ON DUAL AUTOMATA

B. G. Mirkin

Kibernetika, Vol. 2, No. 1, pp. 7–10, 1966

UDC 519.95

Every finite automaton A with m states can be associated with a dual nondeterministic automaton A*, the automaton A* being the inversion P* of the event P represented by automaton A [1]. It turns out that the minimal automaton representing the event P* is obtained as a result of determinization [1] of the dual A*. The number of states of this automaton does not exceed 2^m, and for every natural $m \geq 3$ there exists an automaton A_m for which this estimate is achieved.

This result is also proved in this article, but the discussion is conducted in terms of the operations of left and right partition of events into words, which permits a more thorough clarification of the connection between the automata and the events they represent.

On the basis of the proved theorem, we indicate a method for synthesizing the minimal automaton representing a given (by its regular expression) event.

1. Let $X = \{x_1, \ldots, x_n\}$ be a finite alphabet. We denote the free semigroup over alphabet X (supplemented by the empty word e) by F(X), and the set of events over X by E(X).

Consider the following operations in the set E(X). We call the set of all words q such that $pq \in P$, that is,

$$q \in P_p \longleftrightarrow pq \in P, \tag{1.1}$$

the left quotient P_p of the partition of the event $P \in E(X)$ by the word $P \in F(X)$.

It is not difficult to note that P_p is the greatest of the events R satisfying the inclusion $pR \subset P$ (which also explains the name of the operation).

In an analogous manner, the right quotient ${}_pP$ of the partition of $P \in E(X)$ by $p \in F(X)$ is defined as the set of words q such that $qp \in P$, that is,

$$q \in {}_pP \longleftrightarrow qp \in P. \tag{1.2}$$

We denote the set of all different left (right) quotients of the event P by $P_u({}_uP)$. We note that $P = P_e \in P_u$, $P = {}_eP \in {}_uP$. We denote the number of elements in the set $P_u({}_uP)$ in the usual manner: $|P_u|(|{}_uP|)$.

It is known that P can be represented by a finite automaton if and only if $|P_u|$ is finite.

It is not difficult to establish a relation between left and right partitions of events with the aid of the operation of inversion examined in reference [1]. We recall that the word $p^* = x_{i_k} \ldots x_{i_2} x_{i_1}$ is called the inversion of the word $p = x_{i_1} \ldots x_{i_2} x_{i_k}$, while the inversion of the event P is the next event P*:

$$p \in P^* \longleftrightarrow p^* \in P. \tag{1.3}$$

The following properties are obvious:

$$P^{**} = P, \tag{1.4}$$

$$(P_1 \cup P_2)^* = P_1^* \cup P_2^*, \tag{1.5}$$

$$(P_1 P_2)^* = P_2^* P_1^*. \tag{1.6}$$

It is now easy to prove that

$$(P_p)^* = {}_{p^*}P^*, \tag{1.7}$$

$$({}_pP)^* = P^*_{p^*}. \tag{1.8}$$

2. We associate with each event $P \in E(X)$ two binary equivalence relations on the set F(X):

$$(p, q) \in \varepsilon_p \longleftrightarrow P_p = P_q, \tag{2.1}$$

$$(p, q) \in {}_p\varepsilon \longleftrightarrow {}_pP = {}_qP. \tag{2.2}$$

For any binary relation $\varphi \subset F(X) \times F(X)$, we denote by φ^* the following relation:

$$(p, q) \in \varphi^* \longleftrightarrow (p^*, q^*) \in \varphi. \tag{2.3}$$

It is easy to see that

$$\varepsilon_p = {}_{p^*}\varepsilon^*. \tag{2.4}$$

Indeed, the fact that $(p, q) \in \varepsilon_p$ means that $P_p = P_q$, that is, $(P_p)^* = (P_q)^*$. And, by (1.7), this is equivalent to ${}_{p^*}P^* = {}_{q^*}P^*$, which it was required to prove.

In a like manner

$${}_p\varepsilon = \varepsilon^*_{p^*}. \tag{2.5}$$

It is also easy to see that for any $p \in F(X)$

$$\varepsilon_p({}_pP) \subset {}_pP, \tag{2.6}$$

$${}_p\varepsilon(P_p) \subset P_p. \tag{2.7}$$

Indeed, let $p_1 \in \varepsilon_p({}_pP)$. This means that there exists a $p_2 \in {}_pP$ such that $P_{p_1} = P_{p_2}$. The fact that $p_2 \in {}_pP$ is equivalent to the fact that $p_2p \in P$, that is, $p \in P_{p_2} = P_{p_1}$. Then $p_1p \in P$, that is, $p_1 \in {}_pP$, which it was required to prove. Relation (2.7) is verified in a like manner.

Properties (2.6) and (2.7) mean that the right (left) quotients of the event P are unions of equivalence classes of the left (right) relation $\varepsilon_p({}_p\varepsilon)$. And since the number of equivalence classes of the relation $\varepsilon_p({}_p\varepsilon)$ coincides with $|P_u|(|{}_uP|)$, the following estimates are valid:

$$|{}_uP| \leqslant 2^{|P_u|}, \tag{2.8}$$

$$|P_u| \leqslant 2^{|{}_uP|}. \tag{2.9}$$

Later we shall show that these estimates are exact (that is, the equalities are achieved).

3. We associate the following two Moore automata with each event $P \in E(X)$:

$$A_p = (P_u, X, \{\varnothing, e\}, \delta_p, \chi), \tag{3.1}$$

$$ {}_{p}A = ({}_{u}P, X, \{\varnothing, e\}, {}_{p}\delta, \chi), \qquad (3.2)$$

where $\delta_p(P_p, x) = P_{px}$, ${}_{p}\delta({}_{p}P, x) = {}_{xp}P \ (x \in X)$,

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

It is known [3, 4] that any event $R \in E(X)$ can be uniquely represented in the form

$$R = x_1 R_{x_1} \cup \ldots \cup x_n R_{x_n} \cup \chi(R), \qquad (3.3)$$

$$R = {}_{x_1}R \cdot x_1 \cup \ldots \cup {}_{x_n}R \cdot x_n \cup \chi(R). \qquad (3.4)$$

Therefore, specifying the automaton A_P is equivalent to specifying the system of equations

$$P_p = x_1 P_{px_1} \cup \ldots \cup x_n P_{px_n} \cup \chi(P_p)\,(P_p \in P_u), \qquad (3.5)$$

and specifying the automaton ${}_{P}A$ is equivalent to specifying the system of equations

$${}_{p}P = {}_{x_1 p}P x_1 \cup \ldots \cup {}_{x_n p}P x_n \cup \chi({}_{p}P)\,({}_{p}P \in {}_{u}P). \qquad (3.6)$$

It is easy to see that for any $q \in F(X)$

$$\delta_P(P_p, q) = P_{pq}, \qquad {}_{P}\delta({}_{p}P, q) = {}_{qp}P. \qquad (3.7)$$

It follows from (3.7), (1.7), and (1.8) that

$$\delta_P(P_p, q) = ({}_{P^*}\delta({}_{p^*}P^*, q))^*. \qquad (3.8)$$

It is known [2—4] that the automaton A_P represents each $P_p \in P_u$ by the output signal e of the initial state P_p, that is,

$$q \in P_p \longleftrightarrow e \in \delta_P(P_p, q). \qquad (3.9)$$

It follows from (3.8) and (3.9) that

$$q \in P^*_{p^*} \longleftrightarrow e \in {}_{P}\delta({}_{p}P, q), \qquad (3.10)$$

that is, the automaton ${}_{P}A$ represents the event ${P_{p^*}}^*$ of the initial state ${}_{p}P$.

This means that the automaton ${}_{P}A$ is indistinguishable from the automaton A_{P^*}, and by (2.5) they have the same number of states. Moreover, since the automaton A_P has the least number of states of all the Moore automata representing P [2—4], ${}_{P}A = A_{P^*}$ (correct to an isomorphism).

4. In an arbitrary Moore automaton $A = (S, X, Y, \delta, \mu)$ with $S = s_1, \ldots, s_m$ we fix the initial state s_1 and consider the equivalence relation $\varepsilon_A \subset F(X) \times F(X)$, defined by the expression

$$(p, q) \in \varepsilon_A \longleftrightarrow \delta(s_1, p) = \delta(s_1, q). \qquad (4.1)$$

Let the automaton A have the transition matrix $(a_{ij})_{1 \le i, j \le m}$ (a_{ij} is the union of those input signals that carry the automaton A from the state s_i to state s_j). It is known [5] that the equivalence classes $E_1, \ldots, E_m$ (E_i is the event represented by the state s_i ($i = 1, \ldots, m$)) of the relation ε_A satisfy the system of equations

$$E_i = E_1 a_{1i} \cup \ldots \cup E_m a_{mi} \cup \chi(E_i) \quad (i = 1, \ldots, m), \qquad (4.2)$$

where $\chi(E_i) = \begin{cases} e, & \text{if } i = 1, \\ \phi, & \text{if } i > 1. \end{cases}$

Hence it immediately follows that the equivalence classes $F_1 = E_1^*, \ldots, F_m = E_m^*$ of the relation ε_A^* satisfy the equations

$$F_i = a_{1i} F_1 \cup \ldots \cup a_{mi} F_m \cup \chi(F_i) \quad (i = 1, \ldots, m), \qquad (4.3)$$

where $\chi(F_i) = \begin{cases} e, & \text{if } i = 1, \\ \phi, & \text{if } i > 1. \end{cases}$

Let A represent the event P by the set of output signals $Z \subset Y$. It is not difficult to prove that

$$\varepsilon_A \subset \varepsilon_P. \qquad (4.4)$$

Indeed, let $(p, q) \in \varepsilon_A$. This means that $\delta(s_1, p) = \delta(s_1, q)$. It follows that for any $r \in F(X)$ $pr \in P \longleftrightarrow qr \in P$, that is, $r \in P_p \longleftrightarrow r \in P_q$, but this means that $P_p = P_q$, which is what it was required to prove.

It follows from (4.4) and (2.4) that:

$$\varepsilon_A^* \subset {}_{P}\varepsilon. \qquad (4.5)$$

Formulas (4.4) and (4.5) together with (2.6) and (2.7) mean that right quotients of the event P are unions of classes $E_1, \ldots, E_m$ of the relation ε_A and left quotients of the event P^* are unions of the equivalence classes $F_1, \ldots, F_m$ of the relation ε_A^*.

We shall now show how, with the aid of (4.2) and (4.3), we cannot only express the right quotients of the event P in terms of the events $E_1, \ldots, E_m$, but also show the connection between them.

Let $\{s_{i_1}, \ldots, s_{i_k}\}$ be a set of states marked by output signals from Z (that is, a set of finite states). Then [5]

$$P = E_{i_1} \cup \ldots \cup E_{i_k}, \qquad (4.6)$$

$$P^* = F_{i_1} \cup \ldots \cup F_{i_k}. \qquad (4.7)$$

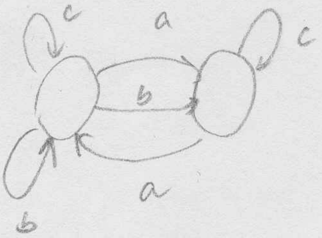
Substituting expressions (4.2) in (4.6) in place of $E_{i_1}, \ldots, E_{i_k}$ and making use of the distributivity of multiplication with respect to unions, we obtain

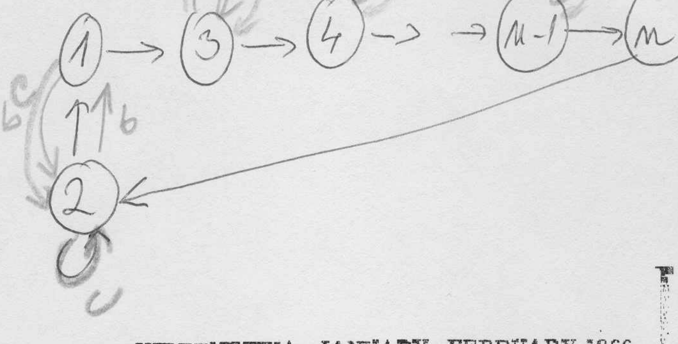
$$P = P_0 = P_1 x_1 \cup \ldots \cup P_n x_n \cup \chi(P),$$

where $P_1, \ldots, P_n$ are unions of events of the set $\{E_1, \ldots, E_m\}$. For those P_i which are not unions of events $E_{i_1}, \ldots, E_{i_k}$ (that is, not equal to $P = P_0$), we again write out the equations using (4.2). If unions of events $E_1, \ldots, E_m$ not previously encountered appear on their right sides, then we continue writing out equations for these new quotients. We continue until new unions of events $E_1, \ldots, E_m$ cease to appear. This takes not more than 2^m steps, since the number of different unions of m elements is equal to 2^m. As a result, we obtain equations of the form (3.6). If the automaton A is connected, then all classes of the relation ε_A are different and nonempty. This means that all events P_i obtained are different and, consequently, the equations obtained define the automaton ${}_{P}A$.

If the connected automaton A represents the event $Q = P^*$, then, in the same way, with the aid of (4.3) and (4.7), we obtain equations of the form (3.5) defining the automaton A_P.

Thus, we have shown how to construct the minimal automaton ${}_{P}A$ (A_P) from a given automaton A representing the event P(P*).





We note that the principal stages of our construction can be expressed in the terminology of Rabin and Scott [1] as follows: 1) construction of the nondeterministic automaton A*, the dual of the automaton A (equations (4.2)), and 2) determinization of the automaton A* (construction of equations of the form (3.6)).

Therefore, we can formulate our result in the same terminology as follows: the connected part of the automaton obtained as a result of determinization of the nondeterministic dual of a given automaton is a minimal automaton.

Example 1. Consider the automaton A_m with states $\{1, 2, \ldots, m\}$ $(m \geq 3)$, output alphabet $X = \{a, b, c\}$, initial state 1, final state 1, and the transition diagram shown in Fig. 1. It is not difficult to see that the dual A^*_m is equal (correct to an isomorphism) to a nondeterministic automaton (source) U_m for which, as shown by Lupanov [6], the equivalent deterministic automaton (special source) contains exactly 2^m states, which also proves the estimates (2.8), (2.9).

5. The algorithm described above can be applied to the synthesis of the minimal automaton representing the event P from its regular expression.

For this, it is first necessary to construct the regular expression of event P*, then with the aid of a known algorithm synthesize automaton A which represents the event P*. This automaton A can also be constructed directly from the regular expression of P by applying the synthesis algorithm to P, not, as usual, from left to right, but from right to left. The transition from automaton A to Ap is realized with the aid of (4.3) and (4.7).

It may happen that $|P_U|$ considerably exceeds the number of states of automaton A (Example 1). In this case, our method for constructing the minimal automaton is preferable to the methods now being used, where one first obtains an automaton with a number of states exceeding $|P_U|$, and only then proceeds to its minimization. However, the opposite may happen: $|P_U|$ may be considerably smaller than the number of states A. In this case, our path is clearly longer,

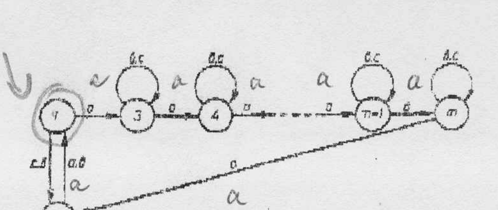


Fig. 1

Example 2. Let $X = \{0, 1\}$. We consider the regular expression

$$P = \{0\{0\{1\}1\}(0 \cup 1)\}.$$

To P we apply the algorithm for constructing a basis [3] from right to left. In virtue of the identity

$$\{R\} = \{R\}R \cup e$$

we have

$$P = P_1 = P0\{0\{1\}1\}(0 \cup 1) \cup e = P_2(0 \cup 1) \cup e,$$
$$P_2 = P_10\{0\{1\}1\} = P_20\{1\}1 \cup P_10 = P_31 \cup P_10,$$
$$P_3 = P_20\{1\} = P_31 \cup P_20.$$

The system of equations obtained gives the automaton pA. Its transition matrix is

$$\begin{pmatrix} \varnothing & 0 \cup 1 & \varnothing \\ 0 & \varnothing & 1 \\ \varnothing & 0 & 1 \end{pmatrix}.$$

Then the system of equations for equivalence classes of the relation $\varepsilon pA^* (= {}_p\varepsilon)$ is of the form:

$$F_1 = 0F_2 \cup e,$$
$$F_2 = 0(F_1 \cup F_3) \cup 1F_1,$$
$$F_3 = 1(F_2 \cup F_3).$$

From this we obtain a system of equations of the (3.5):

$$P = P_1 = F_1 = 0F_2 \cup 1\varnothing \cup e = 0P_2 \cup 1P_3 \cup e,$$
$$P_2 = F_2 = 0(F_1 \cup F_3) \cup 1F_1 = 0P_4 \cup 1P_1,$$
$$P_3 = \varnothing = 0P_3 \cup 1P_3,$$

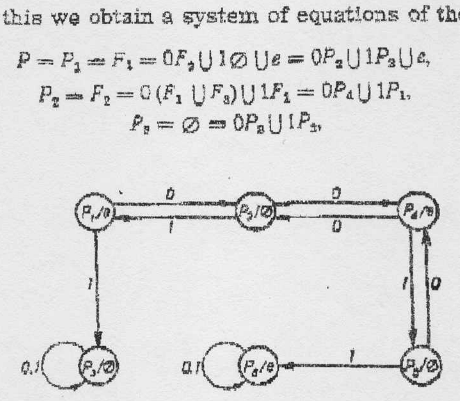


Fig. 2

$$P_4 = F_1 \cup F_3 = 0F_2 \cup 1(F_2 \cup F_3) \cup e = 0P_2 \cup 1P_5 \cup e,$$
$$P_5 = F_2 \cup F_3 = 0(F_1 \cup F_3) \cup 1(F_1 \cup F_2 \cup F_3) =$$
$$= 0P_4 \cup 1P_6,$$
$$P_6 = F_1 \cup F_2 \cup F_3 = 0P_6 \cup 1P_6 \cup e.$$

Thus, the desired automaton Ap has the diagram shown in Fig. 2.

In conclusion, the author thanks M. A. Spivak for his interest in this work.

REFERENCES

1. M. O. Rabin and D. Scott, "Finite automata and their decision problems," IBM J. Research Develop., vol. 3, pp. 114–125, 1959.

2. M. A. Spivak, "A new algorithm for abstract synthesis of automata," Proceedings of Scientific Seminars on Cybernetics, Theory of Automata [in Russian], no. 3, Kiev, 1963.

3. M. A. Spivak, "An algorithm for abstract synthesis of automata for an extended language of regular expressions," Izvestiya AN SSSR, Tekhnicheskaya kibernetika, no. 1, 1965.

4. J. A. Brzozowski, "Derivatives of regular expressions," J. of the Association for Computing Ma-

chinery, 11, 4, 1964.

5. V. G. Bodnarchuk, "Systems of equations in the algebra of events," Zhurnal vychislitel'noy matematiki i matematicheskoy fiziki, 3, 6, 1963.

6. O. B. Lupanov, "A comparison of two types of finite sources," Problemy kibernetiki, no. 9, 1963.

28 May 1965

Saratov State University