An algorithm for recognition of $n$-collapsing words

I.V. Petrov

Department of Mathematics and Mechanics, Ural State University, 620083 Ekaterinburg, Russia

Abstract

A word $w$ over a finite alphabet $\Sigma$ is $n$-collapsing if for an arbitrary deterministic finite automaton $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$, the inequality $|\delta(Q, w)| \leq |Q| - n$ holds provided that $|\delta(Q, u)| \leq |Q| - n$ for some word $u \in \Sigma^+$ (depending on $\mathcal{A}$). We prove that the property of $n$-collapsing is algorithmically recognizable for any given positive integer $n$. We also prove that the language of all $n$-collapsing words is context-sensitive.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Deterministic finite automaton; $n$-collapsing word; Context-sensitive language

1. Main result and its application

Let $\mathcal{A} = \langle Q_{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}} \rangle$ be a deterministic finite automaton (DFA), where $Q_{\mathcal{A}}$ denotes the state set, $\Sigma$ stands for the input alphabet, and $\delta_{\mathcal{A}} : Q_{\mathcal{A}} \times \Sigma \rightarrow Q_{\mathcal{A}}$ is the transition function defining an action of the letters in $\Sigma$ on $Q_{\mathcal{A}}$. This action can be uniquely extended to an action $Q_{\mathcal{A}} \times \Sigma^* \rightarrow Q_{\mathcal{A}}$ of the free monoid $\Sigma^*$ over $\Sigma$ with the empty word $\lambda$; the latter action is still denoted by $\delta_{\mathcal{A}}$. Given a word $w \in \Sigma^*$ and a non-empty subset $X \subseteq Q_{\mathcal{A}}$, we write $\delta_{\mathcal{A}}(X, w)$ for the set $\{\delta_{\mathcal{A}}(x, w) \mid x \in X\}$ and say that the word $w$ acts on the set $X$. The difference $d_{\mathcal{A}}(w) = |Q_{\mathcal{A}}| - |\delta_{\mathcal{A}}(Q_{\mathcal{A}}, w)|$ is called the deficiency of the action of $w$ on the automaton $\mathcal{A}$.

Let $n$ be a positive integer. A DFA $\mathcal{A} = \langle Q_{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}} \rangle$ is said to be $n$-compressible if there is a word $w \in \Sigma^*$ such that $d_{\mathcal{A}}(w) \geq n$. The word $w$ is then called $n$-compressing with respect to $\mathcal{A}$. We note that there is a straightforward algorithm that verifies whether a given DFA is $n$-compressible; the time complexity of this algorithm is a quadratic polynomial of the number of states of the DFA.

A word $w \in \Sigma^*$ is said to be $n$-collapsing if $w$ is $n$-compressing with respect to every $n$-compressible DFA whose input alphabet is $\Sigma$. In other words, a word $w \in \Sigma^*$ is $n$-collapsing if for any DFA $\mathcal{A} = \langle Q_{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}} \rangle$ we have $d_{\mathcal{A}}(w) \geq n$ whenever $\mathcal{A}$ is $n$-compressible. Thus, such a word is a ‘universal tester’ whose action on the state set of an arbitrary DFA with a fixed input alphabet exposes whether or not the automaton is $n$-compressible.

It is known that $n$-collapsing words exist for every $n$ and over every finite alphabet $\Sigma$, see [7, Theorem 3.3] or [4, Theorem 2]. As the existence has been established, the next crucial step is to master, for each positive integer $n$, an algorithm that recognizes whether a given word is $n$-collapsing. This problem is non-trivial whenever $n > 1$ and $|\Sigma| > 1$ which will be assumed throughout. In [2], where the recognition problem was first formulated, it was solved...
for the case \( n = 2 \). A more geometric version of this solution was presented in [1]. The algorithm in [1] produces for a given word \( w \in \Sigma^* \) a finite number of inverse automata such that \( w \) is not \( 2 \)-collapsing if and only if at least one of these inverse automata can be completed to a \( 2 \)-compressible DFA \( \mathcal{A} = (\mathcal{Q}, \Sigma, \delta) \) with \( |\mathcal{Q}| < |w| \) and \( d f_w(\mathcal{A}) = 1 \) (here and below \(|w|\) stands for length of the word \( w \)).

No analogue of the algorithms from [1,2] is known for \( n \)-collapsing words with \( n > 2 \). Therefore, the author has tried another approach aiming to show that the language \( C_n(\Sigma) \) of all \( n \)-collapsing words over \( \Sigma \) is decidable in principle, i.e. \( C_n(\Sigma) \) is a recursive subset of \( \Sigma^* \). For this, it suffices to find, for each positive integer \( n \), a computable function \( f_n : \mathbb{N} \rightarrow \mathbb{N} \) such that a word \( w \in \Sigma^* \) is \( n \)-collapsing provided \( d f_w(\mathcal{A}) \geq n \) for every \( n \)-compressible DFA \( \mathcal{A} = (\mathcal{Q}, \Sigma, \delta) \) with \( |\mathcal{Q}| \leq f_n(|w|) \). Indeed, if such a function exists, then, given a word \( w \), one can calculate the value \( m = f_n(|w|) \) and then check the above condition through all automata with at most \( m \) states. Since there are only finitely many such automata with the input alphabet \( \Sigma \), the procedure will eventually stop. If in the course of the procedure one encounters an \( n \)-compressible DFA \( \mathcal{A} \) with \( d f_w(\mathcal{A}) < n \), then \( w \) is not \( n \)-collapsing by the definition. If no such automaton is found, then \( w \) is \( n \)-collapsing by the choice of the function \( f_n \).

From the results of [1] it follows that, for \( n = 2 \), the function \( f_2(|w|) = \max\{3, |w| - 1\} \) satisfies the desired property. The author first managed to show that the functions \( f_n(|w|) = 3|w|(n-1) + n + 1 \) satisfy the desired property for every \( n \). This result was announced (with an outline of the proof) in the survey paper [3]. Here we improve this result by showing that some smaller functions, namely \( f_n(|w|) = 2|w|(n-1) + 2 \), do the job as well. Thus, the main result of the present paper is the following theorem.

**Theorem 1.** Let \( w \in \Sigma^* \) be a word which is not \( n \)-collapsing. Then there exists an \( n \)-compressible automaton \( \mathcal{B} = (\mathcal{Q}, \Sigma, \delta_\mathcal{B}) \) with \( |\mathcal{Q}| \leq 2|w|(n-1) + 2 \) such that \( d f_w(\mathcal{B}) < n \).

Since the function \( f_n(|w|) = 2|w|(n-1) + 2 \) is linear (with respect to \(|w|\)), we immediately obtain a non-deterministic linear space and polynomial time algorithm recognizing the complement of the language \( C_n(\Sigma) \) of all \( n \)-collapsing words over \( \Sigma \); the algorithm simply makes a guess consisting of a DFA \( \mathcal{A} \) with \( |\mathcal{Q}| \leq 2|w|(n-1) + 2 \) and then verifies that \( \mathcal{A} \) is \( n \)-compressible and that \( w \) is not \( n \)-collapsing with respect to \( \mathcal{A} \). By classical results of formal language theory (cf. [5, Sections 2.4 and 2.5]), this implies that the language \( C_n(\Sigma) \) is context-sensitive. We mention that Pribavkina [6] has shown that the language \( C_2(\Sigma) \) with \(|\Sigma| = 2 \) is not context-free. For the case when either \( n > 2 \) or \(|\Sigma| > 2 \), the problem of locating the language \( C_n(\Sigma) \) with respect to the Chomsky hierarchy still remains open.

### 2. The proof of Theorem 1

It is convenient for us to think of each DFA \( \mathcal{A} = (\mathcal{Q}, \Sigma, \delta_\mathcal{A}) \) as a digraph with the vertex set \( \mathcal{Q} \). We denote by \((p, a, q)\) the edge from \( p \in \mathcal{Q} \) to \( q \in \mathcal{Q} \) labeled by the letter \( a \in \Sigma \). We shall identify the transition function \( \delta_\mathcal{A} : \mathcal{Q} \times \Sigma \rightarrow \mathcal{Q} \) with its graph \( \{(v, a, \delta_\mathcal{A}(v, a)) \mid v \in \mathcal{Q}, a \in \Sigma\} \); that is, the expressions \((p, a, q) \in \delta_\mathcal{A}\) and \( \delta_\mathcal{A}(p, a) = q \) mean the same. We denote the set \( \{(v, a, \delta_\mathcal{A}(v, a)) \mid v \in \mathcal{Q}\} \) by \( \delta_\mathcal{A}(\bullet, a) \), i.e. \( \delta_\mathcal{A}(\bullet, a) \) is the set of all edges labeled by \( a \); on the other hand, \( \delta_\mathcal{A}(\bullet, a) \) is the transformation of the set \( \mathcal{Q} \) under applying the letter \( a \).

We need some notations and definitions. Let \( u \) be a word in \( \Sigma^* \). We denote by \( u[k] \) and \( u_k \) the \( k \)th letter and the prefix of length \( k \) of the word \( u \) (\( k \leq |u| \)). That is if \( u = a_1a_2\ldots a_l \), then \( u[k] = a_k \) and \( u_k = a_1a_2\ldots a_k \) respectively. Furthermore, by definition put \( u_0 = \lambda \).

If \( u, v \) are words over \( \Sigma \) and \( u = vv'' \) for some \( v', v'' \in \Sigma^* \), we say that \( v \) is a factor of \( u \). It is convenient to have a name for the property of a word \( w \in \Sigma^* \) to have all words of length \( n \) among its factors. We say that such a word \( w \) is \( n \)-full. We say that an \( n \)-compressable automaton \( \mathcal{A} \) is \( n \)-proper if no word of length \( n \) is \( n \)-compressing with respect to \( \mathcal{A} \).

The following lemma is a direct corollary of [2, Lemma 2.1].

**Lemma 2.** If a word \( w \) is not \( n \)-full, then there is an \( n \)-compressible automaton \( \mathcal{A} = (\mathcal{Q}, \Sigma, \delta_\mathcal{A}) \) such that \(|\mathcal{Q}| \leq |w| \) and \( d f_w(\mathcal{A}) < n \).

In view of Lemma 2, in the sequel we consider only \( n \)-full words. Fix an \( n \)-full word \( w \in \Sigma^* \) which is not \( n \)-collapsing and consider an \( n \)-compressible DFA \( \mathcal{A} = (\mathcal{Q}, \Sigma, \delta_\mathcal{A}) \) such that \( d f_w(\mathcal{A}) < n \). The word \( w \) has every word of length \( n \) as a factor whence the automaton \( \mathcal{A} \) is \( n \)-proper. Suppose \( d f_w(\mathcal{A}) = k < n \). In this case we extend the automaton \( \mathcal{A} \) to a new automaton \( \mathcal{B} = (\mathcal{Q}, \Sigma, \delta_\mathcal{B}) \) with \( d f_w(\mathcal{B}) = n - 1 \). For this, we append \( n - k \) new states
By extending the transition function to these new states by letting 
\( q_i \), \( i = 1, \ldots, n - k \) and extend the transition function to these new states by letting \( \delta_B(q_i, a) = q_1 \) for all \( i = 1, \ldots, n - k \) and all \( a \in \Sigma \). The following lemma is a direct corollary of the definition of \( B \).

**Lemma 3.** The DFA \( B \) is an \( n \)-proper and \( n \)-compressible automaton, and \( d_{fw}(B) = n - 1 \).

Now assume that some of the states of the DFA \( B \) are covered by tokens and the action of any letter \( a \in \Sigma \) redistributes the tokens according to the following rule: a state \( q \in Q_B \) will be covered by a token after the action of \( a \) if and only if there exists a state \( q' \in Q_B \) such that \( \delta_B(q', a) = q \) and \( q' \) was covered by a token before the action. In more ‘visual’ terms, the rule amounts to saying that tokens slide along the edges labeled by \( q \) if and only if there exists a state \( q' \) such that \( \delta_B(q', a) = q \) and \( q' \) was covered by a token before the action.

Fig. 1 illustrates this rule: its right part shows how tokens are distributed over the state set of a DFA after completing the action of the letter \( a \) on the distribution shown on the left. It is convenient to call a state *empty* if it is not currently covered by a token.

Let \( \ell = |w| \). We cover all states in \( Q_B \) by tokens and let the letters \( w[1], \ldots, w[\ell] \) act in succession. On the \( k \)th step of this procedure we mark all elements of the following sets of states:

\[
\begin{align*}
M(1, k) &= Q_B \setminus \delta_B(Q_B, w_k); \\
M(2, k) &= \delta_B(Q_B \setminus \delta_B(Q_B, w_{k-1}), w[k]) = \delta_B(M(1, k - 1), w[k]).
\end{align*}
\]

The meaning of these sets can be easily explained in terms of the distribution of tokens before and after the action of the letter \( w[k] \). The set \( M(1, k) \) consists of empty states after the action of the letter \( w[k] \). The set \( M(2, k) \) is the set of all states to which the letter \( w[k] \) brings states that had been empty before the action of \( w[k] \). Note that \( M(2, 1) = \emptyset \) because there is no empty state before the action of the first letter of \( w \).

For example, assume that the transition shown in Fig. 1 represents the \( k \)th step of the above procedure (so that \( w[k] = a \)). Then three states get marks as shown on Fig. 2. Indeed, \( M(2, k) = \{4\} \) because 3 was the only empty state before the action of \( a \) and \( \delta_B(3, a) = 4 \). Further, \( M(1, k) = \{2, 5\} \).

Put \( M = \bigcup_{1 \leq k \leq \ell} (M(1, k) \cup M(2, k)) \). We call \( M \) the set of marked states of the DFA \( B \) or the marked set for short.

The next proposition registers an important property of the marked set.

**Proposition 4.** Let \( a \in \Sigma \), \( p, r \in Q_B \) and \( p \neq r \). If \( \delta_B(p, a) = \delta_B(r, a) \), then \( \delta_B(p, a) \in M \).

**Proof.** Since the word \( w \) is \( n \)-full, it has at least one factor \( a^n \). Let \( w = w_i a^n v \). The automaton \( B \) is \( n \)-proper whence \( d_{fw}(B) \leq n - 1 \). Therefore the non-increasing chain \( Q_B \supseteq \delta_B(Q_B, a) \supseteq \delta_B(Q_B, a^2) \supseteq \cdots \) stabilizes after at most \( n - 1 \) steps whence \( a \) acts on the set \( \delta_B(Q_B, a^{n-1}) \) as a permutation.

Let \( q = \delta_B(p, a) = \delta_B(r, a) \). If \( q \notin \delta_B(Q_B, a^{n-1}) \) then

\[
q \in Q_B \setminus \delta_B(Q_B, a^{n-1}) \subseteq Q_B \setminus \delta_B(Q_B, w_i a^n v) = M(1, i + n - 1) \subseteq M.
\]
Now assume $q \in \delta(\mathcal{B}, Q_a^n)$. The states $p$ and $r$ cannot simultaneously belong to $\delta(Q_a^n)$ because $a$ acts as a permutation on this set while $\delta(p, a) = \delta(r, a)$. Without loss of generality, assume that $p \notin \delta(Q_a^n)$. Then

\[ p = Q_a^n \delta(\mathcal{B}, Q_a^n) \subseteq M(1, i + n - 1). \]

Therefore $q \in M(2, i + n) \subseteq M$. \(\Box\)

An edge $e = (q_1, a, q_2)$ of the automaton $\mathcal{B}$ is called:

- **inner** if it connects two marked states of $\mathcal{B}$, i.e. $q_1, q_2 \in M$. By $IE(\mathcal{B}, a)$ we denote the set of all inner edges of the automaton $\mathcal{B}$ labeled by $a$. Let $IE(\mathcal{B}) = \bigcup_{a \in \Sigma} IE(\mathcal{B}, a)$.

- **outgoing** if its starting point is marked while its end point is not, i.e. $q_1 \in M, q_2 \notin M$. By $\mathcal{M}(\mathcal{B}, a)$ we denote the set of all outgoing edges of the automaton $\mathcal{B}$ labeled by $a$.

- **ingoing** if its end point is marked while its starting point is not, i.e. $q_1 \notin M, q_2 \in M$. By $\mathcal{M}(\mathcal{B}, a)$ we denote the set of all ingoing edges of the automaton $\mathcal{B}$ labeled by $a$.

**Lemma 5.** After the action of the prefix $w_{k-1}$, the initial vertex of every outgoing edge $e = (q_1, w[k], q_2)$ labeled by the letter $w[k]$ holds a token.

**Proof.** Arguing by contradiction, suppose that the state $q_1$ is empty after the action of $w_{k-1}$. Then the state $q_2$ belongs to the set $M$ by the definition of $M(2, k)$, whence the edge $e = (q_1, w[k], q_2)$ is inner, a contradiction. \(\Box\)

**Lemma 6.** After the action of the prefix $w_{k-1}$, the initial vertex of every ingoing edge $e = (q_1, w[k], q_2)$ labeled by the letter $w[k]$ holds a token.

**Proof.** Since the edge $e$ is ingoing, the state $q_1$ does not belong to the marked set $M$. Hence, the state $q_1$ never becomes empty. \(\Box\)

**Proposition 7.** For each letter $a \in \Sigma$, the numbers of ingoing and outgoing edges labeled by $a$ in the automaton $\mathcal{B}$ are equal.

**Proof.** Let $M = Q_\mathcal{B} \setminus M$ be the complement of the marked set $M$. By the definition of $M(1, k)$ $(1 \leq k \leq \ell)$, after the action of the word $w_k$ all empty states belong to the set $M$ and hence all states of the set $\overline{M}$ are covered by tokens. Therefore the number of tokens in $\overline{M}$ is equal to $|\overline{M}|$ and remains constant all the time.

For a given letter $a$, we denote by $I_a$ and $O_a$ the number of ingoing and respectively outgoing edges labeled by $a$. Since the word $w$ is $n$-full, there is a position $i$, $1 \leq i \leq \ell$, such that $w[i] = a$.

Consider the action of the letter $w[i]$ and check how it affects the number of the tokens in $\overline{M}$. The number of tokens leaving the set $\overline{M}$ is equal to $I_a$ by Lemma 6. The number of tokens coming to the set $\overline{M}$ is equal to $O_a$ by Lemma 5. Any token in $\overline{M}$ which is removed after the action leaves the set $\overline{M}$. Indeed, it moves along the edge $(p, w[i], q)$ which shares its end point $q$ with another edge $(r, w[i], q)$. The state $q$ is not in $\overline{M}$ by Proposition 4.

We see that after the action of $w[i]$ the number of tokens in $\overline{M}$ is equal to $|\overline{M}| + O_a - I_a$ and, on the other hand, it is always equal to $|\overline{M}|$. Therefore $O_a = I_a$.

Now we are ready to extract from the automaton $\mathcal{B}$ a new automaton $\mathcal{C} = (Q_\mathcal{C}, \Sigma, \delta_\mathcal{C})$. The state set of this new automaton coincides with the marked set $M$ of the automaton $\mathcal{B}$ and the transitions between their states are the inner edges of the automaton $\mathcal{B}$, i.e. $Q_\mathcal{C} = M$ and $\delta_\mathcal{C} = IE(\mathcal{B})$. In general, the automaton $\mathcal{C}$ is not complete because the automaton $\mathcal{B}$ may have outgoing edges.

We complete the automaton $\mathcal{C}$ to a DFA and simultaneously define two maps $\psi_{\text{start}}$ and $\psi_{\text{end}}$. To start with, we put $\psi_{\text{start}}(e) = \psi_{\text{end}}(e) = e$ for each inner edge $e \in IE(\mathcal{B})$ of the automaton $\mathcal{B}$.

Now consider a letter $a \in \Sigma$. By Proposition 7 there is a bijection $\varphi_a : \mathcal{M}, a) \to \mathcal{M}(\mathcal{B}, a)$.

We fix such a bijection $\varphi_a$ and do the following for each outgoing edge $e = (q_1, a, q_2)$. Let $q_3, a, q_4$. It is an ingoing edge. We append a new edge $f = (q_1, a, q_4)$ to the automaton $\mathcal{C}$, connecting the starting point
of $e$ with the end point of $\varphi_a(e)$. We call each such edge an outer edge of the automaton $\mathcal{C}$. Then we define $\psi_{\text{start}}(e) = \psi_{\text{end}}(\varphi_a(e)) = f$.

By performing the described operation for each letter $a \in \Sigma$, we obtain a complete DFA which we still denote by $\mathcal{C}$. This should not lead to any confusion since from now on we shall use the completed version of $\mathcal{C}$ only.

The inner edges of the automaton $\mathcal{B}$ will be also called the inner edges of the automaton $\mathcal{C}$. Now we define

$$IE(\mathcal{C}, a) = \{e = (q_1, a, q_2) | e \text{ is an inner edge in } \mathcal{C}\},$$

$$ME(\mathcal{C}, a) = \{e = (q_1, a, q_2) | e \text{ is an outer edge in } \mathcal{C}\}.$$

Observe that $Q_{\mathcal{B}} = M \subseteq Q_{\mathcal{B}}$, and hence we can apply to any state $q \in M$ the transition functions of both $\mathcal{B}$ and $\mathcal{C}$.

For each letter $a \in \Sigma$ we have $\delta_{\mathcal{C}}(\bullet, a) = IE(\mathcal{C}, a) \cup ME(\mathcal{C}, a)$ and $\delta_{\mathcal{B}}(\bullet, a)|_M = \{(q_1, a, q_2) \in \delta_{\mathcal{B}}(\bullet, a)|q_1 \in M\} = IE(\mathcal{B}, a) \cup ME(\mathcal{B}, a)$.

Observe that the mappings

$$\psi_{\text{start}} : IE(\mathcal{B}, a) \cup ME(\mathcal{B}, a) \rightarrow IE(\mathcal{C}, a) \cup ME(\mathcal{C}, a)$$

and

$$\psi_{\text{end}} : IE(\mathcal{B}, a) \cup ME(\mathcal{B}, a) \rightarrow IE(\mathcal{C}, a) \cup ME(\mathcal{C}, a)$$

are bijections. Both these bijections map inner edges to inner edges. The mapping $\psi_{\text{start}}$ maps outgoing edges to outer edges and $\psi_{\text{end}}$ maps ingoing edges to outer edges. The mapping $\psi_{\text{start}}$ preserves starting points of edges and $\psi_{\text{end}}$ preserves end points of edges.

**Proposition 8.** $|Q_{\mathcal{C}}| = |M| \leq (2\ell - 1)(n - 1)$.

**Proof.** Since $d_{w}(\mathcal{B}) = n - 1$, there are at most $n - 1$ empty states of $\mathcal{B}$ during the action of the word $w$ on the automaton $\mathcal{B}$, that is $|Q_{\mathcal{B}} \setminus \delta_{\mathcal{B}}(Q_{\mathcal{B}}, w_k)| \leq n - 1$ for all $k$, $0 \leq k \leq \ell$. Hence, the inequalities $|M(1, k)| \leq n - 1$ and $|M(2, k)| \leq n - 1$ hold for every $k$, $1 \leq k \leq \ell$. The set $M(2, 1)$ is empty. Thus,

$$|M| = \left| \bigcup_{k=1}^{\ell} M(1, k) \cup \bigcup_{k=2}^{\ell} M(2, k) \right| \leq (2\ell - 1)(n - 1).$$

Now we need an auxiliary construction. Let $s = a_1a_2\cdots a_\ell \in \Sigma^*$ be an arbitrary word and $a \in \Sigma$ be an arbitrary letter. We define an automaton $\mathcal{L}(s, a)$. We start with the incomplete automaton whose state set is $Q_{\mathcal{L}} = \{b_1, b_2, \ldots, b_{\ell+1}\}$ and whose edges are

$$(b_1, a_1, b_2), (b_2, a_2, b_3), \ldots, (b_\ell, a_\ell, b_{\ell+1}), (b_{\ell+1}, a, b_1).$$

After that we complete the automaton to a permutation automaton over $\Sigma$ in an arbitrary way. Finally, we remove the edge $(b_{\ell+1}, a, b_1)$. We call the incomplete automaton $\mathcal{L}(s, a) = (Q_{\mathcal{L}}, \Sigma, \delta_{\mathcal{L}})$ the buffer automaton of the word $s$ with the input–output letter $a$. The state $b_1$ of the automaton $\mathcal{L}(s, a)$ is called the key state and is denoted by $KS(\mathcal{L})$.

It is convenient to imagine that instead of the removed edge $(b_{\ell+1}, a, b_1)$, the automaton $\mathcal{L}(s, a)$ has got two ‘open’ edges $(\bullet, a, b_1)$ and $(b_{\ell+1}, a, \bullet)$ with undefined starting and end points respectively as shown on Fig. 3. We shall use such undefined starting and end points to attach buffer automata to the automaton $\mathcal{C}$. 

![Fig. 3. Building a buffer automaton.](image-url)
Let $e = (q_1, a, q_2)$ be an outer edge of the automaton $\mathcal{C}$ and let $\mathcal{L}(s, a) = \langle Q_\mathcal{L}, \Sigma, \delta_\mathcal{L} \rangle$ be an arbitrary buffer automaton whose input–output letter is $a$.

Then we define the operation $\mathcal{C} \oplus \mathcal{L}(s, a)$ of attaching the buffer automaton $\mathcal{L}(s, a)$ to the DFA $\mathcal{C}$ instead of the edge $e$ (Fig. 4). The result of this operation is a new automaton $\mathcal{T} = \langle Q_\mathcal{T}, \Sigma, \delta_\mathcal{T} \rangle$ defined as follows:

$$Q_\mathcal{T} = Q_\mathcal{C} \cup Q_\mathcal{L} = Q_\mathcal{C} \cup \{b_1, b_2, \ldots, b_{t+1}\}$$

$$\delta_\mathcal{T}(q, c) = \begin{cases} 
\delta_\mathcal{C}(q, c), & \text{if } q \in Q_\mathcal{C} \setminus \{q_1\} \\
\delta_\mathcal{C}(q, c), & \text{if } q = q_1, c \neq a \\
b_1, & \text{if } q = q_1, c = a \\
\delta_\mathcal{L}(q, c), & \text{if } q \in \{b_1, b_2, \ldots, b_t\} \\
\delta_\mathcal{L}(q, c), & \text{if } q = b_{t+1}, c \neq a \\
q_2, & \text{if } q = b_{t+1}, c = a.
\end{cases}$$

We call the state $q_2$ the output of the attached buffer automaton $\mathcal{L}(s, a)$ and we denote this state by $\text{out} \mathcal{L}$.

Let $\{(p_i, a_i, q_i) = e_i | i \in \{1, \ldots, r\}\} \subseteq \hat{M}(\mathcal{C})$ be a subset of the set of outer edges of $\mathcal{C}$ and let $\{x_i | x_i \in \Sigma^*\}_{i=1}^r$ be a set of words. We can attach buffer automata simultaneously instead of $r$ outer edges of the automaton $\mathcal{C}$. We denote the result of this operation by

$$\mathcal{D} = \langle Q_\mathcal{D}, \Sigma, \delta_\mathcal{D} \rangle = \mathcal{C} \oplus \mathcal{L}_1(x_1, a_1) \oplus \cdots \oplus \mathcal{L}_r(x_r, a_r).$$

(1)

The automaton $\mathcal{D}$ depends on the choice of the set of outer edges $e_i$ and the choice of the set of words $x_i$. Therefore we have a series of automata of the form (1).

**Proposition 9.** Any automaton $\mathcal{D}$ of the form (1) is a DFA.

**Proof.** It is obvious due to the definitions of a buffer automaton and the operation of attaching a buffer automaton.

The next lemma gives an important property of these automata.

**Lemma 10.** Every DFA $\mathcal{D}$ of the form (1) satisfies the condition

$$Q_\mathcal{D} \setminus \delta_\mathcal{D}(Q_\mathcal{D}, w_k) = Q_\mathcal{D} \setminus \delta_\mathcal{D}(Q_\mathcal{D}, w_k)$$

for every $k$, $0 \leq k \leq \ell$.

**Proof.** Let $\hat{M} = Q_\mathcal{D} \setminus M$. Let $L = \bigcup_{i=1}^r Q_{\mathcal{L}_i}$, where $Q_{\mathcal{L}_i}$ is the state set of the buffer automaton $\mathcal{L}_i$. Then the state sets of the automata $\mathcal{D}$ and $\mathcal{D}$ can be represented as $Q_\mathcal{D} = M \cup \hat{M}$ and $Q_\mathcal{D} = M \cup L$.

Let $D_k = Q_\mathcal{D} \setminus \delta_\mathcal{D}(Q_\mathcal{D}, w_k)$, $B_k = Q_\mathcal{D} \setminus \delta_\mathcal{D}(Q_\mathcal{D}, w_k)$ be the sets of empty states of the automata $\mathcal{D}$ and respectively $\mathcal{D}$ after the action of the prefix $w_k$. Arguing by contradiction, we choose the minimal integer $k$ ($0 \leq k \leq \ell$) with the property $D_k \neq B_k$. Then there is a state $q \in (B_k \setminus D_k) \cup (D_k \setminus B_k)$. It is clear that

$q \in (B_k \setminus D_k) \cup (D_k \setminus B_k) \subseteq B_k \cup D_k \subseteq \hat{M} \cup M \cup L$. 

Fig. 4. Attaching a buffer automaton.
First we show that \( q \notin L \), then that \( q \notin \overline{M} \) and finally that \( q \notin M \). This will yield a contradiction as desired.

**Step 1.** We prove that \( q \notin L \).

It is clear that \( k \neq 0 \), since \( w_0 \) is the empty word and \( D_0 = \emptyset = B_0 \). Hence by the choice of \( k \) we have
\[
D_{k-1} = B_{k-1} = Q_{\mathcal{D}} \setminus \delta_{\mathcal{D}}(Q_{\mathcal{D}}, w) = M(1, k - 1) \subseteq M.
\]

In particular, all states of the buffer automata \( \mathcal{L}_i \) \( (1 \leq i \leq r) \) are covered by tokens after the action of \( w_{k-1} \) on the set \( Q_{\mathcal{D}} \), because \( Q_{\mathcal{D}} \cap M = \emptyset \).

Let \( w_k = w_{k-1} \). Consider an arbitrary buffer automaton \( \mathcal{L}_i \) attached instead of the edge \( e_i \). If \( e_i = (p_i, a, q_i) \), then the automaton \( \mathcal{L}_i \) is covered by tokens after the action of the word \( w_k \) on \( \mathcal{D} \) because \( a \) acts as a permutation on the set \( Q_{\mathcal{D}} \) by the definition of buffer automata.

**Step 2.** We prove that \( q \notin \overline{M} \). Indeed, by the definition of the set \( M(1, k) \), we have \( \overline{M} \subseteq \delta_{\mathcal{D}}(Q_{\mathcal{D}}, w_k) \) whence \( q \notin \overline{M} \).

**Step 3.** We prove that \( q \notin M \).

Suppose that \( q \in M \). We divide the proof into two cases.

**Case 1.** The state \( q \) is the end point of some ingoing edge \( f = (p, a, q) \) of the automaton \( \mathcal{D} \).

The state \( p \in \overline{M} \) is covered by a token after the action of \( w_{k-1} \) by Lemma 6. Hence \( q \) is covered by a token after the action of \( w_k \) in the automaton \( \mathcal{D} \). Thus, \( q \notin B_k \). The edge \( \psi_{\text{end}}(f) = (s, a, q) \) is an outer edge of the automaton.

If this edge is replaced in \( \mathcal{D} \) by a buffer automaton, then, as we have shown on Step 1, \( q \in \delta_{\mathcal{D}}(Q_{\mathcal{D}}, w_k) \), that is, \( q \notin D_k \). This contradicts the condition \( q \in B_k \cup D_k \).

If the edge \( \psi_{\text{end}}(f) = (s, a, q) \) is not replaced in \( \mathcal{D} \) by a buffer automaton, then the state \( s \) is the starting point of an outgoing edge \( \psi_{\text{start}}(\psi_{\text{end}}(f)) \). Hence by Lemma 5 the state \( s \) is covered after the action of \( w_{k-1} \) in the automaton \( \mathcal{D} \).

In view of the equality \( B_{k-1} = D_{k-1} \) this implies that the state \( s \) is covered after the action of \( w_{k-1} \) in the automaton \( \mathcal{D} \). Therefore \( q \) is covered after the action of \( w_k \) in the automaton \( \mathcal{D} \), that is \( q \notin D_k \). This again contradicts the condition \( q \in B_k \cup D_k \).

**Case 2.** There is no ingoing edge labeled by the letter \( a \) with the end point \( q \).

This means that there is no outer edge labeled by \( a \) with the end point \( q \) in \( \mathcal{D} \). Therefore any edge \( e = (p, a, q) \) in \( \mathcal{B} \) or in \( \mathcal{D} \) is an inner edge. Thus, \( p \in M \). The sets of inner edges of the automata \( \mathcal{B} \) and \( \mathcal{D} \) coincide. If there is an edge \( e = (p, a, q) \) such that \( p \notin B_{k-1} = D_{k-1} \), then \( q \notin B_k \) and \( q \notin D_k \). If there is no edge \( e = (p, a, q) \) such that \( p \notin B_{k-1} = D_{k-1} \), then \( q \in B_k \) and \( q \in D_k \). Both these conclusions contradict the condition \( q \in (B_k \setminus D_k) \cup (D_k \setminus B_k) \).

**Corollary 11.** For any DFA \( \mathcal{D} \) of the form (1), \( d_{f_k}(D) = d_{f_0}(\mathcal{D}) = n - 1 \).

The last step of the proof consists in choosing an \( n \)-compressible automaton \( \mathcal{D} \) of the form (1).

Let \( p \in Q_{\mathcal{D}} \) and \( v \in \Sigma^* \). We call the sequence of edges
\[
tr(p, v) = \{ (\delta_{\mathcal{D}}(p, v_i), v[i], \delta_{\mathcal{D}}(p, v_i)) \}_{i=1}^{\left| v \right|}
\]
the trace of the word \( v \) from the state \( p \).

**Lemma 12.** Suppose \( p \in Q_{\mathcal{D}} \), \( v \in \Sigma^* \) and \( \mathcal{D} \) is a DFA of the form (1). If \( tr(p, v) \subseteq IE(\mathcal{D}) \), then \( \delta_{\mathcal{D}}(p, v) = \delta_{\mathcal{D}}(p, v) \).

**Proof.** Since \( tr(p, v) \subseteq IE(\mathcal{D}) \), the path \( tr(p, v) \) contains no outgoing edges. Hence, the edges \( (\delta_{\mathcal{D}}(p, v_i), v[i]) \) and \( (\delta_{\mathcal{D}}(p, v_i), v[i]) \) coincide for every \( i, 1 \leq i \leq |v| \).
Proposition 13. There exists an n-compressible DFA \( \mathcal{D} = (Q, \Sigma, \delta, \gamma) \) of the form (1) such that \( df_w(\mathcal{D}) < n \) and \( |Q| \leq |M| + n + 1 \).

**Proof.** By Corollary 11, \( df_w(\mathcal{D}) = n - 1 \) for any automaton \( \mathcal{D} \) of the form (1). Our aim is to choose an automaton \( \mathcal{D} \) of the form (1), two different states \( G, H \in \delta(Q, \gamma) \) and a word \( Z \) such that \( \delta(Q, Z) = \delta(Q, H, Z) \). This means that the states \( G \) and \( H \) are covered by tokens after the action of \( w \) and the word \( Z \) removes one of the tokens. Hence the automaton \( \mathcal{D} \) is n-compressible. If we find \( G, H, Z \) and \( \mathcal{D} = (Q, \Sigma, \gamma) \), where \( |Q| \leq |M| + n + 1 \), then we complete the proof of the theorem.

Recall that \( df_w(\mathcal{D}) = n - 1 \) and the automaton \( \mathcal{D} \) is n-compressible. It means that there exists a word \( v \in \Sigma^* \) such that \( df_w(\mathcal{D}) \leq n \). Note that \( df_{vw}(\mathcal{D}) \geq n \). Then there are two different states \( p, r \in \delta(Q, \gamma) \) and a word \( u = a_1a_2 \cdots a_k \in \Sigma^* \) such that \( \delta(p, u) = \delta(p, r) \). Without loss of generality, we may assume that \( \delta(p, u_{k-1}) \neq \delta(p, u_{k-1}) \). Let \( q = \delta(p, u) \). Applying Proposition 4 to the states \( \delta(p, u_{k-1}) \) and \( \delta(p, u_{k-1}) \) and the letter \( a_k \) we obtain that \( q \in M \).

**Case 1.** Assume that \( p \notin M \) and \( r \notin M \).

Then \( e_1 = (p_1, a, p_2) \) and \( e_2 = (r_1, a, r_2) \) are different ingoing edges of the automaton \( \mathcal{D} \). Let \( f_1 = \psi_{\text{end}}(e_1) \) and \( f_2 = \psi_{\text{end}}(e_2) \) be the corresponding outer edges of the automaton \( \mathcal{D} \). Consider two identical buffer automata \( L_1(\lambda, a) \) and \( L_2(\lambda, a) \) of the empty word \( \lambda \) and the input–output letter \( a \). Let \( \mathcal{D} = \mathcal{C} \oplus L_1(\lambda, a) \oplus L_2(\lambda, a) \). Since the mapping \( \psi_{\text{end}} \) preserves the end points of edges, we have \( \mathcal{D}_1 = p_2 \) and \( \mathcal{D}_2 = r_2 \). Hence

\[
\delta_{\mathcal{D}}(KS(L_1), \lambda a v) = \delta_{\mathcal{D}}(out L_1, v) = \delta_{\mathcal{D}}(p_2, v) = \delta_{\mathcal{D}}(p_2, v) = q
\]

and

\[
\delta_{\mathcal{D}}(KS(L_2), \lambda a v) = \delta_{\mathcal{D}}(out L_2, v) = \delta_{\mathcal{D}}(r_2, v) = \delta_{\mathcal{D}}(r_2, v) = q.
\]

Since \( KS(L_1) \notin Q \) and \( KS(L_2) \notin Q \), we have \( KS(L_1) \in \delta(Q, \gamma) \) and \( KS(L_2) \in \delta(Q, \gamma) \). Note that \( |Q| = |M| + 2 \leq |M| + n + 1 \). Now we can put \( G = KS(L_1), H = KS(L_2) \) and \( Z = v \).

**Case 2.** Assume that exactly one of the states \( p_1 \) and \( r_1 \) does not belong to the set \( M \). Without loss of generality we suppose that \( p_1 \in M \) while \( r_1 \notin M \).

**Case 2a.** Assume that there exists a word \( x \in \Sigma^* \) and a state \( s \in \delta(Q, \gamma) \) such that \( tr(s, x) \subseteq IE(\mathcal{D}) \) and \( \delta_g(s, x) = p_1 \). We choose the pair \( (x, s) \) such that the word \( x \) is the shortest with this property. Then the path \( tr(s, x) \) visits each of its state only once. Furthermore, \( \delta_g(s, x_i) \notin \delta(Q, \gamma) \) for each \( i, 1 \leq i \leq |x| \).

Since \( |Q| \leq |M| + n \), we obtain \( |x| \leq n - 1 \). Note that the edge \( e = (r_1, a, r_2) \) is an ingoing edge of \( \mathcal{D} \). Let \( f = \psi_{\text{end}}(e) \) be the corresponding outer edge of \( \mathcal{D} \). We put \( \mathcal{D} = \mathcal{C} \oplus L(x, a) \). Note that \( |Q| = |M| + n \leq |M| + n + 1 \).

Since \( KS(L) \notin Q \), we have \( KS(L) \in \delta(Q, \gamma) \) by Lemma 10. Note that \( s \in Q \), because \( s \in M \). Since \( s \in \delta(Q, \gamma) \), we have

\[
s \notin \delta(Q, \gamma), \delta(Q, \gamma) = Q \setminus \delta(Q, \gamma), \delta(Q, \gamma), \]

whence \( s \in \delta(Q, \gamma) \).

By the definition of the state \( s \) we have

\[
\delta_{\mathcal{D}}(s, x a v) = \delta_{\mathcal{D}}(s, x a v) = \delta_{\mathcal{D}}(p_1, a v) = q.
\]

By the definition of buffer automata we obtain that

\[
\delta_{\mathcal{D}}(KS(L), x a v) = \delta_{\mathcal{D}}(out L, v) = \delta_{\mathcal{D}}(r_2, v) = \delta_{\mathcal{D}}(r_2, v) = q.
\]
Now we can put $G = KS(\mathcal{L})$, $H = s$ and $Z = xav$.

**Case 2b.** Assume that there is no word $x \in \Sigma^*$ and no state $s \in \delta_{\mathcal{B}}(Q_{\mathcal{B}}, w)$ such that $tr(s, x) \subseteq IE(\mathcal{B})$ and $\delta_{\mathcal{B}}(s, x) = p_1$.

Suppose that $tr(p, u_j) \subseteq IE(\mathcal{B})$. Then we have a pair $(u_j, p)$ such that $tr(p, u_j) \subseteq IE(\mathcal{B})$ and $\delta_{\mathcal{B}}(p, u_j) = p_1$. This contradicts the assumption of this case. Hence, $tr(p, u_j) \not\subseteq IE(\mathcal{B})$. This means that there is a triple $(b, x, s)$ such that $b \in \Sigma, x \in \Sigma^*, s \in Q_{\mathcal{B}} \setminus M$, $tr(\delta_{\mathcal{B}}(s, b), x) \subseteq IE(\mathcal{B})$ and $\delta_{\mathcal{B}}(s, bx) = p_1$. We fix a triple $(b, x, s)$ such that the word $x$ is the shortest with these properties. Let $t = \delta_{\mathcal{B}}(s, b)$. Then the path $tr(t, x)$ visits each of its state only once. Note that $\forall i, 0 \leq i \leq |x|$, $\delta_{\mathcal{B}}(t, x_i) \not\in \delta_{\mathcal{B}}(Q_{\mathcal{B}}, w)$. Otherwise, there is a number $i$ such that $\delta_{\mathcal{B}}(t, x_i) \in \delta_{\mathcal{B}}(Q_{\mathcal{B}}, w)$, whence the pair $(x[i+1]|x[i+2]..., \delta_{\mathcal{B}}(t, x_i))$ contradicts the assumption of this case.

Since $|Q_{\mathcal{B}} \setminus \delta_{\mathcal{B}}(Q_{\mathcal{B}}, w)| = n-1$, we obtain $|x| \leq n-2$.

Note that the edges $e_1 = (s, b, t)$ and $e_2 = (r_1, a, r_2)$ are ingoing edges of the automaton $\mathcal{B}$.

**Subcase 2b1.** Assume that $e_1 \not\in e_2$.

Let $f_1 = \psi_{end}(e_1)$ and $f_2 = \psi_{end}(e_2)$ be the corresponding outer edges of the automaton $\mathcal{C}$. Consider the buffer automata $\mathcal{L}_1(\lambda, b)$ and $\mathcal{L}_2(bx, a)$. We put $\mathcal{D} = \mathcal{C} \oplus f_1 \mathcal{L}_1 \oplus f_2 \mathcal{L}_2$. Then $|Q_{\mathcal{D}}| \leq |M| + n + 1$.

Since $KS(\mathcal{L}_1) \not\in Q_{\mathcal{B}}$ and $KS(\mathcal{L}_2) \not\in Q_{\mathcal{B}}$, we have $KS(\mathcal{L}_1) \in \delta_{\mathcal{B}}(Q_{\mathcal{B}}, w)$ and $KS(\mathcal{L}_2) \in \delta_{\mathcal{B}}(Q_{\mathcal{B}}, w)$ by Lemma 10.

By the definition of buffer automata we obtain that
$$\delta_{\mathcal{D}}(KS(\mathcal{L}_1), bxav) = \delta_{\mathcal{D}}(out\mathcal{L}_1, xav) = \delta_{\mathcal{D}}(t, xav) = 12 \delta_{\mathcal{D}}(t, xav).$$

By the choice of the states $s$ and $t$ we have
$$\delta_{\mathcal{B}}(t, xav) = \delta_{\mathcal{B}}(s, bxav) = \delta_{\mathcal{B}}(p_1, av) = q.$$ 

By the definition of buffer automata we obtain that
$$\delta_{\mathcal{D}}(KS(\mathcal{L}_2), bxav) = \delta_{\mathcal{D}}(out\mathcal{L}_2, v) = \delta_{\mathcal{D}}(r_2, v) = 12 \delta_{\mathcal{D}}(r_2, v) = q.$$ 

Now we can put $G = KS(\mathcal{L}_1), H = KS(\mathcal{L}_2)$ and $Z = bxav$.

**Subcase 2b2.** Assume that $e_1 = e_2$, i.e. $s = r_1, t = r_2, b = a$.

Let $f = \psi_{end}(e_1)$ be the corresponding outer edge of the automaton $\mathcal{C}$. Consider the buffer automaton $\mathcal{L}(ax, a)$.

We put $\mathcal{D} = \mathcal{C} \oplus f \mathcal{L}$. Then $|Q_{\mathcal{D}}| \leq |M| + n \leq |M| + n + 1$.

Let $o = \delta_{\mathcal{D}}(KS(\mathcal{L}), ax)$. Since $KS(\mathcal{L}) \not\in Q_{\mathcal{B}}$ and $o \not\in Q_{\mathcal{B}}$, we have $KS(\mathcal{L}) \in \delta_{\mathcal{D}}(Q_{\mathcal{B}}, w)$ and $o \in \delta_{\mathcal{D}}(Q_{\mathcal{B}}, w)$ by Lemma 10.

By the definition of buffer automata we obtain that
$$\delta_{\mathcal{D}}(o, axav) = \delta_{\mathcal{D}}(out\mathcal{L}, xav) = \delta_{\mathcal{D}}(t, xav) = 12 \delta_{\mathcal{D}}(t, xav).$$

By the choice of the states $s$ and $t$ we have
$$\delta_{\mathcal{B}}(t, xav) = \delta_{\mathcal{B}}(s, axav) = \delta_{\mathcal{B}}(p_1, av) = q.$$ 

By the definition of a buffer automaton we obtain that
$$\delta_{\mathcal{D}}(KS(\mathcal{L}), axav) = \delta_{\mathcal{D}}(out\mathcal{L}, v) = \delta_{\mathcal{D}}(r_2, v) = 12 \delta_{\mathcal{D}}(r_2, v) = q.$$ 

Hence, $\delta_{\mathcal{D}}(o, axav) = \delta_{\mathcal{D}}(KS(\mathcal{L}), axav)$. Thus, we can put $G = KS(\mathcal{L}), H = o$ and $Z = axav$. □

By combining Lemma 2, Propositions 8 and 13, we obtain the proof of Theorem 1.
Acknowledgement

This work was supported by the Russian Foundation for Basic Research, grant 05-01-00540.

References