Some Algorithms Concerning Uniquely Decipherable Codes

János Falucskai
Department of Mathematic and Informatics, College of Nyíregyháza
e-mail: falu@nyf.hu

The following problem plays an important role in code theory and its applications: Having a set of codewords we have to decide whether there are two or more sequences of codewords which form the same chain of characters of codewords. The problem can be approached in various ways, so the algorithms concerning uniquely decipherable codes use different devices for testing this property. The algorithm of Sardinas–Patterson is based on sequences of sets, other algorithms solve this problem by using finite automata.

The algorithm of Sardinas – Patterson.

The algorithm of Sardinas – Patterson is based on the following: Let us compute all the remainders in all attempts at a double factorization. It can recognize a double factorization by the fact that the empty word is one of the remainders.

Let A be a set, which we call an *alphabet*. A *word w* on the alphabet A is a finite sequence of elements of A

$$w = (a_1, a_2, \ldots, a_n), \quad a_i \in A$$

The set of all words on the alphabet A is denoted by A^*. If we omit the empty word from A^* then we get A^+. Let X and Y be two subsets of

A^+ and let $x \in X$ and $y \in Y$. Denote $X^{-1}Y$ the following set: w is an element of $X^{-1}Y$ if $xw = y$.

Let C be a subset of A^+, and let

$$\begin{aligned} U_1 &= C^{-1}C \setminus \{\varepsilon\} \\ U_2 &= C^{-1}U_1 \cup U_1^{-1}C \\ &\vdots \\ U_{n+1} &= C^{-1}U_n \cup U_n^{-1}C \end{aligned} \tag{1}$$

Theorem 1 *The set $C \subset A^+$ is a uniquely decipherable code if and only if none of the sets U_n defined above contains the empty word.*

Example 1 $K = \{00, 01, 011, 100\}$

$$\begin{aligned} U_1 = &\ K^{-1}K \setminus \{\varepsilon\} = \{1\} \\ U_2 = &\ K^{-1}U_1 \cup U_1^{-1}K = \{00\} \\ U_3 = &\ K^{-1}U_2 \cup U_2^{-1}K = \{\varepsilon\} \end{aligned}$$

Since U_3 contains the empty word, the code K is not a uniquely decipherable code.

Kayoko Tsuji's algorithm

Let us construct an automaton A_K for set K: $L(A_K)$ is the set of all ambiguous words in K.

Theorem 2 *The set K is a uniquely decipherable code if and only if $L(A_K)$ is empty.*

The automaton is constructed by the following way:
$P(K) = \{p \in K |\, pq \in K, q \neq \varepsilon\}$
$!\,\varphi : P(K) \to \mathbb{N} : 1 \leq \varphi(p) \leq card(P(K))$

$S(K) = \{s \in A^+ |\, qs \in K, q, s \neq \varepsilon\}$
$!\,\psi : S(K) \to \mathbb{N} : card(P(K)) + 1 \leq \psi(p)$

S: initial state

$\varphi(P(K))$: inner states

$S \xrightarrow{p} \varphi(p)$: path

if $u = p^{-1}x$, then $\varphi(p) \xrightarrow{u} \psi(u)$:path and $\psi(u)$ inner state

if $uv = x_1 \dots x_m$ and $\exists w : wv = x_m$, then $\psi(u) \xrightarrow{v} \psi(v)$:path and $\psi(v)$ inner state

$\psi(S(K) \cap K) \cap Q$: terminal states

Example 2 $K = \{01, 00, 011, 100\}$

$P(K) = \{01\}$

$S(K) = \{1, 0, 11, 00\}$

$\varphi(01) = 1$, $\psi(1) = 2, \psi(00) = 3, \psi(11) = 4, \psi(0) = 5$

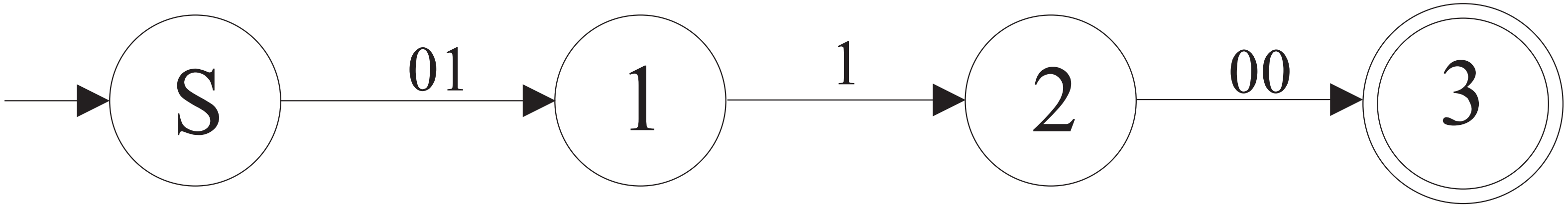
$S \xrightarrow{01} \varphi(01) = 1, \quad 1 \xrightarrow{1} \psi(1) = 2, \quad 2 \xrightarrow{00} \psi(00) = 3$

States:$S, 1, 2, 3$

Terminal states: $\psi(S(K) \cap K) \cap Q =$

$= \psi(\{1, 0, 11, 00\} \cap \{01, 00, 011, 100\}) \cap \{S, 1, 2, 3\} =$

$= \psi(00) \cap \{S, 1, 2, 3\} = \{3\} \cap \{S, 1, 2, 3\} = \{3\}$



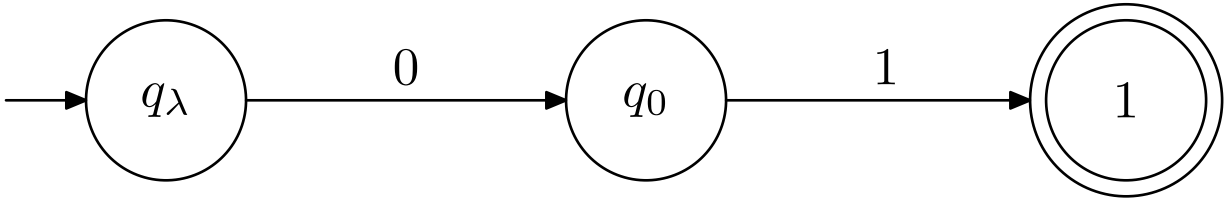
Kayako Tsuji's automaton for code $K = \{01, 00, 011, 100\}$

Our automaton to test codes

Our algorithm is based on theory of finite automata. With the aid of the automata of code words we construct an automaton for the code C over alphabet Δ. If the code word $w_i \in C$ is $x_1 x_2 \ldots x_n, \quad x_j \in \Delta$, then the automaton $\mathcal{A}(\{w_i\})$ is $\mathcal{A}(\{w_i\}) = (Q^{(i)}, q_\lambda, Q_F^{(i)}, A, \delta^{(i)})$. The set $Q^{(i)}$ is the set of states, the state q_λ is the initial state of the automaton $\mathcal{A}(\{w_i\})$ and the singleton $Q_F^{(i)}$ is the set of final state. $Q_F^{(i)} = \{i\}$ and $|Q^{(i)}| = length(w_i) + 1$. Since, the transition rules of automaton $\mathcal{A}(\{w_i\})$ are the followings:

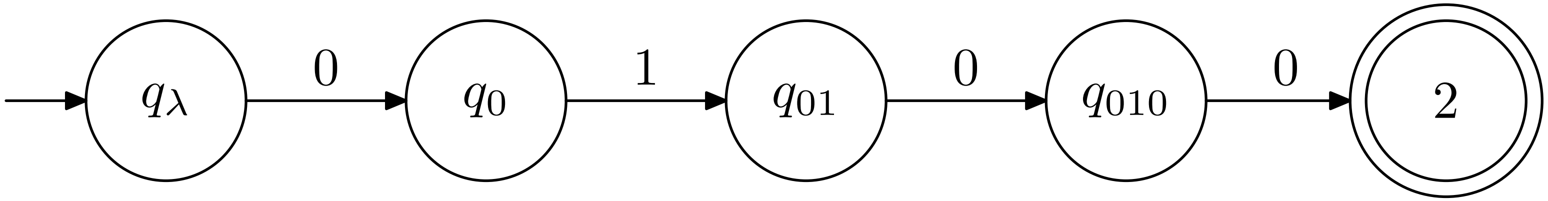
$$\begin{array}{rcl}
\delta(q_\lambda, x_1) & = & q_{x_1} \\
\delta(q_{x_1}, x_2) & = & q_{x_1 x_2} \\
 & \vdots & \\
\delta(q_{x_1 x_2 \ldots x_{n-2}}, x_{n-1}) & = & q_{x_1 x_2 \ldots x_{n-2} x_{n-1}} \\
\delta(q_{x_1 x_2 \ldots x_{n-1}}, x_n) & = & i
\end{array}$$

Thus, the automaton $\mathcal{A}(\{w_i\})$ accepts the code word w_i. The figure represents the automaton of code word $w_1 = \mathtt{01}$.



The automaton $\mathcal{A}(\{w_1\}) = \mathcal{A}(\{\mathtt{01}\})$

Let the code word w_1 be a prefix part of the code word w_2. Then the automaton $\mathcal{A}(\{w_1\})$ and the automaton $\mathcal{A}(\{w_2\})$ have states, which are signed by the same notations. Therefore, $(Q^{(1)} \setminus \{1\}) \subset Q^{(2)}$.



The automaton $\mathcal{A}(\{w_2\}) = \mathcal{A}(\{0100\})$

Since, the code word w_1 is a prefix part of the code word w_2, we can use the following notation:

$$w_1 = x_1 x_2 \cdots x_n; \quad w_2 = x_1 x_2 \cdots x_n x_{n+1} \cdots x_m$$

Denote $\delta^{(1)}$ the set of transition rules of the automaton $\mathcal{A}(\{w_1\})$ and denote $\delta^{(2)}$ the set of transition rules of the automaton $\mathcal{A}(\{w_2\})$. If the code word w_1 is a prefix part of code word w_2, then the sets of transition rules of the automata $\mathcal{A}(\{w_1\})$ and $\mathcal{A}(\{w_2\})$ are the followings.
The set $\delta^{(1)}$:

$$\begin{array}{lcl}
\{\delta(q_\lambda, x_1) & = & q_{x_1}, \\
\delta(q_{x_1}, x_2) & = & q_{x_1 x_2}, \\
 & \vdots & \\
\delta(q_{x_1 x_2 \ldots x_{n-2}}, x_{n-1}) & = & q_{x_1 x_2 \ldots x_{n-2} x_{n-1}}, \\
\delta(q_{x_1 x_2 \ldots x_{n-1}}, x_n) & = & 1\}
\end{array}$$

The set $\delta^{(2)}$:

$$\begin{array}{lcl}
\{\delta(q_\lambda, x_1) & = & q_{x_1}, \\
\delta(q_{x_1}, x_2) & = & q_{x_1 x_2}, \\
 & \vdots & \\
\delta(q_{x_1 x_2 \ldots x_{n-2}}, x_{n-1}) & = & q_{x_1 x_2 \ldots x_{n-2} x_{n-1}}, \\
\delta(q_{x_1 x_2 \ldots x_{n-1}}, x_n) & = & q_{x_1 x_2 \ldots x_{n-1} x_n}, \\
 & \vdots & \\
\delta(q_{x_1 x_2 \ldots x_{m-2}}, x_{m-1}) & = & q_{x_1 x_2 \ldots x_{m-2} x_{m-1}}, \\
\delta(q_{x_1 x_2 \ldots x_{m-1}}, x_m) & = & 2\}
\end{array}$$

Consequently,

$$(\delta^{(1)} \setminus \{\delta(q_{x_1 x_2 \ldots x_{n-1}}, x_n) = 1\}) \subset \delta^{(2)}$$

holds. Therefore,

$$\delta^{(1)} \cup \delta^{(2)} = \delta^{(2)} \cup \{\delta(q_{x_1 x_2 \ldots x_{n-1}}, x_n) = 1\}.$$

Let

$$\mathcal{A}(\{w_1, w_2\}) = (Q, q_\lambda, Q_F, A, \delta^{(1,2)}),$$

where

$$Q^{(1,2)} = Q^{(1)} \cup Q^{(2)},\ Q_F^{(1,2)} = \{1, 2\},$$

and

$$\delta^{(1,2)} = \delta^{(1)} \cup \delta^{(2)} \cup \{\delta(1, x_1) = q_{x_1}, \delta(2, x_1) = q_{x_1}\}.$$

We could see that

$$\begin{array}{lcll} \delta(q_{x_1 x_2 \ldots x_{n-1}}, x_n) & = & 1 & \in \delta^{(1)}, \\ \delta(q_{x_1 x_2 \ldots x_{n-1}}, x_n) & = & q_{x_1 x_2 \ldots x_{n-1} x_n} & \in \delta^{(2)}. \end{array}$$

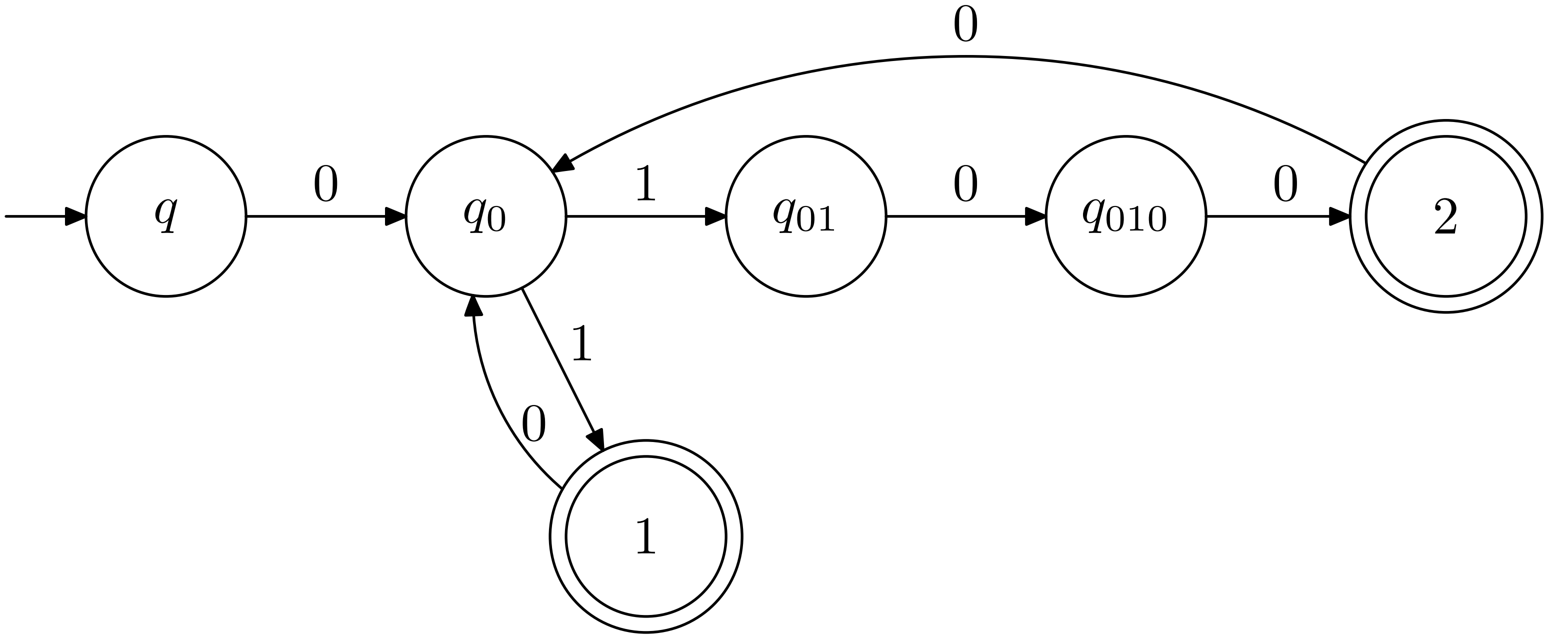
Thus, $\mathcal{A}(\{w_1, w_2\})$ is a nondeterministic automaton.

Let us consider the nondecipherable sequences of code words. It is obvious, that the different factorizations contain at least two code words such that one of them is prefix part of the other. If we join the automata of codewords by the method above, then we get the automaton $\mathcal{A}(\{w_1, \ldots, w_n\})$ for the code $C = \{w_1, \ldots, w_n\}$. Obviously, the automaton $\mathcal{A}(C)$ accepts exactly the language C^+.

Theorem 3 *If the automaton $\mathcal{A}(C)$ is deterministic, then the code C is decipherable.*

Proof 1 *If the automaton $\mathcal{A}(C)$ is deterministic, then the code C is prefix. Every prefix code is decipherable.*

Remark 1 *There are codes, which are nonprefix, but decipherable. For example the code $C = \{01, 0100\}$. That is, the automaton of a decipherable codes can be nondeterministic automaton. Thus, the theorem 3 is not reversible. We demonstrate the graphical presentation of the automaton $\mathcal{A}(\{01, 0100\})$.*



But, the code $\{\mathtt{01}, \mathtt{0100}\}$ *is decipherable. If we use our construction, then the automata of the nondecipherable codes are nondeterministic.*

The condition of the Theorem 3 is sufficient. Next, we give a necessary and sufficient condition of the decipherability. The construction is based on Theorem 4, which gives the relationship of the nondeterministic and its deterministic automata:

Theorem 4 *Every finite automaton equivalent to its deterministic finite automaton.*

If the string $v \in C^+$, then the automaton $\mathcal{A}(C)$ accepts v. That is, the automaton $\mathcal{A}(C)$ reads v and gets to a final state. If the code C is not uniquely decipherable, then we can follow different paths during reading v. We join these different paths by the equivalent deterministic automaton.

Let us construct the deterministic automaton $\mathcal{A}_D(C)$ for the automaton $\mathcal{A}(C)$. If we have two (or more) factorization of a string, then there exists a state of deterministic automaton such that the state contains at least two final states of nondeterministic automaton. Denote $Q_F^{\mathcal{A}(C)}$ the set of final states of the automaton $\mathcal{A}(C)$.

Theorem 5 *A code is decipherable if, and only if in the automaton $\mathcal{A}_D(C)$ at most one – being in the automaton $\mathcal{A}(C)$ – accept state appears on the right side of any transaction rule. That is, for any transaction rule the following holds: if the transaction rule $\delta(\{q_{i_1}, \ldots, q_{i_n}\}, x) = \{q_{j_1}, \ldots, q_{j_m}\}$ is in the automaton $\mathcal{A}_D(C)$, then don't exist $l \neq k$ such that, $q_{j_l} \in Q_F^{\mathcal{A}(C)}$ and $q_{j_k} \in Q_F^{\mathcal{A}(C)}$ hold.*

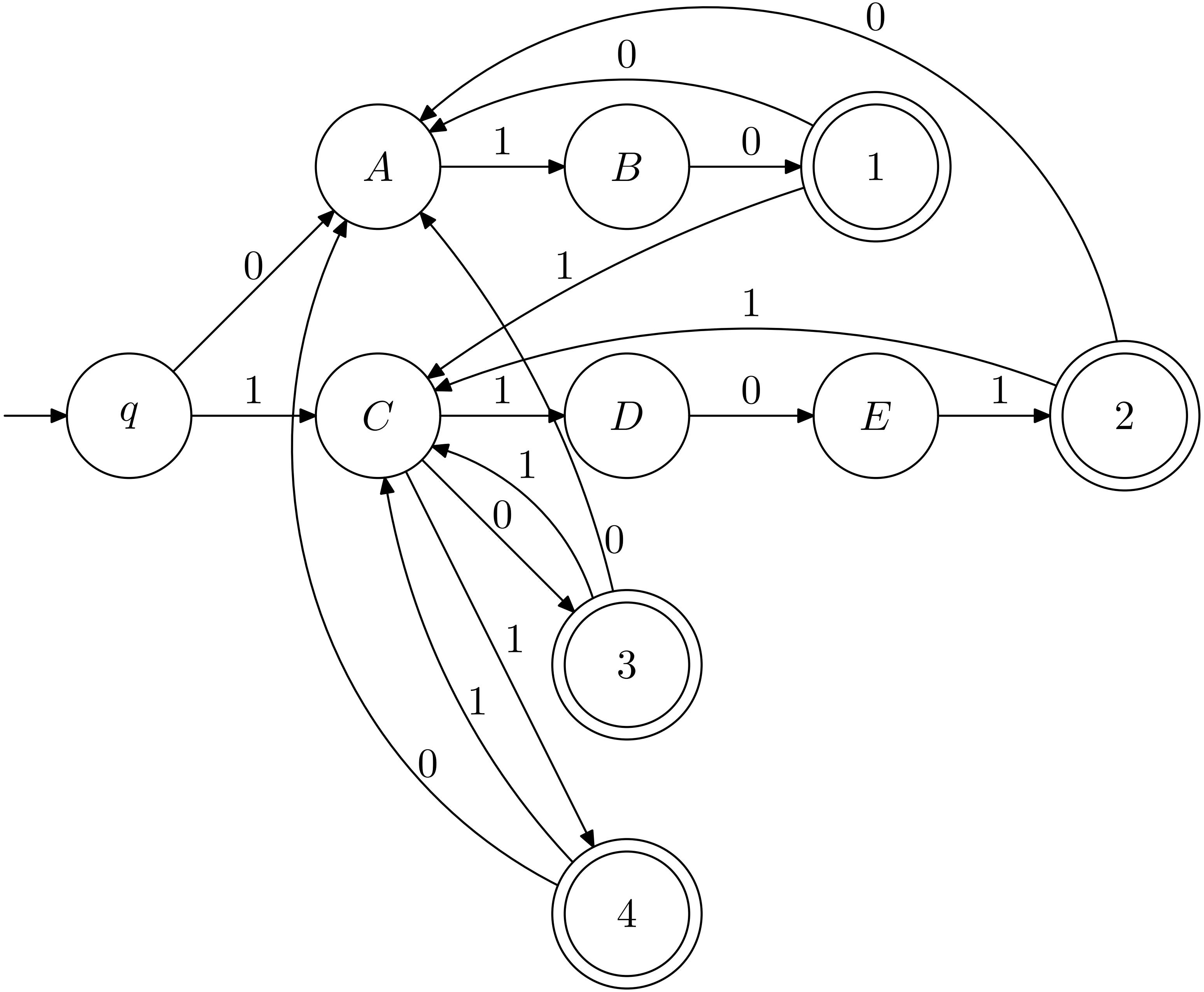
Proof 2 *The proof is carried out in an indirect proof. Assume that a code is decipherable and there exists a state of deterministic automaton such that* *the state contains at least two final states of nondeterministic automaton. That is, there exists a rule $\delta(\{q_{i_1}, \dots, q_{i_n}\}, x) = \{q_{j_1}, \dots, q_{j_m}\}$ in the automaton $\mathcal{A}_D(C)$ and exist $l \neq k$ such that, $q_{j_l} \in Q_F^{\mathcal{A}(C)}$ and $q_{j_k} \in Q_F^{\mathcal{A}(C)}$ hold. Denote v the sequence of symbols which are touched in the course of the path from the initial state q_λ to the state $\{q_{j_1}, \dots, q_{j_m}\}$. That is, $q_\lambda \xrightarrow{v} \{q_{j_1}, \dots, q_{j_m}\}$. Thus, the paths $q_\lambda \xrightarrow{v} q_{j_l}$ and $q_\lambda \xrightarrow{v} q_{j_k}$ are different success paths in the nondeterministic automaton. That is, the sequence v has two different factorization. Therefore, the code is nondecipherable. We have contradiction and the 'if' part is proved.*

To prove the 'only if' part we assume the following: There does not exist state of the deterministic automaton such that the state contains at least two final states of nondeterministic automaton and the code is nondecipherable. Thus, if the code is nondecipherable, then there exists at least two sequences of the codewords such that $w_{i_1} \cdots w_{i_n} = w_{j_1} \cdots w_{j_m}$ and

$w_{i_n} \neq w_{j_m}$. If we read the sequences, then we get to the same state because of the automaton is deterministic. Therefore, this state contains the final states i_n and j_m of the nondeterministic automaton because of the code words w_{i_n} and w_{j_m}. We have contradiction and the theorem is proved.

Example 3 *Let $C = \{\mathtt{010},\ \mathtt{1101},\ \mathtt{10},\ \mathtt{11}\}$.*

The automaton $\mathcal{A}(\{\mathtt{010},\ \mathtt{1101},\ \mathtt{10},\ \mathtt{11}\})$ of the code C is the following:

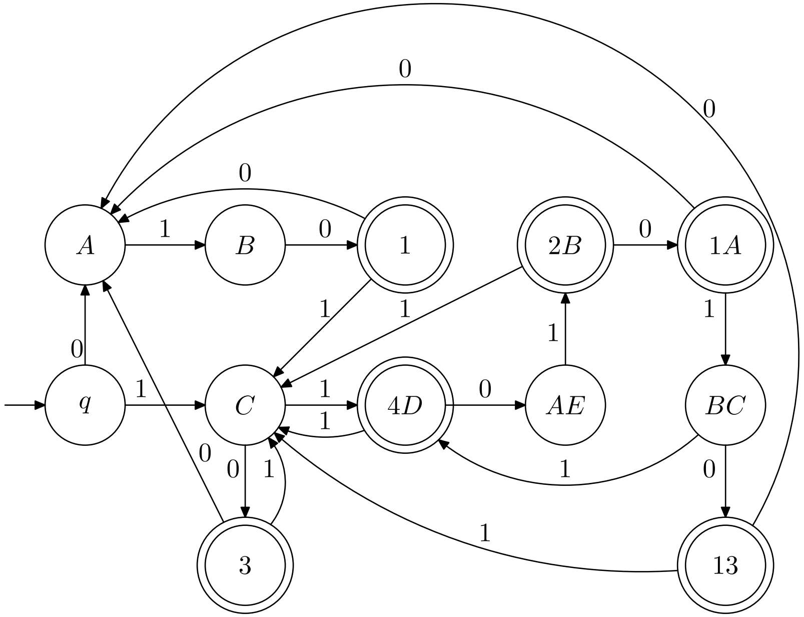


0
0
A
1
B
0
1
0
1
1
q
1
C
1
D
0
E
1
2
1
0
0
1
3
1
0
4

Let us construct its deterministic automaton $\mathcal{A}_D(\{010,\ 1101,\ 10,\ 11\})$*.* *The figure represents the automaton* $\mathcal{A}_D(\{010,\ 1101,\ 10,\ 11\})$*. Since,*

$$\delta(\{B, C\}, 0) = \{1, 3\},$$

the code is nondecipherable.



The automaton $\mathcal{A}_D(\{010,\ 1101,\ 10,\ 11\})$