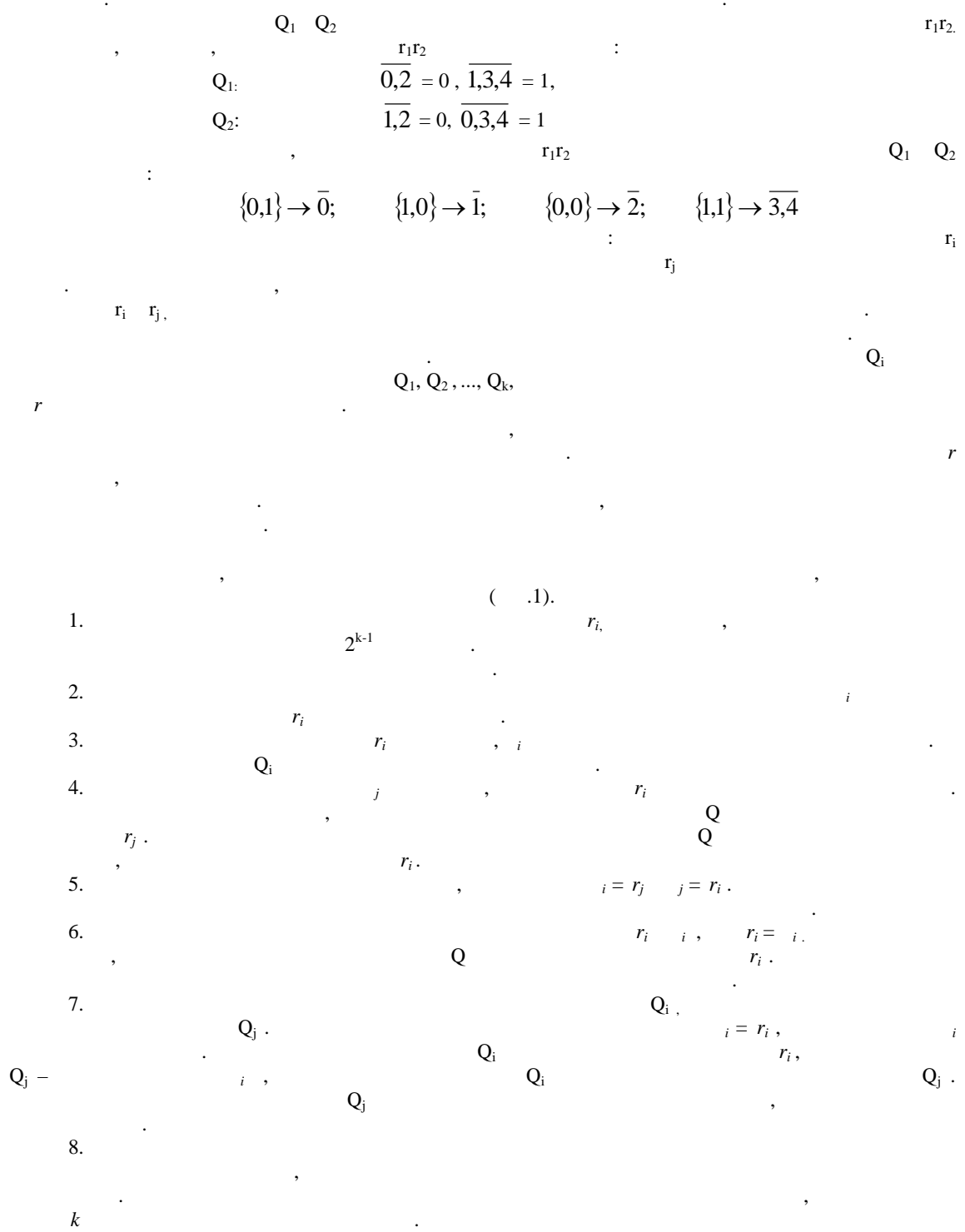
$j(i, j)$

$$2^{k-m}, \quad m -$$

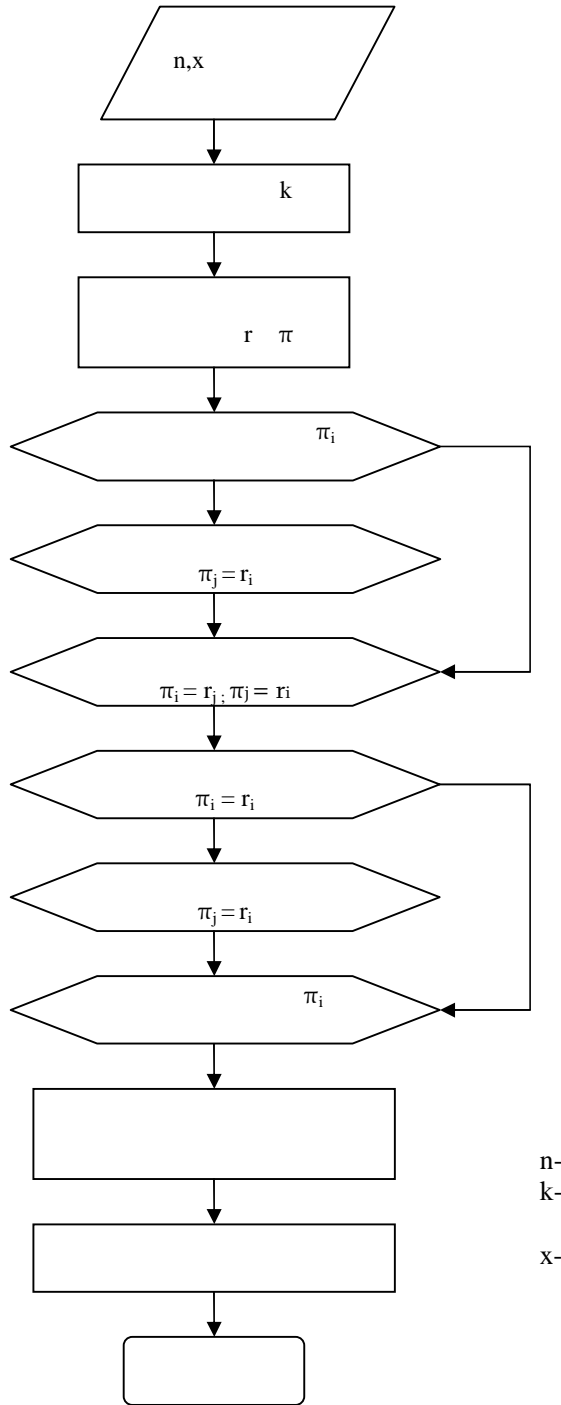
$Q_1, Q_2, \dots, Q_k$

$r_1, r_2, r_3$

$$r_1 r_2 = \{\overline{0}; \overline{1}; \overline{2}; \overline{3,4}\}; \quad r_1 r_3 = \{\overline{0}; \overline{2}; \overline{3}; \overline{1,4}\}; \quad r_2 r_3 = \{\overline{1}; \overline{2}; \overline{3}; \overline{0,4}\}; \quad r_1 r_2 r_3 = \{\overline{0}; \overline{1}; \overline{2}; \overline{3}; \overline{4}\}$$



$r$   $Q_1, Q_2, \dots, Q_k$



.1. -

Таблица переходов

Входной сигнал (x) \ Состояния	Состояния					
	0	1	2	3	4	5
0	1	2	4	5	0	0
1	1	3	5	5	0	0

Таблица выходов

Входной сигнал (x) \ Состояния	Состояния					
	0	1	2	3	4	5
0	y <sub>0</sub>	y <sub>0</sub>	y <sub>0</sub>	y <sub>0</sub>	y <sub>2</sub>	y <sub>2</sub>
1	y <sub>0</sub>	y <sub>0</sub>	y <sub>0</sub>	y <sub>0</sub>	y <sub>2</sub>	y <sub>1</sub>

D-

x(t)	Q <sub>1</sub> (t)	Q <sub>2</sub> (t)	Q <sub>3</sub> (t)	Q <sub>1</sub> (t+1)	Q <sub>2</sub> (t+1)	Q <sub>3</sub> (t+1)	z <sub>1</sub>	z <sub>2</sub>	q <sub>1</sub>	q <sub>2</sub>	q <sub>3</sub>
0	0	1	1	0	0	1	0	1	0	0	1
0	0	0	1	1	0	1	0	1	1	0	1
0	1	0	1	1	1	1	0	1	1	1	1
0	1	0	0	1	1	0	0	1	1	1	0
0	1	1	1	0	1	1	1	1	0	1	1
0	1	1	0	0	1	1	1	1	0	1	1
1	0	1	1	0	0	1	0	1	0	0	1
1	0	0	1	1	0	0	0	1	1	0	0
1	1	0	1	1	1	0	0	1	1	1	0
1	1	0	0	1	1	0	0	1	1	1	0
1	1	1	1	0	1	1	1	1	0	1	1
1	1	1	0	0	1	1	1	0	0	1	1

$$q_1 = \overline{Q_2}, q_2 = Q_1, q_3 = \overline{xQ_3} \vee Q_2, z_1 = Q_1Q_2, z_2 = \overline{xQ_2Q_3}$$

3.

(D - ).

++ .

1. „ » 1963
2. „ » 1989

3. . . . .  
« . . . . . ». 2005

### OPTIMAL ENCODING OF INTERNAL STATES OF FINITE AUTOMATA

Kartvelishvili O., Osipov M.  
Georgian Technical University

#### Summary

Presented article is considering the method of finite automata internal state encoding on the stage of structural synthesis. This method allows the simplification of corresponding switching and output functions, which leads to simplification of automata structure. Presented method is based on two consideration: the first consideration concerns to the decrease of the number of encoding options being sorted out; the second consideration includes the complexity evaluation methods of elementary automata, which does not require to find these functions. This approach uses state separation of given automata into groups determined by code of elementary automata, which gives possibility of to evaluate dependency of every subsequent state of elementary automata on the states of other elementary automatas. Based on the stated rules of the above mentioned dependency determination, automata encoding algorithm was created, which was researched using automatas with different transition tables.

### სასრულო ავტომატის შიგა მდგომარეობის კოდიფიკაცია

ო. ქართველიშვილი, მ. ოსიპოვი  
საქართველოს ტექნიკური უნივერსიტეტი

#### რეზიუმე

განხილვა სასრულო ავტომატის შიგა მდგომარეობის კოდიფიკაციის მეთოდი სტრუქტურული სინთეზის ეტაპზე, რომლის დროს ალგორითმის და გამოვლის ფუნქციები იქნებიან ყველაზე მარტივი, რასაც მიყვარს ავტომატის სტრუქტურის გამარტივებამდე. განხილულ მეთოდს საფუძვლად უდევს ორი მომენტი: პირველი მდგომარეობის კოდიფიკაციის ვარიანტების გადარჩევის რაოდენობის შემცირებაში; მეორე - ალგორითმის ფუნქციების სირთულის ისეთ შეფასების შემოღებაში, რომლის დროსაც არ იქნება საჭირო ამ ფუნქციების მოძებნა. გამოყენებულია მოცემული ავტომატის მდგომარეობის დაყოფა ჯგუფებად, ელემენტარული ავტომატის კოდიფიკაციის საფუძველზე, რაც შესაძლებლობას გვაძლევს შევაფასოთ თვითონეული ელემენტარული ავტომატის მომდევნო მდგომარეობის დამოკიდებულება სხვა ელემენტარული ავტომატებისაგან. ჩამოყალიბებულ წესების საფუძველზე დამუშავებულია ავტომატის კოდიფიკაციის ალგორითმი და პროგრამა C++ ენაზე, რომელიც გამოკვლეული იყო სხვა და სხვა გადასვლის ცხრილის მქონე ავტომატზე.