Jan Mareš

A note on Glushkov's algorithm of the synthesis of finite automata

Kybernetika, Vol. 8 (1972), No. 4, (289)--296

Persistent URL: http://dml.cz/dmlcz/124393

KYBERNETIKA — VOLUME 8 (1972), NUMBER 4

A Note on Glushkov's Algorithm of the Synthesis of Finite Automata

JAN MAREŠ

In this paper the problem of the optimalization of one step in the synthesis of finite automata is solved.

Let L be a *finite* language. Under the *synthesis* we understand here finding the transition function of a (finite) Moore automaton $\mathfrak{A}$ that represents the language L by some subset of the set of its inner states. An algorithm of this synthesis is given in [1]. For concreteness we have in mind the algorithm mentioned on pages 101 to 102, rules 1 to 4. In what follows, a knowledge of the algorithm is supposed.

Let Σ be a *finite alphabet*, i.e. a finite nonempty set of arbitrary symbols. Denote by Σ^* the set of all strings over Σ. Consider the set $\mathfrak{F}$ of all such regular expressions over Σ which involve only operations *sum* (denoted by the symbol $+$) and *catenation* (denoted by juxtaposition) — except symbols belonging to Σ and parentheses, of course. Every such a regular expression will be referred to as "expression".

Each expression represents some finite language in the following sense: We say that an expression R *represents* a language L iff $|R| = L$, where the operation $|\ldots|$ is defined as follows (R_1, R_2 are expressions):

$$|\alpha| = \{\alpha\} \quad \text{for each} \quad \alpha \in \Sigma^* ,$$

$$|R_1 + R_2| = |R_1| \cup |R_2| ,$$

$$|R_1 R_2| = \{\alpha\beta \mid \alpha \in |R_1| \wedge \beta \in |R_2|\} .$$

Every finite language is, of course, represented by many various expressions yielding, after applying the algorithm to them, automata with various numbers of states. Our effort will consist in searching for such an expression which yields an automaton $\mathfrak{A}$ with as few states as possible. However, this automaton $\mathfrak{A}$ is not "absolutely" minimal (in accordance with [1], p. 135). On the other hand, there are algorithms that, for finite languages, construct automata which are minimal; see e.g. [3]. Thus the automaton $\mathfrak{A}$ is "minimal" only in the set of all automata which can be obtained by the mentioned Glushkov's algorithm.

We confine ourselves (obviously without loss of generality) to languages not containing the empty string.

Let R be an expression. Denote by $\mathfrak{A}(R)$ the automaton that is obtained by applying the algorithm to the expression R. Furthermore denote by $\sigma(R)$ the number of states of $\mathfrak{A}(R)$. Finally, if L is a language, put

$$\varrho(L) = \{R \in \mathfrak{F} \mid |R| = L\}.$$

The Formulation of the task:

To find, for an arbitrary *finite* language L, its *minimal form*, i.e. such an expression $R_M \in \varrho(L)$ that

$$\sigma(R_M) = \operatorname*{Min}_{R \in \varrho(L)} \sigma(R).$$

Assume $R, S \in \mathfrak{F}$. In what follows, by $R = S$ we denote the fact that $|R| = |S|$. On the other hand, under the identity $R \equiv S$ we understand "graphical identity" of R, S, i.e. R and S are equal as strings over the "extended" alphabet Σ_0 containing, except symbols belonging to Σ, symbols $+$, left parenthesis and right parenthesis.

Let $R \equiv y_1, \ldots, y_m$ be an expression (of course, here the symbols $y_1, \ldots, y_m$ belong to Σ_0). Each expression $y_k y_{k+1}, \ldots, y_t$, where $1 \leqq k \leqq t \leqq m$, $y_{k-1} \notin \Sigma$ or $y_k \notin \Sigma$, and at the same time, $y_t \notin \Sigma$ or $y_{t+1} \notin \Sigma$, we call a *subexpression* of R.

Assume $R, S, T \in \mathfrak{F}$. The following identities are true.

$$\mathrm{I}_1 \qquad R + R = R\,,$$

$$\mathrm{I}_2 \qquad (R + S) + T = R + (S + T)\,,$$

$$\mathrm{I}_3 \qquad (RS)T = R(ST)\,,$$

$$\mathrm{I}_4 \qquad R + S = S + R\,,$$

$$\mathrm{I}_5 \qquad R(S + T) = RS + RT\,,$$

$$\mathrm{I}_6 \qquad (R + S)\,T = RT + ST\,.$$

Moreover, these identities are *complete* in the following sense. If $R, S \in \mathfrak{F}$, $|R| = |S|$, then there is a sequence $R_1, \ldots, R_m$ such that $R_1 \equiv R$, $R_m \equiv S$ and for each i, $1 \leqq i \leqq m - 1$, R_{i+1} arises from R_i by applying identity I_j for suitable j, $1 \leqq j \leqq 6$.

(We say that $V \in \mathfrak{F}$ arises from $U \in \mathfrak{F}$ by applying identity I_i $(1 \leqq i \leqq 6)$ iff the following condition is true:

If we replace the symbols R, S, T of one side of I_i by suitable expressions in such a way that some subexpression U_0 of U is obtained, then V arises from U by replacing U_0 by V_0, where V_0 arose from the other side of I_i by the same replacing of R, S, T as U_0 did.)

In what follows, we shall consider (without any loss of generality) only such expressions in which parentheses will be written if and only if this is necessary for the correct interpretation of the expression (as regards representation). Thus identities I_2 and I_3 can be omitted.

Let R be an expression. If it is possible to apply to R I_5 or I_6 *from left to right*, denote the expression arisen in such a way by ${}^{L}R$ or by R^{R} respectively. On the other hand, if it is possible to apply to R several times I_4 and then (once) I_5 or I_6 *from right to left*, denote the expression obtained in such a way by $'R$ or by R' respectively; if this is not possible (i.e. after no multiple application of I_4 is it possible to apply I_5), put $'R \equiv R$ or $R' \equiv R$. (Of course, in general there are many expressions that can be denoted by $'R$ or R'.)

It is easy to show that

(1) *if* $R, S \in \mathfrak{F}$ *and* S *arises from* R *by applying* I_4, *then* $\sigma(S) = \sigma(R)$.

If $T \in \mathfrak{F}$, denote by $\mathscr{S}(T)$ the set of all states of the automaton $\mathfrak{A}(T)$.

Now let R be an expression for which some $'R \not\equiv R$ and examine the relation between the numbers $\sigma(R)$ and $\sigma('R)$. Due to (1) we can confine ourselves to that case, when $'R$ arises from R by applying I_5 only (from right to left), i.e. by replacing $SP_1 + SP_2$ by $S(P_1 + P_2)$, where $S, P_1, P_2 \in \mathfrak{F}$. Write down some places and indices in R and $'R$. Corresponding parts have the forms

(2) $$\underset{I}{|} S \underset{J_1}{|} P_1 | + \underset{I}{|} S \underset{J_1}{|} P_2 |,$$

(3) $$\underset{I}{|} S \underset{J_2}{|}(| P_1 | + \underset{J_2}{|} P_2 |)|$$

where I, J_1, J_2 are sets of indices.

Consider, how (2) differs from (3), i.e. R from $'R$. Let M or N be the set of all principal indices in the expression denoted by the first of the second occurrence of S from left in (2), respectively. Assume M is (at the same time) the set of all principal indices in the expression denoted by S in (3).

If in (2) $m \in M$ x_m-follows $m' \in I \cup M$ and $n \in N$ x_m-follows $n' \in I \cup N$, clearly in (3) only m x_m-follows m' (index n does not occur in (3)). In such a way it is possible to assign to each $m \in M$ an $n \in N$ corresponding to it and vice versa (M and N have the same number of elements). Further $J_2 = J_1$. There are no other differences.

The states of the automata $\mathfrak{A}(R)$ and $\mathfrak{A}('R)$ consist of sets of (principal) indices of R and $'R$. The set $\mathscr{S}('R)$ is obtained from $\mathscr{S}(R)$ in such a way that each index $n \in N$ is replaced in *all* states of $\mathscr{S}(R)$ by the corresponding index $m \in M$. From two different states arise two different ones again. Really, an arbitrary state $s \in \mathscr{S}(R)$ contains index n if and only if s contains the index m which corresponds to n. Thus by replacing n by m we get from two different sets of indices two different ones, too.

 Hence

$$\sigma('R) = \sigma(R)\,. \tag{4}$$

We call here every mapping F from Σ^* to $N \cup \{0\}$ (where N is the set of all positive integers) a (finite) *family* iff $F(\alpha) > 0$ for a finite number of strings $\alpha \in \Sigma^*$ only.

If F, G are families, define:

$$F = G \quad \text{iff} \quad \forall \alpha \in \Sigma^* \, (F(\alpha) = G(\alpha))\,,$$

$$(F \oplus G)(\alpha) = F(\alpha) + G(\alpha) \quad \text{for each} \quad \alpha \in \Sigma^*\,,$$

$$(F \ominus G)(\alpha) = F(\alpha) - G(\alpha) \quad \text{for each} \quad \alpha \in \Sigma^*\,;$$

$F \ominus G$ is defined only for such families F, G, for which $F(\alpha) \geqq G(\alpha)$ for each $\alpha \in \Sigma^*$.

Clearly, every family F can be characterized by a finite *list* $\mathscr{L}_F$ of exactly those strings $\alpha \in \Sigma^*$ for which $F(\alpha) > 0$; in this list $\mathscr{L}_F$ each $\alpha \in \Sigma^*$ occurs $F(\alpha)$-times. Two lists which differ only in the order of strings are supposed to be identical.

E.g. if $F(\alpha) = 2$, $F(\beta) = 1$, $F(\gamma) = 3$ and $F(\xi) = 0$ for $\xi \in \Sigma^* - \{\alpha, \beta, \gamma\}$, we write $\mathscr{L}_F = [\alpha, \alpha, \beta, \gamma, \gamma, \gamma]$.

In what follows, we speak about $\mathscr{L}_F$'s themselves as about "families". Each finite set $A \subset \Sigma^*$ will be understood as a family that contains each string $\alpha \in \Sigma^*$ at most once.

Each expression which is a sum of several nonempty strings (belonging to Σ^*) is said to be a *polynomial*.

Obviously, every expression R is *unbracketetable to a polynomial*, i.e. there is a sequence $R_1, \ldots, R_m$ such that

$$R_1 \equiv R\,, \quad R_{i+1} \equiv \bar{R}_i\,,$$

where either $\bar{R}_i \equiv {}^{\mathrm{L}}R_i$ or $\bar{R}_i \equiv R_i^{\mathrm{R}}$ $(i = 1, \ldots, m - 1)$ and R_m is a polynomial.

If P is a polynomial, $P \equiv \alpha_1 + \ldots + \alpha_n$, $\alpha_i \in \Sigma^*$ $(i = 1, \ldots, n)$, put $\|P\| = [\alpha_1, \alpha_2, \ldots, \alpha_n]$. ($\|P\|$ is a family.)

Let R be an expression and let F be a family. We write $\|R\| = F$ iff R is unbracketetable to a polynomial P such that $\|P\| = F$. (Clearly, it is not possible that R is unbracketetable to polynomials P_1 and P_2 such that $\|P_1\| \neq \|P_2\|$.)

Each expression R for which $\|R\| = |R|$ is said to be *simple*.

Lemma. *Let R be an arbitrary expression. There is a simple expression S such that $|S| = |R|$ and $\sigma(S) = \sigma(R)$.*

Proof. (In the proof a knowledge of the notion TotF is necessary; see [2] for it.)

Assume R is not simple. By [2] (Theorem 1) there is a sequence $R_1, \ldots, R_m$ such that $R_1 \equiv R$, $R_{i+1} \equiv {}^{\mathrm{L}}R_i$ and R_m has the form TotF $(i = 1, \ldots, m - 1)$. At the same time by (4) it is $\sigma(R_m) = \sigma(R)$ and of course $\|R_m\| = \|R\|$.

Because R is not simple, there is $\alpha \in \Sigma^*$, $\alpha \equiv x_1, \ldots, x_p$ such that $\|R\| = \|R_m\| = [\alpha, \alpha, \gamma, \ldots]$.

Clearly, it is sufficient to prove that there is $S_m \in \mathfrak{F}$ such that

$$\sigma(S_m) = \sigma(R_m) \quad \text{and} \quad \|S_m\| = \|R_m\| \ominus [\alpha] \, .$$

Assume $R_m \equiv y_1, \ldots, y_d$ $(y_i \in \Sigma_0)$; each subexpression $T \equiv y_{d_1}, \ldots, y_{d_2}$ of R_m such that $y_{d_1 - 1}$ is the left parenthesis, $y_{d_2 + 1}$ is the right parenthesis and for no i, $d_1 \leqq i \leqq d_2$, y_i is a parenthesis, is called a minimal subexpression of R_m. (Clearly, T is a polynomial.)

Suppose that $T_1, \ldots, T_k$ are all minimal subexpressions of R_m. If $\delta \in \Sigma^*$ (possibly $\delta \equiv \Lambda$; Λ denotes the empty string), and T is a polynomial, $T \equiv \delta_1 + \ldots + \delta_q$, put $\|T\| \, \|\delta\| = [\delta_1 \delta, \ldots, \delta_q \delta]$. It is not difficult to verify that

$$\|R_m\| = \|T_1\| \, \|\eta_1\| \oplus \ldots \oplus \|T_k\| \, \|\eta_k\| \oplus [\beta_1, \ldots, \beta_h] \, ,$$

where $\eta_1, \ldots, \eta_k, \beta_1, \ldots, \beta_h \in \Sigma^*$ are suitable strings, $k \geqq 0$, $h \geqq 0$. Put $T_0 \equiv \beta_1 + \ldots + \beta_h$, $\eta_0 \equiv \Lambda$.

Thus there are r, s, $r \neq s$, $0 \leqq r \leqq k$, $0 \leqq s \leqq k$ such that

$$\|T_r\| \, \|\eta_r\| \oplus \|T_s\| \, \|\eta_s\| = [\alpha, \alpha, \gamma, \ldots] \, .$$

Now there are two possibilities: either

$$T_r \equiv \ldots + x_1 \ldots x_{t_1} + \ldots \, , \quad \eta_r \equiv x_{t_1 + 1} \ldots x_p$$

and

$$T_s \equiv \ldots + x_1 \ldots x_{t_2} + \ldots \, , \quad \eta_s \equiv x_{t_2 + 1} \ldots x_p \, ,$$

or

$$T_r \equiv \ldots + x_1 \ldots x_{t_1} + \ldots + x_1 \ldots x_{t_2} + \ldots \, ,$$

$$\eta_r \equiv x_{t_1 + 1} \ldots x_p \equiv x_{t_2 + 1} \ldots x_p$$

and $t_1 \leqq t_2$.

Assume that S_m arises from R_m by deletion of the string $x_1 \ldots x_{t_1}$ (without any other change). Clearly $\|S_m\| = \|R_m\| \ominus [\alpha]$.

$$R_m \equiv \ldots (\ldots + \xi_1 + x_1^1 \ldots x_{t_1}^1 + \xi_2 + \ldots) \ldots (\ldots + x_1^2 \ldots x_{t_2}^2 + \ldots) \ldots \, ,$$

$$S_m \equiv \ldots (\ldots + \xi_1 + \xi_2 + \ldots) \ldots (\ldots + x_1^3 \ldots x_{t_2}^3 + \ldots) \ldots$$

(upper indices are only for distinguishing the same symbols in various places; $\xi_1, \xi_2 \in \Sigma^*$).

It is obvious that x_1^1, x_1^2, x_1^3 have the same "preindex", namely 0. Denote by $\iota(x_i^j)$ the index of x_i^j. Clearly, $S_1 \in \mathscr{S}(R_m)$, where $S_1 = (\iota(x_1^1) \vee \iota(x_1^2) \vee \ldots)$; hence

$S_2 \in \mathscr{S}(R_m)$, where $S_2 = (\iota(x_2^1) \vee \iota(x_2^2) \vee \dots)$ etc.; finally $S_{t_1} \in \mathscr{S}(R_m)$, where $S_{t_1} = (\iota(x_{t_1}^1) \vee \iota(x_{t_1}^2) \vee \dots)$. It is easy to see that no other state $s \in \mathscr{S}(R_m)$ contains any of the indices $\iota(x_1^1), \dots, \iota(x_{t_1}^3)$.

Assume $\mathscr{S}(R_m) = Q \cup \{s_1, \dots, s_{t_1}\}$, $Q \cap \{s_1, \dots, s_{t_1}\} = \emptyset$. Then $\mathscr{S}(S_m) = Q \cup \{\tilde{s}_1, \dots, \tilde{s}_{t_1}\}$, $Q \cap \{\tilde{s}_1, \dots, \tilde{s}_{t_1}\} = \emptyset$ again, where $\tilde{s}_i$ arises from s_i by replacing the couple $\iota(x_i^1) \vee \iota(x_i^2)$ by $\iota(x_i^3)$ $(i = 1, \dots, t_1)$. Furthermore, because no s_j contains $\iota(x_i^3)$ $(i, j = 1, \dots, t_1)$, it is $\tilde{s}_u \neq \tilde{s}_v$ for $u \neq v$ $(u, v = 1, \dots, t_1)$. Hence

$$\sigma(S_m) = \sigma(R_m)\,.$$

As follows from the lemma when searching for a minimal form, we can confine ourselves to *simple* expressions only. Really, if some $R \in \mathfrak{F}$ that is not simple is a minimal form, by the lemma there exists such a simple $S \in \mathfrak{F}$ that S is a minimal form (of the same language), too.

Let P be a polynomial. If we apply several times identity I_6 from right to left to P (i.e. we form a sequence $P_1, \dots, P_m$, where $P_1 \equiv P$, $P_{i+1} \equiv P_i'$ $(i = 1, \dots, m - 1)$), it is easy to see that here T in I_6 represents only *strings* belonging to Σ^* (not arbitrary expressions).

Now let R be an expression and examine the relation between the numbers $\sigma(R)$ and $\sigma(R')$ under the special condition mentioned above, i.e. R' arises from R by replacing P by P', where $P \equiv P_1\alpha + P_2\alpha$, $P' \equiv (P_1 + P_2)\,\alpha$ and $\alpha \equiv x_1 \dots x_k$, $\alpha \in \Sigma^*$. Write down some places and indices in P and P':

$$|P_1\underset{m}{|}x_1\underset{m+1}{|}\;x_2\underset{m+2}{|}\dots|x_k\underset{m+k}{|} + |P_2\underset{n}{|}\,x_1\underset{n+1}{|}\;x_2\underset{n+2}{|}\dots|x_k\underset{n+k}{|}\,, \tag{5}$$

$$|(|P_1\underset{m}{|} + |P_2\underset{n}{|})\underset{\substack{m\\n}}{|}\,x_1\underset{m+1}{|}\;x_2\underset{m+2}{|}\dots|x_k\underset{m+k}{|}\,. \tag{6}$$

Consider, how (5) differs from (6), i.e. R from R'. In (5) $n + 1$ x_1-follows n, in (6) $m + 1$ x_1-follows n. Further, in (5) $m + i + 1$ or $n + i + 1$ x_{i+1}-follows $m + i$ or $n + i$, respectively, while in (6) only $m + i + 1$ x_{i+1}-follows $m + i$ $(i = 1, \dots, k - 1)$. There are no other differences.

Thus the set $\mathscr{S}(R')$ is obtained from $\mathscr{S}(R)$ by replacing index $n + i$ in *all* states belonging to $\mathscr{S}(R)$ by index $m + i$ $(i = 1, \dots, k)$. Hence (because due to the replacing two or more identical states can arise)

$$\sigma(R') \leqq \sigma(R)\,. \tag{7}$$

In general, the inequality $\leqq$ in (7) cannot be replaced by $<$. The latter one, however, holds in "most" of concrete cases, as it can be proved.

Let R be a polynomial and let $R_1, \dots, R_m$ be a sequence of expressions such that $R_1 \equiv R$, $R_{i+1} \equiv R_i'$ $(i = 1, \dots, m - 1)$ and $R_m' \equiv R_m$. Then the expression R_m is called *total right bracketing* of R.

Assume an arbitrary finite language L is given. It is possible in many ways to form a polynomial that represents L. (We simply write down all strings of L in an arbitrary order and place between every two neighbouring strings the symbol $+$.) Further, to each such polynomial there exist various total right bracketings of it. We show, however, that *all such bracketings yield automata with the same number of states* and that *each such bracketing is a minimal form of the language L.*

Really, let P_1, P_2 be arbitrary polynomials that represent the language L and let T_i be an arbitrary total right bracketing of P_i $(i = 1, 2)$. It is clear that e.g. T_2 is also a total right bracketing of P_1. (Really, by definition of P_1' it is possible to apply to P_1 firstly several times identity I_4; in such a way P_2 can be obtained.)

In [2] (Theorem 3) it is proved that every two total right bracketings of an arbitrary polynomial can be obtained each from the other by multiple application of the identity I_4. Whence and by (1) it follows that

$$\sigma(T_1) = \sigma(T_2) .$$

Now suppose there is given an arbitrary (simple) expression $R \in \varrho(L)$. In [2] (Theorem 1) it is proved, that there is a sequence $P_1, \ldots, P_m, S_1, \ldots, S_n$ such that $P_1 \equiv R$, $P_{i+1} \equiv {}^{\mathrm{L}}P_i$, $S_1 \equiv P_m$, $S_{j+1} \equiv S_j^{\mathrm{R}}$ $(i = 1, \ldots, m-1;\ j = 1, \ldots, n-1)$ and S_n is a polynomial.

By (4) we then have

$$\sigma(P_m) = \sigma(R) \tag{8}$$

(if $P_{i+1} \equiv {}^{\mathrm{L}}P_i$, then ${}'P_{i+1} \equiv P_i$).

Further, there is a sequence $R_1, \ldots, R_k$ $(k \geqq n)$ such that $R_1 \equiv S_n$, $R_{i+1} \equiv R_i'$ $(i = 1, \ldots, k-1)$. $R_k' \equiv R_k$ and at the same time $R_i \equiv S_{n-i+1}$ $(i = 1, \ldots, n)$; particularly, $R_n \equiv S_1 \equiv P_m$.

Hence and by (7)

$$\sigma(R_k) \leqq \sigma(P_m)$$

and then (by (8))

$$\sigma(R_k) \leqq \sigma(R) .$$

(The formula (7) can be applied because in [2] the results we use were proved under the same special condition as (7) was.)

Thus the number of states of $\mathfrak{A}(R)$ *is not less* than the number of states of $\mathfrak{A}(T)$, where T is some (and then arbitrary) total right bracketing of a polynomial P which represents the language L. Hence each total right bracketing of P is a minimal form of the language L.

(Received October 21, 1969.)

[1] В. М. Глушков: Синтез цифровых автоматов. Физматгиз, Москва 1962.
[2] J. Mareš: Two Properties of Expressions in a Certain Free Universal Algebra. (In Czech.) Kybernetika *5* (1969), 3, 190–200.
[3] J. A. Brzozowski: Derivatives of Regular Expressions. Journal of ACM *11* (1964), 481–494.

VÝTAH

Poznámka ke Gluškovovu algoritmu syntézy konečných automatů

JAN MAREŠ

Při syntéze konečného automatu pomocí regulárních výrazů závisí počet vnitřních stavů konstruovaného automatu na tvaru regulárního výrazu, od kterého při syntéze vycházíme. V tomto článku je ukázáno, jak lze k libovolnému *konečnému* jazyku L nalézt takový regulární výraz R (reprezentující jazyk L), který je *minimální* v tom smyslu, že je-li S libovolný jiný regulární výraz reprezentující jazyk L, není počet stavů automatu zkonstruovaného pomocí S menší než počet stavů automatu zkonstruovaného pomocí R.

Jan Mareš, Matematický ústav ČSAV (Institute of Mathematics — Czechoslovak Academy of Sciences), Žitná 25, Praha 1.