UNIVERSITY OF NIŠ
The scientific journal **FACTA UNIVERSITATIS**
Series: **Mechanical Engineering** Vol.1, N° 4, 1997 pp. 491 - 496
Editor of series: Nenad Radojković
Address: Univerzitetski trg 2, 18000 Niš, YU,
Tel: (018) 547-095, Fax: (018)-24-488

THEORY OF GRAPHS AND SOME POSSIBILITIES OF FINITE AUTOMATES OPTIMIZATION

UDK: 519.14

Miodrag M. Stojiljković[*] Milorad V. Rančić[**]

[*]Mechanical Engineering Faculty, 18000 Niš, Beogradska Street, 14
[**]Higher School of Mechanical Engineering, 23000 Zrenjanin, Dj. Stratimirović Street, 23

Abstract. *The theory of graphs is a very suitable mathematical device for describing finite automates. By defining and forming a finite automate graph some problems of the finite automate algebra can be solved in a far simpler way. This primarily refers to the possibility of optimizing by the decomposition method by which a graph is decomposed in its substructures. An algorithm has been formed by which a complex finite automate is decomposed into subautomates with the ultimate aim of turning them into elementary automates. All the considerations are at an abstract level.*

1. INTRODUCTION

A relation can be established between the theory of graphs and that of finite automates. The possibility of the graphs' matrix description in addition to its geometric interpretation enable the solving of some problems related to the finite automate algebra. In this paper attention is paid to the so-called problem of automates' decomposition, that is, to the problem of decomposing a complex automate into many simpler mutually-connected subautomates. The ultimate goal is to carry on the decomposition to elementary automates and to obtain in this way an optimal structure of the complex automate. The introduction of the theory of graphs into this kind of problem is very efficient since all the considerations are carried out at an abstract level.

2. DEFINITION OF FINITE AUTOMATE AND GRAPHS

The finite abstract automate is defined in the form of the following model:

$$A = (X, Y, Q, \psi, \varphi), \quad \text{where} \tag{1}$$

Received January 8, 1997

X- a set of input letters named the input alphabet,
Y- a set of output letters named the output alphabet,
Q- a set of internal states named the state alphabet,
φ: $Q \times X \to Q$ - transition function, and,
ψ: $Q \times X \to Y$ - output function.

The automate performs its operation in discrete time: $t = 0,1,2,3, \ldots , t_{max}$. At the initial moment t=0 the automate is in its initial position denoted by q_0. At the next moment of time t=1 the automate input is added the input letter X_1 and under its influence the automate passes into the state $q_1=\varphi(q_0,X_1)$, while at the automate output the output letter $Y_1=\psi(q_0,X_1)$ appears. At the moment t=2 the automate input is added the second output letter X_2. The automate then passes into the state $q_2=\varphi(q_1,X_2)$ while at its output the output letter $Y_2=\psi(q_1,X_2)$ appears. The automate's operation continues in this way so long as the input letters of the set X are brought to its input. They are defined for all the moments t=1,2,3...t_{max}. For $t = t_{max}+1$ they are not defined. This means that at the moment $t_{max}+1$ the automate ceases to operate. It can be seen from the description of the finite automate's operation that its input is consecutively been affected by X_1, X_2, X_3, ..., X_n that is, the letters from the set X which makes the input alphabet, and that at its output the letters $Y_1,Y_2,...,Y_m$ from the set Y appear, namely those that make up the output alphabet, while at the same time the automate passes through internal states q_1, q_2, ... ,q_k from the set Q which makes up the state alphabet. It can be concluded that the finite automate actually does the mapping of the set of the input letters X into the set of the output letters Y while simultaneously satisfying the output function ψ and the transition function φ. It is this conclusion which makes possible to establish a close connection between the finite automate theory and the theory of graphs.

As it is very well known, mathematically speaking, the graph is assumed to be an algebraic structure $G = (C, \alpha, \beta)$ where α and β are the mappings

$$\alpha: C \to C_0$$
$$\beta: C \to C_0,\ C_0 \subset C$$

satisfying the following conditions

$$\begin{aligned} \alpha(\alpha(x)) &= \beta(\alpha(x)), \quad \forall x \in C \setminus C_0 \\ \alpha(\beta(x)) &= \beta(\beta(x)) = \beta(x), \quad \forall x \in C \setminus C_0 \end{aligned} \tag{2}$$

The elements of the set $\{x \mid x \in C_0\}$ are termed nodes while the elements of the set $\{x \mid x \in C \setminus C_0\}$ are considered as the graph branches.

3. Graph of the Finite Automate

The finite automate can be represented in many ways such as:

- Analytically, that is, in the form of the model given in (1) under the condition that the output functions ψ and the transition functions φ are defined,
- In a table form, in the form of the tables of the states and those of inputs and transitions,

- In a matrix form, in the form of respective matrices, and,
- Graphically, in the form of the finite automate graph, that is, its graphoid.

The last of the above-mentioned ways will primarily be used in the following considerations. For each finite automate there is a an adequate graph formed in the following way:

- each node is allotted one internal state of the automate,
- each branch of the graph which leads from one node to another is loaded with one input letter under whose influence the automate passes from one state to another, as well as with one output letter which appears at the automates output during this transition.

Thus formed graph (multi-graph) is named the graph of the finite automate, that is, the graphoid.

The formation of the finite automate graph will be shown in the following example.

Example:

An automate A is given having $X = \{X_1, X_2, X_3\}$, $Y = \{Y_1, Y_2\}$, and $Q = \{q_1, q_2, q_3, q_4\}$, while the transition functions φ and the output functions ψ are represented by the tables:

X \ q	q_1	q_2	q_3	q_4
X_4	q_1	q_2	q_2	q_1
X_2	q_2	q_2	q_1	q_1
X_3	q_2	q_1	q_2	q_2

X \ q	q_1	q_2	q_3	q_4
X_1	Y_1	Y_2	Y_2	Y_1
X_2	Y_2	Y_2	Y_1	Y_1
X_3	Y_2	Y_1	Y_2	Y_2

The graph of the given finite automate is given in Fig. 1.

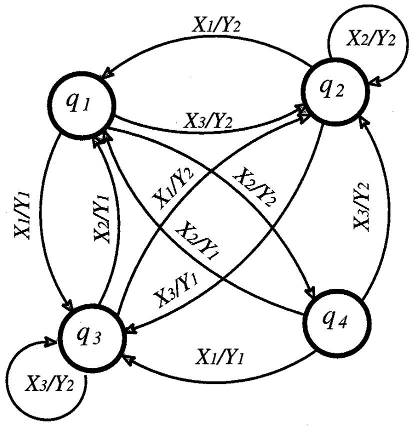

Fig. 1

4. SOME ELEMENTS OF THE ABSTRACT AUTOMATES' ALGEBRA

By using the abstract automate algebra one of the frequent problems can be solved, namely, the problem of forming complex automates out of many simpler ones. This is achieved by an adequate connection of automates. There are two basic cases: the series connection (Fig. 2) and the parallel connection (Fig. 3).

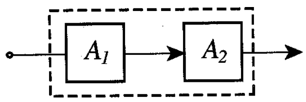

Fig. 2

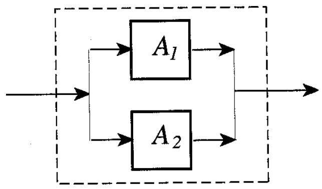

Fig. 3

Regarding the serially-connected automates $A_1=(X_1,Y_1,Q_1,\varphi_1,\psi_1)$ and $A_2=(X_2,Y_2,Q_2,\varphi_2,\psi_2)$ for the obtained automate $A=(X,Y,Q,\varphi,\psi)$ the following conditions should be satisfied $X=X_1$, $Y=Y_2$, $Y_1=X_2$ and $Q=Q_1\times Q_2$, while the functions φ and ψ are defined in the following way:

$$\begin{aligned}\varphi((q_i^{(1)},q_j^{(2)}),X_k^{(1)}) &= \varphi_1((q_i^{(1)},X_k^{(1)}),\varphi_2(q_j^{(2)},\psi_1(q_i^{(1)},X_k^{(1)})))\\ \psi((q_i^{(1)},q_j^{(2)}),X_k^{(1)}) &= \psi_2(q_j^{(2)},\psi_1(q_i^{(1)},X_k^{(1)}))\end{aligned} \tag{3}$$

The automates A_1 and A_2 are connected in such a way that the output signal of the automate A_1 is led to the input of the automate A_2.

Regarding the parallel-connected automates A_1 and A_2 for the obtained automate A the following should be valid: $X=X_1\times X_2$, $Y=Y_1\times Y_2$, $Q=Q_1\times Q_2$ while the functions φ and ψ are defined in the following way:

$$\begin{aligned}\varphi((q_i^{(1)},q_j^{(2)}),(X_k^{(1)},X_i^{(2)})) &= (\varphi_1(q_i^{(1)},X_k^{(1)}),\varphi_2(q_j^{(2)},X_1^{(2)}))\\ \psi((q_i^{(1)},q_j^{(2)}),(X_k^{(1)},X_i^{(2)})) &= (\psi_1(q_j^{(1)},X_k^{(1)}),\psi_2(q_j^{(2)},X_1^{(2)}))\end{aligned} \tag{4}$$

The inputs of the automates A_1 and A_2 are simultaneously influenced by the letters ($X_k^{(1)}$, and $X_l^{(2)}$) while a couple of letters ($Y_k^{(1)}$, $Y_l^{(2)}$) are regarded as the output letter.

By combining these two connections complex structures can be obtained, namely, considerably more complex automates.

Another frequent problem, opposite to this one, is that of the automates' decomposition. In fact, it comes to decomposing a complex finite automate into two or more simpler subautomates.

5. OPTIMAL DECOMPOSITION OF AUTOMATES

The procedure for decomposing an automate, that is, to decompose it into several simpler automates should be carried out until elementary automates are obtained (if it is possible). Thus, an "optimal decomposition" would give, for instance, a minimal number of logical elements entering into the composition of the automate's combinatory part.

The decomposition of a complex automate is done by applying the algebra of finite automate as its has already been mentioned. With respect to it, the parallel, the series and the mixed kinds of decomposition can be distinguished. As for the parallel decomposition, the automate is decomposed into a product or sum of two or more automates which are simpler than the initial one. The series decomposition is achieved by decomposing the automate by the superposition operation, while the mixed decomposition is carried out by combining both of these two mentioned operations.

Regarding the fact that one of the ways of representing the finite automate's operation is its graph, that is, its graphoid, the automate's decomposition can be also solved by applying the theory of graph. The problem is now reduced to the decomposition of the graph by breaking it into its substructures. The procedure for decomposing the finite automate's graph is carried out in several phases. It is necessary to do the following:

I. To determine whether the finite automate's graph is isomorphous,
II. To find the criteria by which the possibility for decomposition is determined, and,
III. If the graph G is decomposable, it is necessary to find subgraphs G_1, G_2, ... which it can be decomposed into.

The theory of graphs knows of many criteria for recognizing isomorphisms. According to the Ref. [1, 2], two graphs, G and H, are isomorphous if the assumption $t \in T$ exists, where T is a symmetrical group of arrangements of the set of elements, namely, the one which brings into an unambiguous accordance all the elements of $x_i \in X$ of the graph G and the elements $y_j \in Y$ of the graph H so that the input semidegrees and the output semidegress are equal for all the nodes of the graphs G and H.

An algorithm is suggested for a parallel decomposition of the automate's graph shown in the Ref. [3, 4, 5].

A finite automate A is given in the form of its graphoid and it should be decomposed into a product of two subautomates A_1 and A_2.

Theorem: The automate A having n states, where $n=k \times l$ can be represented by a product of the automates of A_1 and A_2 if and only if there is an arrangement of $t \in T$ of the state alphabet which transforms the neighborhood matrix of the automate A into the form of a *regular elementary connection matrix* (RECM).

Decomposition Algorithm

1. Count the number of n states of the automate A. If $n=k \times l$, proceed to the point 2. If n is a prime number, then the point 7 is to be proceeded to.

2. The matrix of the connection R_A of the automate A should be decomposed into k^2 elements of which every one is of the order *1*. If R_A is a regular elementary connection matrix, then proceed to 6. Otherwise proceed to 3.

3. According to the connection matrix R_A write down the neighborhood matrix R of the automate A. Proceed to 4.

4. Apply the method of the graph decomposition into a product of two graphs while looking for an arrangement which translates the neighborhood matrix R into a regular elementary matrix R'. If $t \in T$ is the desired arrangement, proceed to 5. If not, to 7.

5. Apply the obtained arrangement to the connection matrix R_A of the automate A. If a

regular elementary connection matrix R_A' is obtained, proceed to 6. Otherwise, to 7.

6. According to the matrix R_A', we form the connection matrix R_{A1} and R_{A2} of the automates A_1 and A_2 as defined in the theorem.

7. The automate A is not decomposable.

The proof of the previous theorem, of the decomposition algorithm and of the definitions of all the concepts related to the algorithm are presented in the Ref. [3].

By applying the proposed algorithm the connection matrix R_A, the neighborhood matrix R, and the regular elementary connection matrix R_A' are formed on the basis of the connection matrixes R_{A1} and R_{A2} which define respective graphoids of the subautomates A_1 and A_2. Thus the procedure of decomposition is completed.

$$A \to R_A \to R \to R_A' \begin{array}{l} \nearrow R_{A1} \to A_1 \\ \searrow R_{A2} \to A_2 \end{array}$$

By introducing the theory of graphs into the finite automate theory a suitable mathematical device is obtained for decomposing a complex automate into its subautomates. This should be done by an optimal decomposition until elementary automates are obtained. Though the whole procedure is carried out at an abstract level it can be important in the phase of designing a complex automate structure when “real” elementary automates are passed on to and when an optimal functional scheme is being formed.

REFERENCES

1. Belov V.V i dr., Teorija grafov, Moskva, Višaja škola, 1976.
2. Melihov A.N. i dr., Primenenije grafov dla projektirovaije diskretnih ustrojstv, "Nauka", Moskva, 1974.
3. Rančić M., Dekompoziciona metoda strukturne sinteze automata, 21.stručni skup o hidropneumatičkoj automatici i novim tehnologijama HIPNEF'86, SMEITS-Beograd, Beograd, 1986.
4. M.Stojiljković, M.Rančić, Dekompozicija automata i širenje polugrupa, Naučno informativni časopis DIT-Zrenjanin, god II, br. 4/96, 1996.
5. M.Stojiljković, M.Rančić, Projektovanje strukture konačnih automata primenom paralelne dekompozicije, Časopis instituta IMK "14 oktobar"-Kruševac,"IMK-14 istraživanje i razvoj", god.III, br. 1/97.

TEORIJA GRAFOVA I NEKE MOGUĆNOSTI OPTIMIZACIJE KONAČNIH AUTOMATA

Miodrag M. Stojiljković, Milorad V. Rančić

Teorija grafova je veoma pogodan matematički aparat za opisivanje konačnih automata. Kada se definiše i formira graf konačnog automata, neki problemi algebre konačnih automata mogu da se reše daleko jednostavnije. To se pre svega odnosi na mogućnosti optimizacije metodom dekompozicije, gde se graf razbija na svoje podstrukture. Formiran je algoritam kojim se složeni konačni automat razlaže na podautomate, sa krajnjim ciljem da to budu elementarni automati. Sva razmatranja su na apstraktnom nivou.