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 \( \mathcal{A} = (Q, \Sigma, \delta) \), 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 being \( n \)-collapsing is algorithmically recognizable for any given positive integer \( n \). We also prove that the language of all \( n \)-collapsing words is context-sensitive.

Key words: Deterministic finite automaton, \( n \)-collapsing word, Context-sensitive language

1 Main result and its application

Let \( \mathcal{A} = (Q_\mathcal{A}, \Sigma, \delta_\mathcal{A}) \) 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 \( df_w(\mathcal{A}) = |Q_\mathcal{A}| - |\delta_\mathcal{A}(Q_\mathcal{A}, w)| \) is called the deficiency of the action of \( w \) on the automaton \( \mathcal{A} \).

Let \( n \) be a positive integer. A DFA \( \mathcal{A} = (Q_\mathcal{A}, \Sigma, \delta_\mathcal{A}) \) is said to be \( n \)-compressible if there is a word \( w \in \Sigma^* \) such that \( df_w(\mathcal{A}) \geq n \). The word \( w \) is then called

---

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

Preprint submitted to Elsevier Science 9 May 2007
compressing with respect to $A$. 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 terms, a word $w \in \Sigma^*$ is $n$-collapsing if for any DFA $\mathcal{A} = \langle Q, \Sigma, \delta, \omega \rangle$ we have $df_w(\mathcal{A}) \geq n$ 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 if a given word is $n$-collapsing. This problem is non-trivial whenever $n > 1$ and $|\Sigma| > 1$ that 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 $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$ with $|Q| < |w|$ and $df_w(\mathcal{A}) = 1$ (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 $\mathcal{C}_n(\Sigma)$ of all $n$-collapsing words over $\Sigma$ is decidable in principle, i.e. $\mathcal{C}_n(\Sigma)$ 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 $df_w(\mathcal{A}) \geq n$ for every $n$-compressible DFA $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$ with $|Q| \leq f_n(|w|)$. 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 $df_w(\mathcal{A}) < n$, 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 \( \mathcal{A} = \langle Q_{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}} \rangle \) with \( |Q_{\mathcal{A}}| \leq 2|w|(n - 1) + 2 \) such that \( df_w(\mathcal{A}) < n \).

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 \( \mathcal{A} = \langle Q_{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}} \rangle \) with \( |Q_{\mathcal{A}}| \leq 2|w|(n - 1) + 2 \) and then verifies that \( \mathcal{A} \) is \( n \)-compressible and that \( w \) is not \( n \)-compressing 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 \( \mathcal{A} = \langle Q_{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}} \rangle \) as a digraph with the vertex set \( Q_{\mathcal{A}} \). We denote by \((p, q, a)\) the edge from \( p \in Q_{\mathcal{A}} \) to \( q \in Q_{\mathcal{A}} \) labeled by the letter \( a \in \Sigma \). We shall identify the transition function \( \delta_{\mathcal{A}} : Q_{\mathcal{A}} \times \Sigma \to Q_{\mathcal{A}} \) with its graph \( \{(v, \delta_{\mathcal{A}}(v, a)) \mid v \in Q_{\mathcal{A}}, a \in \Sigma\} \); that is, the expressions \((p, q, a) \in \delta_{\mathcal{A}}\) and \( \delta_{\mathcal{A}}(p, a) = q \) mean the same. We denote the set \( \{(v, \delta_{\mathcal{A}}(v, a)) \mid v \in Q_{\mathcal{A}}\} \) by \( \delta_{\mathcal{A}}(\bullet, a) \), i.e. \( \delta_{\mathcal{A}}(\bullet, a) \) is the set of all edges labeled by \( a \).

We need some notation 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 \( u = a_1a_2\ldots a_t \) 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 \( u = v'vv'' \) 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 \( w \) is \( n \)-full.

We say that an \( n \)-compressible 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 \( \mathcal{A} = \langle Q_{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}} \rangle \) such that \( |Q_{\mathcal{A}}| \leq |w| \) and \( df_w(\mathcal{A}) < n \).

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

**Lemma 3** The DFA \( \mathcal{B} \) is an \( n \)-proper and \( n \)-compressible automaton, and \( \text{df}_w(\mathcal{B}) = n - 1 \).

Now assume that some of the states of the DFA \( \mathcal{B} \) are covered by tokens and the action of any letter \( a \in \Sigma \) redistributes the tokens according to the following rule: a state \( q \in Q_{\mathcal{B}} \) will be covered by a token after the action of \( a \) if and only if there exists a state \( q' \in Q_{\mathcal{B}} \) such that \( \delta(q', a) = q \) 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 \( a \) and, whenever several tokens arrive at the same state, all but one of them are removed. 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.

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

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 $w$.

![Diagram](image)

Fig. 2. Marking induced by the transition shown in Fig. 1

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 $\delta(3, a) = 4$. Further, $M(1, k) = \{2, 5\}$.

Put $M = \bigcup_{1 \leq k \leq t} (M(1, k) \cup M(2, k))$. We call $M$ the set of marked states of the DFA $\mathcal{B}$ or the marked set for short. The next proposition registers an important property of the marked set.

**Proposition 4** Let $a \in \Sigma$, $p, r \in Q, a \neq r$. If $\delta(p, a) = \delta(r, a)$, then $\delta(a, a) \in M$.

**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 $\mathcal{B}$ is $n$-proper whence $df_n(\mathcal{B}) \leq n - 1$. Therefore the non-increasing chain $Q \supseteq \delta(Q, a) \supseteq \delta(Q, a^2) \supseteq \ldots$ stabilizes after at most $n - 1$ steps whence $a$ acts as a permutation on the set $\delta(Q, a^n-1)$ as a permutation.

Let $q = \delta(p, a) = \delta(r, a)$. If $q \notin \delta(Q, a^n-1)$ then

$$q \in Q \setminus \delta(Q, a^n-1) \subseteq Q \setminus \delta(Q, w_i a^n-1) = M(1, i + n - 1) \subseteq M.$$ 

Now assume $q \in \delta(Q, a^n-1)$. The states $p$ and $r$ cannot simultaneously belong to $\delta(Q, a^n-1)$ because $a$ acts as a permutation on this set while $\delta(p, a) = \delta(r, a)$. Without loss of generality, assume that $p \notin \delta(Q, a^n-1)$. Then

$$p \in Q \setminus \delta(Q, w_i a^n-1) \subseteq M(1, i + n - 1).$$

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

An edge $e = (q_1, q_2, a)$ 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 \( \bar{M}(\mathcal{B}, a) \) 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 \( \bar{M}(\mathcal{B}, a) \) 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 \( e = (q_1, q_2, w[k]) \) 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 \( e = (q_1, q_2, w[k]) \) is inner, a contradiction.

**Lemma 6** After the action of the prefix \( w_{k-1} \), the initial vertex of every ingoing edge \( e = (q_1, q_2, w[k]) \) 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 gets empty.

**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 \( \mathcal{M} = Q_{\mathcal{B}} \setminus M \) 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 \( \mathcal{M} \) are covered by tokens. Therefore the number of tokens in \( \mathcal{M} \) is equal to \( |\mathcal{M}| \) 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 \( \mathcal{M} \). The number of tokens leaving the set \( \mathcal{M} \) is equal to \( I_a \) by Lemma 6. The number of tokens coming to the set \( \mathcal{M} \) is equal to \( O_a \) by Lemma 5. Any token in \( \mathcal{M} \) which is removed after the action leaves the set \( \mathcal{M} \). Indeed, it moves along the edge \( (p, q, w[i]) \) which shares its end point \( q \) with another edge \( (r, q, w[i]) \). The state \( q \) is not in \( \mathcal{M} \) by Proposition 4.

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

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

We complete the automaton $C$ 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 $e \in IE(B)$ of the automaton $B$.

Now consider a letter $a \in \Sigma$. By Proposition 7 there is a bijection

$$\varphi_a : \tilde{M}(B, a) \rightarrow \tilde{M}(B, a).$$

We fix such a bijection $\varphi_a$ and do the following for each outgoing edge $e = (q_1, q_2, a)$. Let $\varphi_a(e) = (q_3, q_4, a)$. It is an ingoing edge. We append a new edge $f = (q_1, q_4, a)$ to the automaton $C$, 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 $C$. 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 $C$. This should not lead to any confusion since from now on we shall use the completed version of $C$ only.

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

$$IE(C, a) = \{e = (q_1, q_2, a) \mid e \text{ is an inner edge in } C\}, \quad IE(C) = \bigcup_{a \in \Sigma} IE(C, a),$$

$$\tilde{M}(C, a) = \{e = (q_1, q_2, a) \mid e \text{ is an outer edge in } C\}, \quad \tilde{M}(C) = \bigcup_{a \in \Sigma} \tilde{M}(C, a).$$

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

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

Observe that the mappings

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

and

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

are well-defined.
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** $|Q_\mathcal{E}| = |M| \leq (2\ell - 1)(n - 1)$.

**Proof.** Since $df_w(\mathcal{B}) = n - 1$, 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 $|Q_\mathcal{E} \setminus \delta_\mathcal{E}(Q_\mathcal{E}, w_k)| \leq n - 1$ 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 $s = a_1a_2 \cdots a_t \in \Sigma^*$ 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 $Q_\mathcal{L} = \{b_1, b_2, \ldots, b_{t+1}\}$ and whose edges are

$$(b_1, b_2, a_1), (b_2, b_3, a_2), \ldots, (b_t, b_{t+1}, a_t), (b_{t+1}, b_1, a).$$

After that we complete the automaton to a permutation automaton over $\Sigma$ in an arbitrary way. Finally, we remove the edge $(b_{t+1}, b_1, a)$. 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 $KS(\mathcal{L})$.

![Fig. 3. Building a buffer automaton](image-url)

It is convenient to imagine that instead of the removed edge $(b_{t+1}, b_1, a)$, the automaton $\mathcal{L}(s, a)$ has got two ‘open’ edges $(\bullet, b_1, a)$ and $(b_{t+1}, \bullet, a)$ with undefined starting and end points respectively. We shall use such undefined starting and end points to attach buffer automata to the automaton $\mathcal{C}$.

Let $e = (q_1, q_2, a)$ be an outer edge of the automaton $\mathcal{C}$ and let $\mathcal{L}(s, a) = (Q_\mathcal{L}, \Sigma, \delta_\mathcal{L})$ be an arbitrary buffer automaton whose input-output letter is $a$. 

8
Then we define the operation $C \circ L(s, a)$ of attaching the buffer automaton $L(s, a)$ to the DFA $C$ instead of the edge $e$. The result of this operation is a new automaton $T = (Q_T, \Sigma, \delta_T)$ defined as follows:

$$Q_T = Q_C \cup Q_L = Q_C \cup \{b_1, b_2, \ldots, b_{t+1}\}$$

$$\delta_T(q, c) = \begin{cases} 
\delta_C(q, c), & \text{if } q \in Q_C \setminus \{q_1\} \\
\delta_C(q, c), & \text{if } q = q_1, c \neq a \\
b_1, & \text{if } q = q_1, c = a \\
\delta_L(q, c), & \text{if } q \in \{b_1, b_2, \ldots, b_t\} \\
\delta_L(q, c), & \text{if } q = b_{t+1}, c \neq a \\
q_2, & \text{if } q = b_{t+1}, c = a 
\end{cases}$$

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

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

$$D = (Q_D, \Sigma, \delta_D) = C \circ L_1(x_1, a_1) \circ \cdots \circ L_r(x_r, a_r). \quad (1)$$

The automaton $D$ 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 $D$ 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 $D$ of the form (1) satisfies the condition

$$Q_D \setminus \delta_D(Q_D, w_k) = Q_D \setminus \delta_D(Q_D, w_k)$$

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

**Proof.** Let $M = Q_D \setminus M$. Let $L = \bigcup_{i=1}^r Q_L$, where $Q_L$ is the state set of the buffer automaton $L_i$. Then the state sets of the automata $M$ and $D$ can be
Suppose that \( D_k = Q_\mathcal{D} \setminus \delta_\mathcal{D}(Q_\mathcal{D}, w_k) \), \( B_k = Q_\mathcal{D} \setminus \delta_\mathcal{D}(Q_\mathcal{D}, w_k) \) be the sets of empty states of the automata \( \mathcal{D} \) and respectively \( \mathcal{B} \) 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

\[
q \in (B_k \setminus D_k) \cup (D_k \setminus B_k) \subseteq B_k \cup D_k \subseteq M \cup M \cup L.
\]

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

\[
D_{k-1} = B_{k-1} = Q_\mathcal{D} \setminus \delta_\mathcal{D}(Q_\mathcal{D}, w_{k-1}) = M(1, k - 1) \subseteq M.
\]

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 \( Q_{\mathcal{L}_i} \cap M = \emptyset \).

Let \( w_k = w_{k-1}a \). Consider an arbitrary buffer automaton \( \mathcal{L}_i(x_i, a) \) attached instead of the edge \( e_i \). If \( e_i = (p_i, q_i, b) \), \( b \neq a \) 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 \( Q_{\mathcal{L}_i} \) by the definition of buffer automata.

If \( e_i = (p_i, q_i, a) \), then there is an outgoing edge \((p_i, r_i, a) = \psi^{-1}_\text{start}(e_i) \) of \( \mathcal{B} \) corresponding to \( e_i \). We have \( p_i \in \delta_\mathcal{D}(Q_\mathcal{D}, w_{k-1}) \) by Lemma 5. Hence \( p_i \) is covered by a token in \( \mathcal{D} \) after the action of \( w_{k-1} \). Therefore the transformation \( \delta_\mathcal{D}(\bullet, a) \) pushes the token into \( Q_{\mathcal{L}_i} \). There is only one edge \( f_i = (s_i, q_i, a) = \psi^{-1}_\text{end}(e_i) \) outgoing from the set \( Q_{\mathcal{L}_i} \) in \( \mathcal{D} \) and there is no pair of edges labeled by \( a \) with a common end point in \( \mathcal{L}_i \). It means that the number of tokens in \( Q_{\mathcal{L}_i} \) is not decreasing during the action of \( a \) and the transformation \( \delta_\mathcal{D}(\bullet, a) \) pops the token from \( Q_{\mathcal{L}_i} \) via the edge \( f_i \). Whence \( \mathcal{L}_i \) is covered after the action of \( w_k \) and the state \( q_i \) is also covered. That is \( Q_{\mathcal{L}_i} \subseteq \delta_\mathcal{D}(Q_\mathcal{D}, w_k) \), \( q_i \in \delta_\mathcal{D}(Q_\mathcal{D}, w_k) \).

Therefore, \( q \notin L \).

**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 $f = (p, q, a)$ of the automaton $\mathcal{B}$.

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

This means that there is no outer edge labeled by $a$ with the end point $q$ in $\mathcal{C}$. Therefore any edge $e = (p, q, a)$ 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 $e = (p, q, a)$ 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 $e = (p, q, a)$ 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), $df_w(\mathcal{D}) = df_w(\mathcal{B}) = n - 1$.

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

Let $p \in Q_{\mathcal{B}}$ and $v \in \Sigma^*$. We call the sequence of edges

$$tr(p, v) = \{\delta_{\mathcal{B}}(p, v_{i-1}), \delta_{\mathcal{B}}(p, v_i), v[i]\}_{i=1}^{v}$$

the trace of the word $v$ from the state $p$.

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

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

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 $G, H \in \delta_\mathcal{D}(Q_\mathcal{D}, w)$ and a word $Z$ such that $\delta_\mathcal{D}(G, Z) = \delta_\mathcal{D}(H, Z)$. 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 $\mathcal{D} = (Q_\mathcal{D}, \Sigma, \delta_\mathcal{D})$, where $|Q_\mathcal{D}| \leq |M| + n + 1$, 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 $df_w(\mathcal{D}) \geq n$. Note that $df_w(\mathcal{D}) \geq n$. Then there are two different states $p, r \in \delta_\mathcal{D}(Q_\mathcal{D}, w)$ ($p \neq r$) and a word $u = a_1a_2\ldots a_k \in \Sigma^*$ such that $\delta_\mathcal{D}(p, u) = \delta_\mathcal{D}(r, u)$. Without loss of generality, we may assume that $\delta_\mathcal{D}(p, u_{k-1}) \neq \delta_\mathcal{D}(r, u_{k-1})$. Let $q = \delta_\mathcal{D}(p, u)$. Applying Proposition 4 to the states $\delta_\mathcal{D}(p, u_{k-1})$ and $\delta_\mathcal{D}(r, u_{k-1})$ and the letter $a_k$ we obtain that $q \in M$.

Let $N = \{ j | 0 \leq j \leq |u|, \delta_\mathcal{D}(p, u_j) \notin M \text{ or } \delta_\mathcal{D}(r, u_j) \notin M \}$. If $N = \emptyset$, i.e. $tr(p, u) \subseteq IE(\mathcal{B})$ and $tr(r, u) \subseteq IE(\mathcal{B})$. Then by Lemma 12 $\delta_\mathcal{D}(r, u) = \delta_\mathcal{D}(r, u) = \delta_\mathcal{D}(p, u)$. This means that we can put $\mathcal{D} = \mathcal{C}$, $G = p$, $H = r$ and $Z = u$. Note that $|Q_\mathcal{D}| = |M| \leq |M| + n + 1$.

Suppose that $N \neq \emptyset$. We define $j = \max N$. Note that $j < |u|$ because $\delta_\mathcal{D}(p, u) = \delta_\mathcal{D}(r, u) \in M$. Let $p_1 = \delta_\mathcal{D}(p, u_j)$, $p_2 = \delta_\mathcal{D}(p, u_{j+1})$, $r_1 = \delta_\mathcal{D}(r, u_j)$, $r_2 = \delta_\mathcal{D}(r, u_{j+1})$, $a = u[j + 1]$. We denote the word $a_1a_2\ldots a_k$ by $v$, whence $u = u_jav$.

Case 1. Assume that $p_1 \notin M$ and $r_1 \notin M$.

Then $e_1 = (p_1, p_2, a)$ and $e_2 = (r_1, r_2, a)$ 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 outgoing edges of the automaton $\mathcal{C}$. Consider two identical buffer automata $\mathcal{L}_1(\lambda, a)$ and $\mathcal{L}_2(\lambda, a)$ of the empty word $\lambda$ with the input-output letter $a$. Let $\mathcal{D} = \mathcal{C} \uplus \mathcal{L}_1(\lambda, a) \uplus \mathcal{L}_2(\lambda, a)$. Since the mapping $\psi_{\text{end}}$ preserves the end points of edges, we have $out\mathcal{L}_1 = p_2$ and $out\mathcal{L}_2 = r_2$. Whence

$$\delta_\mathcal{D}(KS(\mathcal{L}_1), \lambda v) = \delta_\mathcal{D}(out\mathcal{L}_1, v) = \delta_\mathcal{D}(p_2, v) \overset{\text{Lemma 12}}{=} \delta_\mathcal{D}(p_2, v) = q$$

and

$$\delta_\mathcal{D}(KS(\mathcal{L}_2), \lambda v) = \delta_\mathcal{D}(out\mathcal{L}_2, v) = \delta_\mathcal{D}(r_2, v) \overset{\text{Lemma 12}}{=} \delta_\mathcal{D}(r_2, v) = q.$$
Now we can put $G = KS(\mathcal{L}_1)$, $H = KS(\mathcal{L}_2)$ 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 $s \in \delta_B(Q_B, w)$ such that $tr(s, x) \subseteq IE(B)$ and $\delta_B(s, x) = p_1$. 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, $\delta_B(s, x_i) \notin \delta_B(Q_B, w)$ for each $i, 1 \leq i \leq |x|$.

Since $|Q_B \setminus \delta_B(Q_B, w)| = n - 1$, we obtain $|x| \leq n - 1$. Note that the edge $e = (r_1, r_2, a)$ is an ingoing edge of $B$. Let $f = \psi_{\text{end}}(e)$ be the corresponding outer edge of $C$. We put $\mathcal{G} = C \upharpoonright \mathcal{L}(x, f)$. Note that $|Q_\mathcal{G}| \leq |M| + n \leq |M| + n + 1$.

Since $KS(\mathcal{L}) \notin Q_B$, we have $KS(\mathcal{L}) \in \delta_B(Q_B, w)$ by Lemma 10. Note that $s \in Q_B$, because $s \in M$. Since $s \in \delta_B(Q_B, w)$, we have

$$s \notin Q_B \setminus \delta_B(Q_B, w) \overset{\text{Lemma 10}}{=} 10 Q_B \setminus \delta_B(Q_B, w),$$

whence $s \in \delta_B(Q_B, w)$.

By the definition of the state $s$ we have

$$\delta_B(s, xav) \overset{\text{Lemma 12}}{=} 12 \delta_B(s, xav) = \delta_B(p_1, av) = q.$$

By the definition of buffer automata we obtain that

$$\delta_B(KS(\mathcal{L}), xav) = \delta_B(out.\mathcal{L}, v) = \delta_B(r_2, v) \overset{\text{Lemma 12}}{=} 12 \delta_B(r_2, v) = q.$$

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 $s \in \delta_B(Q_B, w)$ such that $tr(s, x) \subseteq IE(B)$ and $\delta_B(s, x) = p_1$.

Suppose that $tr(p, u_j) \subseteq IE(B)$. Then we have a pair $(u_j, p)$ such that $tr(p, u_j) \subseteq IE(B)$ and $\delta_B(p, u_j) = p_1$. This contradicts the assumption of this case. Hence, $tr(p, u_j) \not\subseteq IE(B)$. This means that there is a triple $(b, x, s)$ such that $b \in \Sigma$, $x \in \Sigma^*$, $s \in Q_B \setminus M$, $tr(\delta_B(s, b), x) \subseteq IE(B)$ and $\delta_B(s, bx) = p_1$. We fix a triple $(b, x, s)$ such that the word $x$ is the shortest with these properties. Let $t = \delta_B(s, b)$. Then the path $tr(t, x)$ visits each of its state only once.

Note that $\forall i$, $0 \leq i \leq |x|$, $\delta_B(t, x_i) \notin \delta_B(Q_B, w)$. Otherwise, there is a number $i$ such that $\delta_B(t, x_i) \in \delta_B(Q_B, w)$ whence the pair $(x[i+1]x[i+2] \ldots, \delta_B(t, x_i))$
contradicts the assumption of this case.

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

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

**Subcase 2b1.** Assume that $e_1 \neq e_2$.

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

Since $KS(\mathcal{L}_1) \notin Q_\mathcal{G}$ and $KS(\mathcal{L}_2) \notin Q_\mathcal{G}$, we have $KS(\mathcal{L}_1) \in \delta_\mathcal{G}(Q_\mathcal{G}, w)$ and $KS(\mathcal{L}_2) \in \delta_\mathcal{G}(Q_\mathcal{G}, w)$ by Lemma 10.

By the definition of buffer automata we obtain that

$$\delta_\mathcal{D}(KS(\mathcal{L}_1), bxav) = \delta_\mathcal{D}(\text{out}, bxav) = \delta_\mathcal{D}(q, bxav) = \delta_\mathcal{D}(t, xav) \overset{\text{Lemma 12}}{=} \delta_\mathcal{D}(t, xav).$$

By the choice of the states $s$ and $t$ we have

$$\delta_\mathcal{D}(t, xav) = \delta_\mathcal{D}(s, bxav) = \delta_\mathcal{D}(p_1, av) = q.$$

By the definition of buffer automata we obtain that

$$\delta_\mathcal{D}(KS(\mathcal{L}_2), bxav) = \delta_\mathcal{D}(\text{out}, bxav) = \delta_\mathcal{D}(r_2, v) \overset{\text{Lemma 12}}{=} \delta_\mathcal{D}(r_2, v) = q.$$

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 $f = \psi_{\text{end}}(e_1)$ be the corresponding outer edge of the automaton $\mathcal{C}$. Consider the buffer automaton $\mathcal{L}(ax, a)$. We put $\mathcal{D} = \mathcal{C} \oplus \mathcal{L}$. Then $|Q_\mathcal{G}| \leq |M| + n \leq |M| + n + 1$.

Let $o = \delta_\mathcal{G}(KS(\mathcal{L}), ax)$. Since $KS(\mathcal{L}) \notin Q_\mathcal{G}$ and $o \notin Q_\mathcal{G}$, we have $KS(\mathcal{L}) \in \delta_\mathcal{G}(Q_\mathcal{G}, w)$ and $o \in \delta_\mathcal{G}(Q_\mathcal{G}, w)$ by Lemma 10.

By the definition of buffer automata we obtain that

$$\delta_\mathcal{D}(o, axav) = \delta_\mathcal{D}(\text{out}, axav) = \delta_\mathcal{D}(q, bxav) \overset{\text{Lemma 12}}{=} \delta_\mathcal{D}(t, xav).$$
By the choice of the states $s$ and $t$ we have
\[
\delta_\phi(t, xav) = \delta_\phi(s, axav) = \delta_\phi(p_1, av) = q.
\]

By the definition of a buffer automaton we obtain that
\[
\delta_\phi(KS(L), axav) = \delta_\phi(\text{out}, L, v) = \delta_\phi(r_2, v) \overset{\text{Lemma 12}}{=} \delta_\phi(r_2, v) = q.
\]

Hence, $\delta_\phi(o, axav) = \delta_\phi(KS(L), axav)$. Thus, we can put $G = KS(L)$, $H = o$ and $Z = axav$.

Combining Lemma 2, Proposition 8 and Proposition 13 we obtain the proof of Theorem 1.

References


