Available online at www.sciencedirect.com

SCIENCE DIRECT®

Theoretical Computer Science 327 (2004) 375–390

Theoretical Computer Science

www.elsevier.com/locate/tcs

Minimizing finite automata is computationally hard

Andreas Malcher

Institut für Informatik, Johann Wolfgang Goethe Universität, D-60054 Frankfurt am Main, Germany

Received 17 September 2003; received in revised form 19 March 2004; accepted 19 March 2004

Abstract

It is known that deterministic finite automata (DFAs) can be algorithmically minimized, i.e., a DFA M can be converted to an equivalent DFA M' which has a minimal number of states. The minimization can be done efficiently (in: Z. Kohavi (Ed.), Theory of Machines and Computations, Academic Press, New York, 1971, pp. 189–196). On the other hand, it is known that unambiguous finite automata and nondeterministic finite automata can be algorithmically minimized too, but their minimization problems turn out to be NP-complete and PSPACE-complete, respectively (SIAM J. Comput. 22(6) (1993) 1117–1141). In this paper, the time complexity of the minimization problem for two restricted types of finite automata is investigated. These automata are nearly deterministic, since they only allow a small amount of nondeterminism to be used. The main result is that the minimization problems for these models are computationally hard, namely NP-complete. Hence, even the slightest extension of the deterministic model towards a nondeterministic one, e.g., allowing at most one nondeterministic move in every accepting computation or allowing two initial states instead of one, results in computationally intractable minimization problems.

Keywords: Finite automata; Limited nondeterminism; Minimization; NP-complete

1. Introduction

Finite automata are a well-investigated concept in theoretical computer science with a wide range of applications such as lexical analysis, pattern matching, or protocol specification in distributed systems. Owing to time and space constraints it is often very useful to provide minimal or at least succinct descriptions of such automata. Deterministic finite automata (DFAs) and their corresponding language class, the set of regular languages,

E-mail address: malcher@psc.informatik.uni-frankfurt.de (A. Malcher).

0304-3975/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2004.03.070

possess many nice properties such as, for example, closure under many language operations and many decidable questions. In addition, most of the decidability questions for DFAs, such as membership, emptiness, or equivalence, are efficiently solvable (cf. Section 5.2 in [16]). Furthermore, in [6] a minimization algorithm for DFAs is provided working in time $O(n \log n)$, where n denotes the number of states of the given DFA.

It is known that both nondeterministic finite automata (NFAs) and DFAs accept the set of regular languages, but NFAs can achieve exponentially savings in size when compared to DFAs [14]. Unfortunately, certain decidability questions, which are solvable in polynomial time for DFAs, are computationally hard for NFAs such as equivalence, inclusion, or universality [15,16]. Furthermore, the minimization of NFAs is proven to be PSPACE-complete in [8]. In the latter paper, it is additionally shown that unambiguous finite automata (UFAs) have an NP-complete minimization problem.

Therefore, we can summarize that determinism permits efficient solutions whereas the use of nondeterminism often makes solutions computationally intractable. Thus, one might ask what amount of nondeterminism is necessary to make things computationally hard, or, in other words, what amount of nondeterminism may be allowed so that efficiency is preserved.

Measures of nondeterminism in finite automata were first considered in [12] and [4] where the relation between the amount of nondeterminism of an NFA and the succinctness of its description is studied. Here, we look at computational complexity aspects of NFAs with a fixed finite amount of nondeterminism. In particular, these NFAs are restricted such that within every accepting computation at most a fixed number of nondeterministic moves is allowed to be chosen. It is easily observed that certain decidability questions then become solvable in polynomial time in contrast to arbitrary NFAs. However, the minimization problem for such NFAs is proven to be NP-complete.

We further investigate a model where the nondeterminism used is not only restricted to a fixed finite number of nondeterministic moves, but is additionally cut down such that only the first move is allowed to be a nondeterministic one. Hence, we come to DFAs with multiple initial states (MDFAs) which were introduced in [2] and recently studied in [11,5]. The authors of the latter paper examine the minimization problem for MDFAs and prove its PSPACE-completeness. Their proof is a reduction from the finite state automata intersection problem [1] which states that it is PSPACE-complete to answer the question whether there is a string $x \in \Sigma^*$ accepted by each A_i, where DFAs $A_1, A_2, \ldots, A_n$ are given. As is remarked in [1], the problem becomes solvable in polynomial time when the number of DFAs is fixed. We would like to point out that the number of initial states is not part of the instance of the minimization problem for MDFAs discussed in [5]. Thus, one might ask whether minimization of MDFAs with a fixed number of initial states is possible in polynomial time. We will show in Section 3 that the minimization problem of such MDFAs is NP-complete even if only two initial states are given. In analogy to NFAs with fixed finite branching, certain decidability questions can be shown to be efficiently solvable.

The paper is organized as follows. In the next section, we will provide and introduce the necessary definitions and notations. Section 3 contains the proof that it is NP-complete to minimize MDFAs with a fixed number of initial states. In Section 4, some details of this proof will be useful to prove the NP-completeness of the minimization problem for NFAs with a fixed finite amount of nondeterminism. A short summary and some open problems conclude the paper.

2. Preliminaries and definitions

Let Σ^* denote the set of all strings over the finite alphabet Σ, ε the empty string, and $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$. By $|w|$ we denote the length of a string w and by $|S|$ the cardinality of a set S. We assume that the reader is familiar with the common notions of formal language theory as presented in [7] as well as with the common notions of computational complexity theory that can be found in [1]. Let L be a regular set; then $\text{size}(L)$ denotes the number of states of the minimal DFA accepting L. We say that two finite automata are equivalent if both accept the same language. The size of an automaton M, denoted by $|M|$, is defined to be the number of states. A state of a finite automaton will be called *trap state* when no accepting state can be obtained from that state on every input.

Concerning the definitions of NFAs with finite branching and MDFAs we follow the notations introduced in [4,11].

A nondeterministic finite automaton over Σ is a tuple $M = (Q, \Sigma, \delta, q_0, F)$, with Q a finite set of states, $q_0 \in Q$ the initial state, $F \subseteq Q$ the set of accepting states, and δ a function from $Q \times \Sigma$ to 2^Q. A move of M is a triple $\mu = (p, a, q) \in Q \times \Sigma \times Q$ with $q \in \delta(p, a)$. A computation for $w = w_1 w_2 \ldots w_n \in \Sigma^*$ is a sequence of moves $\mu_1 \mu_2 \ldots \mu_n$ where $\mu_i = (p_{i-1}, w_i, p_i)$ with $p_0 = q_0$ and $p_i \in Q$ for $1 \leqslant i \leqslant n$. It is an accepting computation if $p_n \in F$. The language accepted by M is $T(M) = \{w \in \Sigma^* | \delta(q_0, w) \cap F \neq \emptyset\}$. M is an (incomplete) deterministic finite automaton if $|\delta(q, a)| \leqslant 1$ for all pairs (q, a). The branching $\beta_M(\mu)$ of a move $\mu = (q, a, p)$ is defined to be $\beta_M(\mu) = |\delta(q, a)|$. The branching is extended to computations $\mu_1 \mu_2 \ldots \mu_n, n \geqslant 0$, by setting $\beta_M(\mu_1 \mu_2 \ldots \mu_n) = \beta_M(\mu_1) \cdot \beta_M(\mu_2) \cdot \ldots \cdot \beta_M(\mu_n)$. For each word $w \in T(M)$, let $\beta_M(w) = \min \beta_M(\mu_1 \mu_2 \ldots \mu_n)$ where $\mu_1 \mu_2 \ldots \mu_n$ ranges over all accepting computations of M with input w. The branching β_M of the automaton M is $\beta_M = \sup \{\beta_M(w) | w \in T(M)\}$. The set of all NFAs with branching $\beta = k$ is defined as $\text{NFA}(\beta = k) = \{M | M \text{ is NFA and } \beta_M = k\}$.

A DFA with multiple initial states (MDFA) is a tuple $M = (Q, \Sigma, \delta, Q_0, F)$ and M is identical to a DFA except that there is a set of initial states Q_0. The language accepted by an MDFA M is $T(M) = \{w \in \Sigma^* | \delta(Q_0, w) \cap F \neq \emptyset\}$. An MDFA with $k = |Q_0|$ initial states is denoted by k-MDFA.

Let $\mathcal{A}, \mathcal{B}$ be two classes of finite automata. Following the notation of [8], we say that $\mathcal{A} \longrightarrow \mathcal{B}$ denotes the problem of converting a type-$\mathcal{A}$ finite automaton to a minimal type-$\mathcal{B}$ finite automaton. Formally:

PROBLEM $\mathcal{A} \longrightarrow \mathcal{B}$.
INSTANCE A type-$\mathcal{A}$ finite automaton M and an integer l.
QUESTION Is there an l-state type-$\mathcal{B}$ finite automaton M' such that
$T(M') = T(M)$?

3. Minimizing MDFAs is computationally hard

In this section, we are going to show that the minimization problem for k-MDFAs is NP-complete. Throughout this section, $k \geqslant 2$ denotes a constant integer.

Lemma 1. *k-MDFA $\longrightarrow$ k-MDFA is in* NP.

Proof. The problem is in NP, since a k-MDFA M' with $|M'| \leqslant l$ can be determined nondeterministically and the equality $T(M) = T(M')$ can be tested in polynomial time as is shown below. At first M and M' are converted to DFAs in the following manner. Let $M = (Q, \Sigma, \delta, \{q_0^1, q_0^2, \ldots, q_0^k\}, F)$. Then, we define k DFAs as follows:

$$M_1 = (Q, \Sigma, \delta, q_0^1, F), M_2 = (Q, \Sigma, \delta, q_0^2, F), \ldots, M_k = (Q, \Sigma, \delta, q_0^k, F).$$

We observe that $T(M_1) \cup T(M_2) \cup \ldots \cup T(M_k) = T(M)$ and we construct a DFA $\widehat{M}$ as the Cartesian product of $M_1, M_2, \ldots, M_k$ accepting $T(M_1) \cup \ldots \cup T(M_k)$ in the usual way. A DFA $\widehat{M}'$ can be constructed from M' analogously. Both constructions can be done in polynomial time. Since the time complexity of the inequivalence problem of two DFAs is in NLOGSPACE $\subseteq$ P [10], we can test the equivalence $T(\widehat{M}) = T(\widehat{M}')$ in polynomial time. □

The NP-hardness of the problem will be shown by reduction from the minimum inferred DFA problem which was shown to be NP-hard in [3]. In [8] the NP -hardness of the minimum inferred DFA problem is used to prove that the minimum union generation problem is NP-complete.

PROBLEM	Minimum inferred DFA [1].
INSTANCE	Finite alphabet Σ, two finite subsets $S, T \subset \Sigma^*$, integer l.
QUESTION	Is there an l-state DFA that accepts a language L such that $S \subseteq L$ and $T \subseteq \Sigma^* \setminus L$?

Such an l-state DFA will be called *consistent* with S and T.

The essential idea of the reduction in [8] is to design a language L_5 depending on S and T such that L_5 can be decomposed into the union of two DFAs with certain size bounds if and only if there is a DFA being consistent with S and T and satisfying a certain size bound. The difficult part is the “only-if”- portion. To this end, the notions of a “tail” and a “waist” are introduced, i.e., a sequence of states connected with #-edges only and ending at some state with no outgoing edges or with at least one outgoing edge, respectively. With the help of these two elements in L_5, it is possible to show that exactly one DFA contains a tail or a waist, respectively. Then it is not difficult to construct a DFA consistent with S and T.

We now want to adopt the basic ideas of the above construction. We have to show that a modified language L_5' is accepted by a k-MDFA if and only if there is a DFA being consistent with S and T and satisfying a certain size bound. To apply the above result, our goal here is to decompose the k-MDFA into two sub-DFAs whose union is the language L_5 from [8]. To this end, the beginning of all words in L_5 is suitably modified. This enables us to show that there are two initial states such that from each of these states either all words ending in the tail or all words ending in the waist are accepted. This fact finally leads to the desired decomposition.

We follow the notations given in [8]. W.l.o.g. we may assume that $S \cap T = \emptyset$. Let #, \$ and £ be symbols not in Σ. Let $\Sigma' = \Sigma \cup \{\#, \$, £\}$, $m = l + \mathrm{size}(\overline{T} \cap \overline{S})$, and $t = \max(k, m)$.

$$L_1' = \overline{T},$$
$$L_2' = \overline{T} \cap \overline{S},$$
$$L_3' = \{\$, £\}\#^t L_2' \#^m (£\#^t L_2' \#^m)^*,$$

$$L'_4 = \$\#^t L'_1 \#^m,$$
$$L'_5 = L'_3 \cup L'_4.$$

The main task in the below proof of the NP-hardness of k-MDFA $\longrightarrow$ k-MDFA is to show that a DFA consistent with S and T can be constructed from a k-MDFA accepting a language L'_5 where both automata are satisfying certain size bounds. To this end, L'_5 is defined as the union of two languages L'_3 and L'_4 which either contain a waist or a tail. Additionally, certain symbols and another sequence of #-symbols, called "core," are added to the beginning of the languages. These elements help to show that there are only two essential initial states from which either all words containing subwords from $S \subset L'_1 = \overline{T}$ or containing subwords from $L'_2 = \overline{T} \cap \overline{S}$ can be accepted. Then it is shown how such a 2-MDFA can be divided into two DFAs from which a DFA being consistent with S and T can be constructed.

We first show a technical lemma following [8].

Lemma 2. *Let R be regular and M' a DFA consistent with S and T.*
(1) $\text{size}(\$\#^t R\#^m) = \text{size}((\$\#^t R\#^m)^+) = t + m + 1 + \text{size}(R)$,
(2) $\text{size}(L'_3) = t + m + 1 + \text{size}(L'_2)$,
(3) $\$\#^t L'_1 \#^m = \$\#^t (L'_2 \cup T(M'))\#^m$.

Proof. Claims (1) and (3) can be shown similarly to the Claims 4.1. and 4.2. in [8]. Claim (2) can be shown similarly to the first claim. □

Lemma 3. *k-MDFA $\longrightarrow$ k-MDFA is NP-hard.*

Proof. We first present the reduction and then show its correctness.

Let $M_1 = (Q_1, \Sigma', \delta_1, q_0^1, F_1)$ and $M_2 = (Q_2, \Sigma', \delta_2, q_0^2, F_2)$ be two minimal DFAs such that $T(M_1) = L'_3$ and $T(M_2) = L'_4$. W.l.o.g. we may assume that $Q_1 \cap Q_2 = \emptyset$. We choose $k - 2$ additional states $\{q_0^3, \ldots, q_0^k\}$ not in $Q_1 \cup Q_2$. Then we can construct a k-MDFA:

$$M = (Q_1 \cup Q_2 \cup \{q_0^3, \ldots, q_0^k\}, \Sigma', \delta, \{q_0^1, q_0^2, \ldots, q_0^k\}, F_1 \cup F_2).$$

For $\sigma \in \Sigma'$ we define $\delta(q, \sigma) = \delta_1(q, \sigma)$ if $q \in Q_1$, $\delta(q, \sigma) = \delta_2(q, \sigma)$ if $q \in Q_2$, and $\delta(q_0^i, \sigma) = \delta(q_0^1, \sigma)$ for $i \in \{3, \ldots, k\}$. Then $T(M) = L'_5$. The instance S, T, l has been transformed to the instance $M, 3m + 2t + k$. Let $m' = \sum_{w \in S \cup T} |w| + l$ be the size of the instance of the minimum inferred DFA problem, then it is easily seen that M can be constructed from S, T, l in time bounded by a polynomial in m'. The correctness of the reduction is shown by the following claim.

Claim 4. *There is an l-state DFA consistent with S and T if and only if $T(M) = L'_5$ is accepted by a k-MDFA M' having at most $3m + 2t + k$ states.*

"$\Rightarrow$":

Let M'' be a DFA consistent with S and T and $|M''| \leqslant l$. Let M_1 and M_2 be the minimal DFAs with $T(M_1) = L'_3$ and $T(M_2) = \$\#^t T(M'')\#^m$. Then we have $|M_1| = t + m + 1 + \text{size}(L'_2) = t + 2m + 1 - l$, $|M_2| \leqslant t + m + l + 1$ and therefore $|M_1| + |M_2| \leqslant 3m + 2t + 2$.

Considering the two symbols \$, £ we can show analogously to [8] that $T(M_1) \cup T(M_2) = L'_5$. Now we choose $k-2$ additional initial states $\{q_0^3, \dots, q_0^k\} \not\subseteq Q_1 \cup Q_2$ and construct a k-MDFA $M' = (Q_1 \cup Q_2 \cup \{q_0^3, \dots, q_0^k\}, \Sigma', \delta, \{q_0^1, q_0^2, \dots, q_0^k\}, F_1 \cup F_2)$ in the above-mentioned manner. We thus obtain a k-MDFA such that $|M'| \leqslant 3m+2t+k$ and $T(M') = L'_5$.
"$\Leftarrow$":

Let $M = (Q, \Sigma', \delta, \{q_0^1, q_0^2, \dots, q_0^k\}, F)$ be a k-MDFA such that $T(M) = L'_5$ and $|M| \leqslant 3m + 2t + k$. We may assume that M is minimal. We have to construct an l-state DFA M' consistent with S and T. At first we show that M can be modified such that M has the form depicted in Fig. 1. Then we prove that M can be easily decomposed into two DFA M_1 and M_2 such that $|M_1| + |M_2| \leqslant 3m$ and $T(M_1) \cup T(M_2) = L'_1\#^m \cup (L'_2\#^m)^+$. This situation is exactly the situation of the "if"-part in Claim 4.3 of [8]. Hence, we can conclude that an l-state DFA M' consistent with S and T can be constructed.

(1) W.l.o.g. $S \neq \emptyset$. If $S = \emptyset$, then any DFA accepting the empty set is a DFA consistent with S and T. Hence, there is a one-state DFA accepting the empty set, and there is in particular an l-state DFA M' consistent with S and T.

(2) Let $w = \$w_1$ with $w_1 \in \#^t S \#^m$ and $w' = w'_1 w'_2$ with $w'_1, w'_2 \in £\#^t L'_2 \#^m$ be two words in L'_5. Then there are initial states q_0^i and q_0^j such that $\delta(q_0^i, w) \in F$ and $\delta(q_0^j, w') \in F$. We remark that q_0^i and q_0^j may be identical.

(3) *Claim*: M contains exactly one waist, one tail and two distinct cores.

According to [8], a waist is defined as a sequence of states $q_1, q_2, \dots, q_m$ such that $\delta(q_i, \#) = q_{i+1}$ for all $i \in \{1, 2, \dots, m-1\}$ and q_m is an accepting state and has an outgoing £-edge. A tail is defined as a sequence of states $q_1, q_2, \dots, q_m$ such that $\delta(q_i, \#) = q_{i+1}$ for all $i \in \{1, 2, \dots, m-1\}$ and q_m is an accepting state and has no outgoing edges. A core is defined as a sequence of states $q_1, q_2, \dots, q_{t+1}$ such that $\delta(q_i, \#) = q_{i+1}$ for all $i \in \{1, 2, \dots, t\}$ and q_{t+1} is nonaccepting and has outgoing edges, but no outgoing £-edge. Obviously, M contains at least one waist, one tail, and one core. We observe that all initial states from which a word in L'_5 can be accepted have a \$-edge or £-edge or both to the first state of a core. Consider the above word $w = \$w_1$. If we have exactly one core, then $\delta(q_0^i, \$) = \delta(q_0^j, £)$ and hence $\delta(q_0^j, £w_1) = \delta(\delta(q_0^i, \$), w_1) = \delta(q_0^i, w) \in F$ which is a contradiction. If M contains two cores which are not distinct, then there are initial states q_0^i, q_0^j, a state $q \in Q$, and $x \in S$ such that $\delta(q_0^i, \$\#^{i'}) = q = \delta(q_0^j, £\#^{j'})$ with $1 \leqslant i', j' \leqslant t$ and $\delta(q, \#^{t-i'} x \#^m) \in F$. Then $\delta(q_0^j, £\#^{j'}\#^{t-i'} x \#^m) \in F$—contradiction. If M contains more than two cores, more than one waist, or more than one tail, then $|M|$ exceeds $3m + 2t + k$, since M requires at least $2m$ states for waist and tail, $2t + 2$ states for two cores, k initial states, and at least $m \leqslant t$ states for an additional waist, tail, or core.

(4) W.l.o.g. we may assume that w will be accepted from q_0^i passing through core$_1$ and the tail and w' will be accepted from q_0^j passing through core$_2$ and the waist. Let $q_t = \delta(q_0^i, w)$ and $q_w = \delta(q_0^j, w'_1)$ denote the last states in the tail and the waist. Let $q^1 = \delta(q_0^i, \$)$ and $q^2 = \delta(q_0^j, £)$ denote the first states of core$_1$ and core$_2$. By $q_c^1 = \delta(q_0^i, \$\#^t)$ and $q_c^2 = \delta(q_0^j, £\#^t)$ we denote the last states of core$_1$ and core$_2$. Since w is accepted passing through core$_1$, we can conclude that $q^2 = \delta(q_w, £)$ is the starting state of the loop.

(5) *Claim*: All initial states have no incoming edges.

Let q_0^p with $p \in \{1, 2, \ldots, k\}$ be an initial state. We may assume that from q_0^p at least one word in L_5' can be accepted, otherwise all incoming edges can be removed without affecting the accepted language. Now, assume that q_0^p has an incoming edge. Then this must be a #-edge. We have to show that $q_0^p \neq q_t$ and $q_0^p \neq q_w$. If $q_0^p = q_t$ or $q_0^p = q_w$, then $q_0^p \in F$ by definition of q_t and q_w and therefore $\varepsilon \in L_5'$— contradiction.

(6) *Claim*: $\delta(q^1, \#^t S \#^m) \subseteq F$ and $\delta(q^2, \#^t S \#^m) \cap F = \emptyset$.

By way of contradiction we assume that there is a string $x \in \#^t S \#^m$ such that $\delta(q^1, x) \notin F$. Since \$$x \in L_5'$, we then know that $\delta(q^2, x) \in F$ and therefore $\delta(q_0^j, £x) = \delta(q^2, x) \in F$ which is a contradiction. To show the second claim we assume that there is a string $x \in \#^t S \#^m$ such that $\delta(q^2, x) \in F$. Since $\delta(q_0^j, £) = q^2$, we have $\delta(q_0^j, £x) \in F$— contradiction.

(7) *Claim*: $\delta(q^2, \#^t L_2' \#^m (£\#^t L_2' \#^m)^*) \subseteq F$.

We assume that there is a string $x \in \#^t L_2' \#^m (£\#^t L_2' \#^m)^*$ such that $\delta(q^2, x) \notin F$. Since $£x \in L_5'$, we know that $\delta(q^1, x) \in F$. We have $\delta(q_0^j, w_1' £x) = \delta(q^2, x) \notin F$. Then there must be an initial state q_0' with $\delta(q_0', w_1' £x) = \delta(q^1, x) \in F$, in particular $\delta(q_0', w_1' £) = q^1$. Then we have $\delta(q_0', w_1' £ w_1) = \delta(q^1, w_1) \in F$ which is a contradiction.

(8) *Claim*: M can be modified to the form depicted in Fig. 1. (The initial states $q_0^3, \ldots, q_0^k$ are not included.)

At first we remove all edges from initial states to any other states. We choose two different initial states q_0^1 and q_0^2 and then insert the following edges: $q_0^1 \xrightarrow{\$} q^1$, $q_0^2 \xrightarrow{\$,£} q^2$, and $q_0^i \xrightarrow{\$,£} q^2$ for $i \in \{3, \ldots, k\}$. We observe that due to (5), (6), and (7) the modified automaton still recognizes L_5'. In particular, L_3' is accepted from q_0^2 and all words in \$$\#^t S \#^m$ are accepted only from q_0^1.

(9) We now look at the two DFAs obtained when considering only one initial state in M. We define the set of reachable states as follows:

$$\mathcal{E}(q_0^1) = \{q \in Q | \exists x, x' \in (\Sigma')^* : \delta(q_0^1, x) = q \wedge \delta(q, x') \in F\}$$

$\mathcal{E}(q_0^2)$ is defined analogously.

We first claim that there is no edge from $p \in \mathcal{E}(q_0^2)$ to a state q from which q_t can be obtained. Assume by way of contradiction that there are $p \in \mathcal{E}(q_0^2)$, $q \in Q$, $s \in \Sigma'$, and $u \in (\Sigma')^*$ such that $\delta(p, s) = q$ and $\delta(q, u) = q_t \in F$. Since $p \in \mathcal{E}(q_0^2)$, there are strings $x, x' \in (\Sigma')^*$ such that $\delta(q_0^2, x) = p$ and $\delta(p, x') \in F$. Owing to (8), we may assume that x starts with £. We then know that $\delta(q_0^2, xsu) = q_t \in F$, but $\delta(q_0^2, xsuxsu) \notin F$, because q_t has no outgoing edges. Moreover, $\delta(q_0^1, xsuxsu) \notin F$, since q_0^1 has no outgoing £-edge. Hence $xsuxsu \notin L_5'$ which is a contradiction, because $xsu \in L_3'$ and therefore $xsuxsu \in L_3' \subset L_5'$. Furthermore, we observe that all edges from states in $\mathcal{E}(q_0^1)$ to states in $\mathcal{E}(q_0^2)$ can be removed. If we have such an edge, all words passing this edge will be accepted in the waist and therefore are in L_3'. Due to (7) and (8), these words can already be accepted from q_0^2. So, removing such edges does not affect the accepted language. We observe that this modification yields $\mathcal{E}(q_0^1) \cap \mathcal{E}(q_0^2) = \emptyset$.

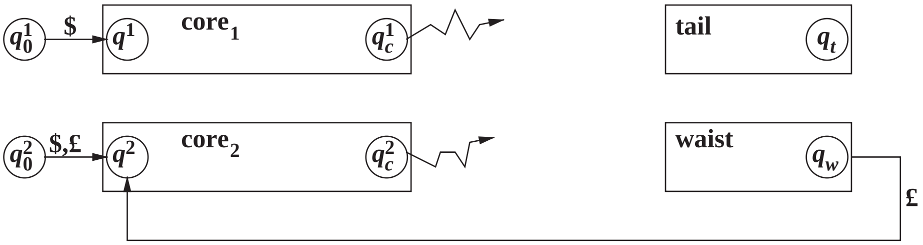

Fig. 1. The modified k-MDFA M.

(10) Since the sets of reachable states are distinct, we then obtain two DFAs $M_1' = (Q_1', \Sigma', \delta_1', q_0^1, F_1')$ and $M_2' = (Q_2', \Sigma', \delta_2', q_0^2, F_2')$ after having minimized the DFAs $(\mathcal{E}(q_0^1), \Sigma', \delta, q_0^1, F)$ and $(\mathcal{E}(q_0^2), \Sigma', \delta, q_0^2, F)$. Due to (6) and (8), we know that $L_4' \supseteq T(M_1') \supseteq \$\#^t S\#^m$ and $T(M_2') = L_3'$. Furthermore, $|M_1'| + |M_2'| \leqslant 3m + 2t + 2$, since $Q_1' \cap Q_2' = \emptyset$.

(11) Starting from M_1' we define another DFA M_1 by removing q_0^1 and the first t states of core$_1$. We define q_c^1 as new initial state and observe that $L_1'\#^m \supseteq T(M_1) \supseteq S\#^m$. Starting from M_2' we define another DFA M_2 by removing q_0^2 and the first t states of core$_2$. We define q_c^2 as new initial state. The £-edge from q_w to q^2 is replaced by the following edges: if $\delta_2'(q_c^2, \sigma) = q$ for $\sigma \in \Sigma$, we add a σ-edge from q_w to q. It is easy to see that $T(M_2) = (L_2'\#^m)^+$. Hence we have $T(M_1) \cup T(M_2) = L_1'\#^m \cup (L_2'\#^m)^+$. Moreover, $|M_1| + |M_2| \leqslant 3m$.

(12) We have $|M_2| = \text{size}((L_2'\#^m)^+) = m + \text{size}(\overline{T} \cap \overline{S}) = 2m - l$ and therefore $|M_1| \leqslant 3m - |M_2| = 3m - 2m + l = m + l$. Removing the tail in M_1 yields an l-state DFA M' consistent with S and T.

This shows that the reduction is correct and thus the NP-hardness of the problem. □

Lemmas 1 and 3 imply the following theorem.

Theorem 5. *k-MDFA $\longrightarrow$ k-MDFA is* NP*-complete.*

Corollary 6. *Let $k, k' \geqslant 2$ be two constant numbers. Then DFA $\longrightarrow$ k-MDFA and k-MDFA $\longrightarrow$ k'-MDFA are* NP*-complete.*

Theorem 7. *Let M be a k-MDFA and M' be a k'-MDFA. Then the following problems can be solved in polynomial time. Is $T(M) = T(M')$? Is $T(M) \subseteq T(M')$? Is $T(M) \subset T(M')$? Is $T(M) = \Sigma^*$?*

Proof. According to the construction given in Lemma 1, k-MDFAs can be converted to DFAs in polynomial time. Since the above-mentioned decidability questions are solvable for DFAs in polynomial time [9,10,16], the theorem is proven. □

4. Minimizing NFAs with fixed finite branching is computationally hard

In this section, we are going to show that the minimization problem for NFAs with branching $\beta = k$ is NP-complete for $k \geqslant 3$. At first we show that it is decidable in polynomial time whether or not an arbitrary NFA has branching k for a fixed number k. In general, it is PSPACE-complete to decide whether or not an arbitrary NFA has finite branching [13].

Lemma 8. *Let M be an NFA and $k \geqslant 2$ be a constant integer. Then the question whether M has branching k can be solved in polynomial time.*

Proof. We consider the language

$$T_k(M) = \{w \in \Sigma^* | \exists \text{ accepting computation } \pi \text{ of } w \text{ with } \beta_M(\pi) \leqslant k\}.$$

It is shown in [4] that a DFA M_k accepting $T_k(M)$ can be constructed. We reproduce the construction and observe that it can be done in time polynomially bounded in $|M|$ and that the resulting DFA has size $\mathrm{O}(|M|^k)$. Let $\Sigma_T = \{[p, a, q] \in Q \times \Sigma \times Q | q \in \delta(p, a)\}$ be the alphabet of triples corresponding to moves of M. The set

$$\begin{aligned} R = \{&[q_0, a_1, q_1][q_1, a_2, q_2] \ldots [q_{n-1}, a_n, q_n] \in \Sigma_T^* | n \geqslant 1, q_n \in F\} \\ &\cup \{\varepsilon | q_0 \in F\} \end{aligned}$$

is then the regular set of all accepting computations of M. Obviously, a DFA accepting R is the “deterministic version” of M with $\text{size}(R) = \mathrm{O}(|M|)$ that can be constructed in time $\mathrm{O}(|M| \cdot |\Sigma_T|) = \mathrm{O}(|M|^3)$. We consider two homomorphisms $f : \Sigma_T^* \to \Sigma^*$ and $g : \Sigma_T^* \to \{c, d\}^*$ such that $f([p, a, q]) = a$ and $g([p, a, q]) = \varepsilon$ if $|\delta(p, a)| = 1$ and $g([p, a, q]) = c^{|\delta(p,a)|}d$ otherwise. Furthermore,

$$S_k = \{c^{j_1}d \cdots c^{j_t}d | t \geqslant 1, \text{ each } j_i \geqslant 2, j_1 \cdot j_2 \cdot \ldots \cdot j_t \leqslant k\} \cup \{\varepsilon\}.$$

Since k is a constant number, it follows that $\text{size}(S_k)$ and $\text{size}(g^{-1}(S_k))$ are in $\mathrm{O}(1)$ and the corresponding DFAs can be constructed in constant time and $\mathrm{O}(|\Sigma_T|) = \mathrm{O}(|M|^2)$, respectively. Constructing the Cartesian product of R and $g^{-1}(S_k)$, we obtain a DFA accepting $R \cap g^{-1}(S_k)$ of size $\mathrm{O}(|M|)$ in time $\mathrm{O}(|M|^3)$. The construction of an NFA M' accepting $f(R \cap g^{-1}(S_k))$ can be done by relabeling of the edges of the DFA for $R \cap g^{-1}(S_k)$, and can be performed in time $\mathrm{O}(|M|^3)$. We observe that M' has branching k, $|M'| = \mathrm{O}(|M|)$, and $T(M') = f(R \cap g^{-1}(S_k)) = f(\{\pi \in R | \beta_M(\pi) \leqslant k\}) = T_k(M)$. Applying the construction presented in [11], we can convert M' to a k-MDFA with at most $k|M'| + 1 = \mathrm{O}(|M|)$ states in time $\mathrm{O}(|M|)$. Then, this k-MDFA can be converted to a DFA M_k with at most $\mathrm{O}(|M|^k)$ states in time $\mathrm{O}(|M|^k)$ analogous to the construction of Lemma 1.

Since $T_k(M) \subseteq T(M)$, we have $T(M) \setminus T_k(M) = \emptyset \Leftrightarrow \beta_M \leqslant k$. Since M_k is a DFA, we can simply construct a DFA M'_k accepting the complement $\Sigma^* \setminus T_k(M)$.

$$\begin{aligned} \beta_M \leqslant k \ &\Leftrightarrow \ T(M) \setminus T_k(M) = \emptyset \Leftrightarrow T(M) \cap \overline{T_k(M)} = \emptyset \\ &\Leftrightarrow \ T(M) \cap T(M'_k) = \emptyset. \end{aligned}$$

Since M is an NFA and M'_k is a DFA, we can construct, in polynomial time, an NFA $\widehat{M}$ of size $\mathrm{O}(|M| \cdot |M|^k)$ as the Cartesian product of M and M'_k accepting $T(M) \cap T(M'_k)$. Due to the result in [9], the nonemptiness of $T(\widehat{M})$ can be tested in NLOGSPACE $\subseteq$ P. If $T(\widehat{M}) \neq \emptyset$, then $\beta_M > k$. If $T(\widehat{M}) = \emptyset$, then we know that $\beta_M \leqslant k$. To find out whether $\beta_M = k$, we construct $T_{k-1}(M)$ if $k - 1 \geqslant 1$. This can be done in polynomial time as well as the test for inequivalence of $T_{k-1}(M)$ and $T_k(M)$. If both sets are inequivalent, then $\beta_M = k$; otherwise $\beta_M < k$. □

Lemma 9. NFA($\beta = k$) $\longrightarrow$ NFA($\beta = k$) *is in* NP *for* $k \geqslant 2$.

Proof. We first determine an NFA M' with $|M'| \leqslant l$ nondeterministically. Owing to Lemma 8, we can test whether M' has branching k in polynomial time. We next convert M and M' to k-MDFAs $\widehat{M}$ and $\widehat{M}'$ with at most $k|M|+1$ and $k|M'|+1$ states applying the construction presented in [11]. The equality of $T(\widehat{M})$ and $T(\widehat{M}')$ can then be tested in polynomial time analogous to the considerations of Lemma 1. Hence the above problem is in NP. □

The NP-hardness of the problem will be shown by reduction from the minimum inferred DFA problem similar to the proof for MDFAs. In detail, we want to transform an NFA with fixed finite branching k into a 2-MDFA accepting L_5'. Owing to Lemma 3, we then can construct an appropriate DFA consistent with S and T. An obvious, but essential observation for NFAs with finite branching is that an accepting computation with minimum branching does not contain a move with branching $\beta > 1$ which is located within a loop, because otherwise the branching of the NFA would be infinite. Thus, we modify the language L_5' by adding loops at the beginning of all words. Therefore, we can show that a given NFA with fixed finite branching k has exactly one nondeterministic move with branching k. Furthermore, this move has to be the first move. It is then easy to convert this NFA to a 2-MDFA accepting L_5'.

Let $m = l + \text{size}(\overline{T} \cap \overline{S})$ and $n = 5m+1$. Then we define the following languages.

$$
\begin{aligned}
L_1'' &= L_1',\\
L_2'' &= L_2',\\
L_3'' &= \{\$, \pounds\}\#^m(\#^{m+1})^* L_2''\#^m(\pounds\#^m(\#^{m+1})^* L_2''\#^m)^*,\\
L_4'' &= \$\#^m(\#^{m+1})^* L_1''\#^m,\\
L^i &= \{(\$\#^{in^k-1})^+\} \ (1 \leqslant i \leqslant k-2),\\
L_5'' &= L^1 \cup L^2 \cup \ldots \cup L^{k-2},\\
L_6'' &= L_3'' \cup L_4'' \cup L_5''.
\end{aligned}
$$

Lemma 10. *Let R be a regular language, M' a DFA consistent with S and T, and $L = \{\$\#^m(\#^{m+1})^*\} \cup L_5''$:*

(1) $\text{size}(\$\#^m(\#^{m+1})^* R\#^m) = \text{size}((\$\#^m(\#^{m+1})^* R\#^m)^+) = 2m+1+\text{size}(R)$,
(2) $\text{size}(L_3'') = 2m+1+\text{size}(L_2'')$,
(3) $\$\#^m(\#^{m+1})^* L_1''\#^m = \$\#^m(\#^{m+1})^*(L_2'' \cup T(M'))\#^m$,
(4) $\text{size}(L^i) = in^k + 1$,
(5) $\text{size}(L) \geqslant n^k + 2n^k + \cdots + (k-2)n^k + (k-2)n^k + 1 + (m+1)$.

Proof. Claims (1)–(3) can be shown analogously to those of Lemma 2. Claim (4) is obvious. To show (5) we use the Nerode equivalence relation $\equiv_L$ on L and prove that the index

$$\text{index}(\equiv_L) \geqslant n^k + 2n^k + \cdots + (k-2)n^k + (k-2)n^k + 1 + (m+1).$$

For $x, y \in \Sigma^*$, $\equiv_L$ is defined as

$$x \equiv_L y :\Longleftrightarrow xz \in L \Leftrightarrow yz \in L \text{ for all } z \in \Sigma^*.$$

Let $1 \leqslant i \leqslant k-2$; we define the following sets of strings:

$$\begin{aligned}
A_i &= \{a_{i,0}, a_{i,1}, \ldots, a_{i,in^k-1}\} \text{ with } a_{i,j} = \$\#^{in^k-1}\$\#^j \text{ and } 0 \leqslant j \leqslant in^k - 1,\\
B &= \{b_1, b_2, \ldots, b_{m+1}\} \text{ with } b_j = \$\#^{(k-2)n^k-1}\#^j \text{ and } 1 \leqslant j \leqslant m+1,\\
C &= \{c_0, c_1, \ldots, c_{(k-2)n^k-1}\} \text{ with } c_j = \$\#^j \text{ and } 0 \leqslant j \leqslant (k-2)n^k - 1,\\
D &= \{\varepsilon\}.
\end{aligned}$$

Obviously, $|A_i| = in^k$, $|B| = m+1$, $|C| = (k-2)n^k$, and $|D| = 1$. We have to show that each two words from $A_1 \cup A_2 \cup \ldots \cup A_{k-2} \cup B \cup C \cup D$ are not $\equiv_L$-equivalent.

(1) *Claim*: Let $x, y \in A_i$ such that $x \neq y$. Then $x \not\equiv_L y$.

Let $x = \$\#^{in^k-1}\$\#^{i'}$ and $y = \$\#^{in^k-1}\$\#^{j'}$ with $0 \leqslant i' < j' \leqslant in^k - 1$. We define $z = \#^{in^k-1-j'}\$\#^{in^k-1}$ and obtain that $xz \notin L$ and $yz \in L$.

(2) *Claim*: Let $x, y \in B$ such that $x \neq y$. Then $x \not\equiv_L y$.

Let $x = \$\#^{(k-2)n^k-1}\#^{i'}$ and $y = \$\#^{(k-2)n^k-1}\#^{j'}$ with $1 \leqslant i' < j' \leqslant m+1$. Then $j' = i' + r$ with $1 \leqslant r \leqslant m$. Let $i'' \geqslant 0$ be the minimal integer such that $(k-2)n^k - 1 + i' + i'' - m$ is a multiple of $m+1$. Then $(k-2)n^k - 1 + i' + i'' = m + t(m+1)$ with $t \geqslant 1$. We now set $z = \#^{i''}$ and observe that $xz = \$\#^{(k-2)n^k-1}\#^{i'}\#^{i''} = \$\#^{m+t(m+1)} \in L$, but $yz = \$\#^{(k-2)n^k-1}\#^{j'}\#^{i''} \notin L$, since $m + t(m+1) = (k-2)n^k - 1 + i' + i'' < (k-2)n^k - 1 + j' + i'' = (k-2)n^k - 1 + i' + r + i'' = m + t(m+1) + r < m + (t+1)(m+1)$.

(3) *Claim*: Let $x, y \in C$ such that $x \neq y$. Then $x \not\equiv_L y$.

Let $x = \$\#^{i'}$ and $y = \$\#^{j'}$ with $0 \leqslant i' < j' \leqslant (k-2)n^k - 1$. We set $z = \#^{(k-2)n^k-1-j'}\$\#^{(k-2)n^k-1}$ and obtain that $xz \notin L$ and $yz \in L$.

(4) *Claim*: Let $x \in A_i$ and $y \in A_j$ with $1 \leqslant j \leqslant k-2$ and $i \neq j$. Then $x \not\equiv_L y$.

Let $x = \$\#^{in^k-1}\$\#^{i'}$ and $y = \$\#^{jn^k-1}\$\#^{j'}$ with $0 \leqslant i' \leqslant in^k - 1$ and $0 \leqslant j' \leqslant jn^k - 1$. W.l.o.g. we may assume that $i < j$. We define $z = \#^{jn^k-1-j'}\$\#^{jn^k-1}$ and obtain that $xz \notin L$ and $yz \in L$.

(5) *Claim*: Let $x \in A_i$ and $y \in B$. Then $x \not\equiv_L y$.

Let $x = \$\#^{in^k-1}\$\#^{i'}$ and $y = \$\#^{(k-2)n^k-1}\#^{j'}$ with $0 \leqslant i' \leqslant in^k - 1$ and $1 \leqslant j' \leqslant m+1$. We set $z = \#^{in^k-1-i'}\$\#^{in^k-1}$ and obtain that $xz \in L$ and $yz \notin L$.

(6) *Claim*: Let $x \in A_i$ and $y \in C$. Then $x \not\equiv_L y$.

Let $x = \$\#^{in^k-1}\$\#^{i'}$ and $y = \$\#^{j'}$ with $0 \leqslant i' \leqslant in^k - 1$ and $0 \leqslant j' \leqslant (k-2)n^k - 1$. Let $j'' \geqslant 0$ be the minimal integer such that $j' + j'' - m$ is a multiple of $m+1$. Then $j' + j'' = m + t(m+1)$ with $t \geqslant 0$. We now set $z = \#^{j''+(m+1)(in^k-1)}$ and observe that $yz = \$\#^{j'+j''+(m+1)(in^k-1)} = \$\#^{m+(m+1)(t+in^k-1)} \in L$, but $xz = \$\#^{in^k-1}\$\#^{i'}\#^{j''+(m+1)(in^k-1)} \notin L$, since $i' + j'' + (m+1)(in^k-1) > in^k - 1$.

(7) *Claim*: Let $x \in B$ and $y \in C$. Then $x \not\equiv_L y$.

Let $x = \$\#^{(k-2)n^k-1}\#^{i'}$ and $y = \$\#^{j'}$ with $1 \leqslant i' \leqslant m+1$ and $0 \leqslant j' \leqslant (k-2)n^k - 1$. We define $z = \#^{(k-2)n^k-1-j'}\$\#^{(k-2)n^k-1}$ and obtain that $xz \notin L$ and $yz \in L$.

(8) *Claim*: Let $x \in A_1 \cup \ldots \cup A_{k-2} \cup B \cup C$ and $y \in D$. Then $x \not\equiv_L y$.

Let $x \in A_1 \cup \ldots \cup A_{k-2} \cup B \cup C$, $y = \varepsilon$, and $z = \$\#^{m+(m+1)((k-2)n^k-1)}$. Then $xz \notin L$, since $m + (m+1)((k-2)n^k - 1) > (k-2)n^k - 1$, and $yz = z \in L$. Hence, $x \not\equiv_L y$.

Thus,

$$\begin{aligned} \text{index}(\equiv_L) &\geqslant |A_1| + \cdots + |A_{k-2}| + |B| + |C| + |D| \\ &= n^k + 2n^k + \cdots + (k-2)n^k + (k-2)n^k + (m+1) + 1 \end{aligned}$$

and (5) is proven. □

Lemma 11. NFA$(\beta = k) \longrightarrow$ NFA$(\beta = k)$ *is* NP*-hard for* $k \geqslant 3$.

Proof. We first present the reduction and then show its correctness.

Let $M_1 = (Q_1, \Sigma', \delta_1, q_0^1, F_1)$ and $M_2 = (Q_2, \Sigma', \delta_2, q_0^2, F_2)$ be two minimal DFAs such that $T(M_1) = L_3''$ and $T(M_2) = L_4''$. Furthermore, we consider $k-2$ minimal DFAs $M_i = (Q_i, \Sigma', \delta_i, q_0^i, F_i)$ $(3 \leqslant i \leqslant k)$ accepting $L^1, L^2, \ldots, L^{k-2}$. W.l.o.g. we may assume that Q_1, Q_2, …, and Q_k are pairwise distinct. We observe that for $3 \leqslant i \leqslant k$ the states q_0^i have no incoming edges and only one outgoing edge to a nontrap state, namely a \$-edge. Moreover, q_0^1 has no incoming edges and only two outgoing edges to nontrap states, namely a \$-edge and a £-edge. We remove q_0^1 from M_1 and q_0^i from M_i for $3 \leqslant i \leqslant k$ and construct an NFA $M = (Q, \Sigma', \delta, q_0^2, F)$ with

$$\begin{aligned} &Q = (Q_1 \setminus \{q_0^1\}) \cup Q_2 \cup (Q_3 \setminus \{q_0^3\}) \cup \ldots \cup (Q_k \setminus \{q_0^k\}), \\ &F = F_1 \cup F_2 \cup \ldots \cup F_k. \end{aligned}$$

For $\sigma \in \Sigma'$ and $1 \leqslant i \leqslant k$ we define $\delta(q, \sigma) = \delta_i(q, \sigma)$ if $q \in Q_i$. Furthermore, $\delta(q_0^2, £) = \{\delta_1(q_0^1, £)\}$ and $\delta(q_0^2, \$) = \{\delta_i(q_0^i, \$) | i \in \{1, 2, \ldots, k\}\}$. Then, $T(M) = L_6''$ and M is an NFA with branching k.

The instance S, T, l has been transformed to M, $5m + 1 + \sum_{i=1}^{k-2} in^k$. Let $m' = \sum_{w \in S \cup T} |w| + l$ be the size of the instance of the minimum inferred DFA problem. Then M can be constructed from S, T, l in time bounded by a polynomial in m'. The correctness of the reduction is shown by the following claim.

Claim 12. *There is an* l*-state DFA consistent with* S *and* T *if and only if* $T(M) = L_6''$ *is accepted by an NFA* M' *with branching* $\beta_M = k$ *that has at most* $5m + 1 + \sum_{i=1}^{k-2} in^k$ *states*. (See Fig. 2).

“⇒”:

Let M'' be a DFA consistent with S and T and $|M''| \leqslant l$. Let M_1 and M_2 be the minimal DFAs with $T(M_1) = L_3''$, $T(M_2) = \$\#^m(\#^{m+1})^* T(M'')\#^m$. Furthermore, $M_3, \ldots, M_k$ are minimal DFAs accepting $L^1, \ldots, L^{k-2}$. Analogous to the proof of Lemma 3 and the above considerations we can construct an NFA M' with $\beta_{M'} = k$ such that $T(M') = L_6''$ and $|M'| \leqslant 5m + 1 + \sum_{i=1}^{k-2} in^k$.

“⇐”:

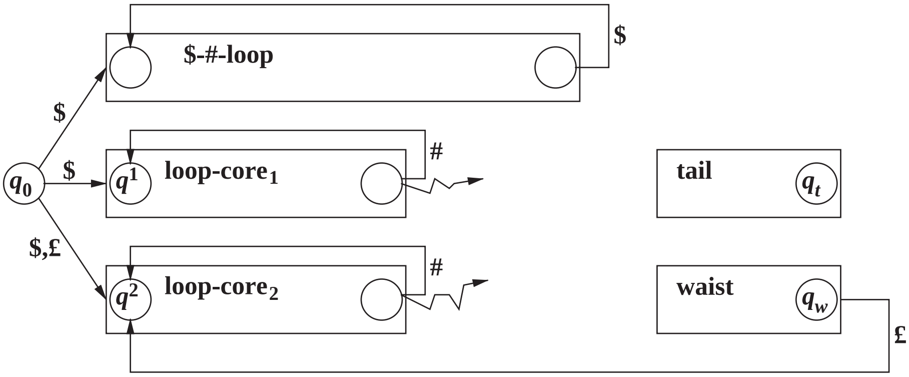

Fig. 2. The NFA($\beta = 3$) M accepting L''_6.

Let $M = (Q, \Sigma', \delta, q_0, F)$ be an NFA with branching $\beta_M = k$, $T(M) = L''_6$, and $|M| \leqslant 5m + 1 + \sum_{i=1}^{k-2} in^k$. We may assume that M is minimal. We have to construct an l-state DFA M' consistent with S and T. In consequence of the definition of L''_6, we can show that the nondeterministic moves of M have to start in q_0. Then, M can be converted to a 2-MDFA M'' such that $|M''| \leqslant 3m + 2t + 2$ (setting $t = m$) and $T(M'') = L'_5$. Owing to the proof of Lemma 3, we then can conclude that an l-state DFA M' consistent with S and T can be constructed:

(1) W.l.o.g. $S \neq \emptyset$. Let $w = \$\#^m w_1 \#^m$ with $w_1 \in S$ and $w' = w'_1 w'_2$ with $w'_1, w'_2 \in £\#^m L''_2 \#^m$ be two words in L''_6.

(2) *Claim*: M contains exactly one waist, one tail, two distinct loop-cores, and $k - 2$ \$-#-loops of length $n^k, 2n^k, \ldots, (k-2)n^k$.

A loop-core is defined as a sequence of states $q_1, q_2, \ldots, q_m, q_{m+1}$ such that $\delta(q_i, \#) = q_{i+1}$ for all $i \in \{1, 2, \ldots, m\}$ and q_{m+1} is nonaccepting, has outgoing edges, in particular a #-edge to q_1, but no outgoing £-edge. A \$-#-loop of length jn^k with $1 \leqslant j \leqslant k-2$ is defined as a sequence of states $q_1, q_2, \ldots, q_{jn^k}$ such that $\delta(q_i, \#) = q_{i+1}$ for all $i \in \{1, 2, \ldots, jn^k - 1\}$ and q_{jn^k} is accepting and has an outgoing \$-edge to q_1. Obviously, M contains at least one waist, one tail, and one loop-core. Consider the above word $w \in L'_5$. If we have exactly one loop-core, then there is a state $q \in \delta(q_0, \$) \cap \delta(q_0, £)$ and $\delta(q, \#^m w_1 \#^m) \cap F \neq \emptyset$. Hence, we have that $\delta(q_0, £\#^m w_1 \#^m) \cap F \neq \emptyset$ which is a contradiction. If M contains two loop-cores which are not distinct, then there is a state $q \in \delta(q_0, \$\#^i) \cap \delta(q_0, £\#^j)$ with $1 \leqslant i, j \leqslant m$ and $\delta(q, \#^{m-i} w_1 \#^m) \cap F \neq \emptyset$. Then $\delta(q_0, £\#^{j+m-i} w_1 \#^m) \cap F \neq \emptyset$ — contradiction. It is easy to see that the states of the tail and the waist are distinct from those of an \$-#-loop. Furthermore, the states of a loop-core and a \$-#-loop are distinct. By way of contradiction we assume that there exist $1 \leqslant j \leqslant k-2$ and a state $q \in \delta(q_0, \$\#^i) \cap \delta(q_0, \$\#^{jn^k-1}\$\#^{j'})$ and $\delta(q, \#^{i'} w_1 \#^m) \cap F \neq \emptyset$ with $1 \leqslant i \leqslant m$, $1 \leqslant j' \leqslant jn^k - 1$. Then $\delta(q_0, \$\#^{jn^k-1}\$\#^j \#^{i'} w_1 \#^m) \cap F \neq \emptyset$ which is a contradiction. We now show that \$-#-loops of different length have distinct states; hence M contains $k-2$ \$-#-loops of length $n^k, 2n^k, \ldots, (k-2)n^k$. Assume by way of contradiction that there is a state $q \in \delta(q_0, \$\#^{in^k-1}\$\#^{i'}) \cap \delta(q_0, \$\#^{jn^k-1}\$\#^{j'}) \neq \emptyset$

with $i \neq j$, $i' \leqslant in^k - 1$, $j' \leqslant jn^k - 1$, and $\delta(q, \#^{jn^k-1-j'}\$\#^{jn^k-1}) \cap F \neq \emptyset$. Then, $\$\#^{in^k-1}\$\#^{i'+jn^k-1-j'}\$\#^{jn^k-1} \in L''_6$ which is a contradiction. If M contains more than two loop-cores, more than one waist, more than one tail, or more than one \$-#-loop of the same length, then $|M|$ exceeds $5m+1+\sum_{i=1}^{k-2} in^k$, since M requires at least $2m$ states for waist and tail, $2m+2$ states for two loop-cores, $\sum_{i=1}^{k-2} in^k$ states for the $k-2$ \$-#-loops, one initial state, and at least m states for an additional waist, tail, loop-core, or \$-#-loop.

(3) W.l.o.g. we may assume that the word w will be accepted passing through loop-core$_1$ and the tail and w' will be accepted passing through loop-core$_2$ and the waist. Let $q_t \in \delta(q_0, w)$ and $q_w \in \delta(q_0, w'_1)$ denote the last states in the tail and the waist. By q^1 and q^2 we denote the states obtained after having read \$ and £ when M passes through the accepting computations of w and w'_1. Since w is accepted passing through loop-core$_1$, we can conclude that $\{q^2\} = \delta(q_w, \pounds)$ is the starting state of the loop in L''_3.

(4) All computations starting in $q^2 \in \delta(q_0, \pounds)$ and leading to an accepting state, thus computations of words in

$$\pounds\#^m(\#^{m+1})^* L''_2\#^m(\pounds\#^m(\#^{m+1})^* L''_2\#^m)^*$$

have branching one. This is obvious, since even one move with a branching greater than one would imply that M contains accepting computations with infinite branching due to the £-edge from q_w to q^2.

(5) The loop-cores and \$-#-loops contain no moves with branching greater than one, since due to their loops there would be computations with infinite branching.

(6) All computations starting in $\delta(q_0, \$)$ and leading to an accepting state, thus computations of words in $\$\#^m(\#^{m+1})^* S'\#^m$ with $S \subseteq S' \subseteq L''_1, L''_5$, and $\$\#^m(\#^{m+1})^* L''_2\#^m(\pounds\#^m(\#^{m+1})^* L''_2\#^m)^*$, have branching one. Due to (4) and (5) the moves with branching greater than one have to be located either in the states before entering the loop-core and the \$-#-loops, or in the states recognizing $S'\#^m$. First of all, we assume that all moves with branching greater than one start before entering the loop-core and the \$-#-loops. Then we can shift the branching to q_0: we remove any outgoing \$-edges from q_0 and insert $k-2$ \$-edges to the first states of the \$-#-loops and two \$-edges to loop-core$_1$ and loop-core$_2$. It follows that the modified automaton still recognizes L''_6, but there is at least one unnecessary state $q \in \delta(q_0, \$)$. Hence, M was not minimal which is a contradiction. We now assume that there is at least one move with branching 2 within the states recognizing $S'\#^m$. Then $L = \{\$\#^m(\#^{m+1})^*\} \cup L''_5$ must be recognized by an NFA with a branching of at most $\lfloor k/2 \rfloor$. Owing to Lemma 10 we know that a DFA for L needs at least $n^k + 2n^k + \cdots + (k-2)n^k + (k-2)n^k + 1 + (m+1)$ states. Analogous to the considerations in (2), one can show that every NFA accepting L with finite branching contains $k-2$ different \$-#-loops of length $n^k, 2n^k, \ldots, (k-2)n^k$, a loop-core of length $m+1$ and an initial state. In comparison with the minimal DFA, an NFA with finite branching can therefore achieve savings in size only through nondeterministic moves that start in states which are not part of a loop. Subtracting the loop-states from $n^k + 2n^k + \cdots + (k-2)n^k + (k-2)n^k + 1 + (m+1)$, there remain $(k-2)n^k + 1$ states. In [4] it is shown that the best possible reduction of states that an NFA with branching

i can achieve in comparison with the corresponding minimal DFA is at most the ith root of the size of the DFA. Hence, an NFA accepting L with branching $\lfloor k/2 \rfloor$ has at least $n^k + 2n^k + \cdots + (k-2)n^k + (m+1) + ((k-2)n^k + 1)^{1/\lfloor k/2 \rfloor}$ states. Since $((k-2)n^k + 1)^{1/\lfloor k/2 \rfloor} \geqslant n^2$, we have that $|M| \geqslant \sum_{i=1}^{k-2} in^k + n^2$ which is a contradiction to $|M| \leqslant 5m + 1 + \sum_{i=1}^{k-2} in^k = n + \sum_{i=1}^{k-2} in^k$.

It follows that $|\delta(q_0, \$)| > 1$. From q_0 we then have a \$-edge to q^1 and the first states of the $k-2$ \$-#-loops. Furthermore, we can assume to have an \$-edge to q^2. If there is no such edge, we can insert one without affecting the accepted language. We next remove the $k-2$ \$-#-loops and reduce the two loop-cores to cores by removing their #-loops. We then have an NFA with branching 2 with $3m + 2t + 1$ states ($t = m$) accepting L'_5. Now, we remove the \$-edge from q_0 to q^1 and we insert an additional state q'_0 which has an outgoing \$-edge to q^1. Thus, we have a 2-MDFA with $3m + 2t + 2$ states accepting L'_5. Owing to Lemma 3 we can construct an l-state DFA M' consistent with S and T. This shows the correctness of the reduction and thus the NP-hardness of the problem. □

Lemmas 9 and 11 imply the following theorem.

Theorem 13. NFA($\beta = k$) $\longrightarrow$ NFA($\beta = k$) *is* NP-*complete for* $k \geqslant 3$.

Corollary 14. *Let* $k \geqslant 2$ *and* $k' \geqslant 3$ *be constant integers. Then the problems* DFA $\longrightarrow$ NFA($\beta = k'$) *and* NFA($\beta = k$) $\longrightarrow$ NFA($\beta = k'$) *are* NP-*complete*.

Theorem 15. *The following problems, which are* PSPACE-*complete when arbitrary* NFAs *are considered, are solvable in polynomial time*:

(1) *Given two* NFAs M, M' *with* $\beta_M = k$ *and* $\beta_{M'} = k'$. *Is* $T(M) = T(M')$? *Is* $T(M) \subseteq T(M')$? *Is* $T(M) \subset T(M')$? *Is* $T(M) = \Sigma^*$?

(2) *Given any* NFA M *and an* NFA M' *with* $\beta_{M'} = k$. *Is* $T(M) \subseteq T(M')$?

Proof. Claim (1) results from the fact that NFAs with branching k can be efficiently converted to DFAs whose size is bounded by a polynomial in k (Lemma 8), and that the decidability questions are efficiently solvable for DFAs. To prove (2) we observe that $T(M) \subseteq T(M') \Leftrightarrow T(M) \cap \overline{T(M')} = \emptyset$. M' can be converted to a DFA of size $O(|M'|^k)$ and a DFA accepting $\overline{T(M')}$ has then $O(|M'|^k)$ states as well. Analogous to the construction of Lemma 8, we obtain an NFA $\widehat{M}$ accepting $T(M) \cap \overline{T(M')}$ and test its emptiness. We observe that the construction and the test can be performed in polynomial time. □

5. Conclusions

In this paper, we have shown that the minimization of finite automata equipped with a very small and fixed amount of nondeterminism is computationally hard. In particular, the minimization problems for DFAs with a fixed number of initial states as well as for NFAs with fixed finite branching have been proven NP-complete. Hence, even the slightest amount of nondeterminism makes minimization computationally intractable whereas

Table 1
Computational complexity results

	DFA	k-MDFA	NFA($\beta = k$)	NFA
Emptiness	P	P	P	P
Equivalence	P	P	P	PSPACE-complete
Inclusion	P	P	P	PSPACE-complete
Universality	P	P	P	PSPACE-complete
Minimization	P	NP-complete	NP-complete	PSPACE-complete

equivalence, inclusion, or universality questions preserve their efficient solutions. The results are summarized in Table 1.

Now the question arises whether there are extensions of the deterministic model at all that preserve polynomial time minimization algorithms. Two candidates result from our considerations. At first, the computational complexity of the problem NFA($\beta = k$) $\longrightarrow$ NFA($\beta = 2$) remains open. The problem is in NP, but the NP-hardness cannot be shown using the approach of Lemma 11. Moreover, the two constructions in Lemmas 3 and 11 present finite automata which are not unambiguous. It is currently unknown whether unambiguous k-MDFAs or unambiguous NFAs with branching k provide efficient minimization algorithms.

References

[1] M.R. Garey, D.S. Johnson, Computers and Intractability, W.H. Freeman and Co., San Francisco, 1979.

[2] A. Gill, L. Kou, Multiple-entry finite automata, J. Comput. System Sci. 9 (1974) 1–19.

[3] E.M. Gold, Complexity of automaton identification from given data, Inform. Control 37 (3) (1978) 302–320.

[4] J. Goldstine, C.M.R. Kintala, D. Wotschke, On measuring nondeterminism in regular languages, Inform. Comput. 86 (2) (1990) 179–194.

[5] M. Holzer, K. Salomaa, S. Yu, On the state complexity of k-entry deterministic finite automata, J. Automata, Languages Combin. 6 (4) (2001) 453–466.

[6] J.E. Hopcroft, An n log n algorithm for minimizing states in a finite automaton, in: Z. Kohavi (Ed.), Theory of Machines and Computations, Academic Press, New York, 1971, pp. 189–196.

[7] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison-Wesley, Reading MA, 1979.

[8] T. Jiang, B. Ravikumar, Minimal NFA problems are hard, SIAM J. Comput. 22 (6) (1993) 1117–1141.

[9] N.D. Jones, Space-bounded reducibility among combinatorial problems, J. Comput. System Sci. 11 (1) (1975) 68–85.

[10] N.D. Jones, E.Y. Lien, W.T. Laaser, New problems complete for nondeterministic log space, Math. Systems Theory 10 (1) (1976) 1–17.

[11] M. Kappes, Descriptional complexity of deterministic finite automata with multiple initial states, J. Automata, Languages Combin. 5 (3) (2000) 269–278.

[12] C.M.R. Kintala, D. Wotschke, Amounts of nondeterminism in finite automata, Acta Inform. 13 (2) (1980) 199–204.

[13] H. Leung, On finite automata with limited nondeterminism, Acta Inform. 35 (7) (1998) 595–624.

[14] A.R. Meyer, M.J. Fischer, Economy of descriptions by automata, grammars, and formal systems, IEEE Symp. Foundations of Computer Science, 1971, pp. 188–191.

[15] L. Stockmeyer, A.R. Meyer, Word problems requiring exponential time: preliminary report, Fifth Ann. ACM Symp. Theory of Computing, 1973, pp. 1–9.

[16] S. Yu, Regular languages, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Vol. 1, Springer, Berlin, Heidelberg, 1997, pp. 41–110.