

# Minimizing Finite Automata Is Computationally Hard

Andreas Malcher

Institut für Informatik, Johann Wolfgang Goethe Universität  
D-60054 Frankfurt am Main, Germany  
`malcher@psc.informatik.uni-frankfurt.de`

**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 [6]. On the other hand, it is known that unambiguous finite automata (UFAs) and nondeterministic finite automata (NFAs) can be algorithmically minimized too, but their minimization problems turn out to be NP-complete and PSPACE-complete, respectively [8]. 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 non-deterministic 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.

## 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. Due 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, 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. Sect. 5.2 in [15]). 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 [13]. Unfortunately, certain decidability questions, which are solvable in polynomial time for DFAs, are computationally hard for NFAs such as equivalence, inclusion, or universality [14,15]. 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 [2] 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 [5] and recently studied in [11] and [3]. The authors of the latter paper examine the minimization problem for MDFAs and prove its PSPACE-completeness. Their proof is a reduction from Finite State Automata Intersection [4] 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  $A_1, A_2, \dots, A_n$  are given. As is remarked in [4], 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 [3]. Thus, one might ask whether minimization of MDFAs with a fixed number of initial states is possible in polynomial time. We will show in Sect. 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.

## 2 Preliminaries and Definitions

Let  $\Sigma^*$  denote the set of all strings over the finite alphabet  $\Sigma$ ,  $\epsilon$  the empty string, and  $\Sigma^+ = \Sigma^* \setminus \{\epsilon\}$ . 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 [4]. Let  $L$  be a regular set; then  $size(L)$  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 [2] and [11].

A nondeterministic finite automaton over  $\Sigma$  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  $\delta$  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  $w = w_1 w_2 \dots w_n \in \Sigma^*$  is a sequence of moves  $\mu_1 \mu_2 \dots \mu_n$  where  $\mu_i = (q_{i-1}, w_i, q_i)$  with  $1 \leq i \leq n$ . It is an accepting computation if  $q_n \in F$ . The language accepted by  $M$  is  $T(M) = \{w \in \Sigma^* \mid \delta(q_0, w) \cap F \neq \emptyset\}$ .  $M$  is an (incomplete) deterministic finite automaton if  $|\delta(q, a)| \leq 1$  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  $\mu_1 \mu_2 \dots \mu_n$ ,  $n \geq 0$ , by setting  $\beta_M(\mu_1 \mu_2 \dots \mu_n) = \beta_M(\mu_1) \cdot \beta_M(\mu_2) \cdot \dots \cdot \beta_M(\mu_n)$ . For each word  $w \in T(M)$ , let  $\beta_M(w) = \min \beta_M(\mu_1 \mu_2 \dots \mu_n)$  where  $\mu_1 \mu_2 \dots \mu_n$  ranges over all accepting computations of  $M$  with input  $w$ . The branching  $\beta_M$  of the automaton  $M$  is  $\beta_M = \sup \{\beta_M(w) \mid w \in T(M)\}$ . The set of all NFAs with branching  $\beta = k$  is defined as  $\text{NFA}(\beta = k) = \{M \mid M \text{ is NFA and } \beta_M = k\}$ .

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  $T(M) = \{w \in \Sigma^* \mid \delta(Q_0, w) \cap F \neq \emptyset\}$ . 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  $\mathcal{A} \rightarrow \mathcal{B}$  denotes the problem of converting a type- $\mathcal{A}$  finite automaton to a minimal type- $\mathcal{B}$  finite automaton. Formally:

PROBLEM  $\mathcal{A} \rightarrow \mathcal{B}$

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,  $k \geq 2$  denotes a constant integer.

**Theorem 1.**  $k$ -MDFA  $\rightarrow k$ -MDFA is NP-complete.

*Proof.* The problem is in NP, since a  $k$ -MDFA  $M'$  with  $|M'| \leq l$  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  $M = (Q, \Sigma, \delta, \{q_0^1, q_0^2, \dots, q_0^k\}, F)$ ,  $M_1 = (Q, \Sigma, \delta, q_0^1, F)$ ,  $M_2 = (Q, \Sigma, \delta, q_0^2, F)$ ,  $\dots$ ,  $M_k = (Q, \Sigma, \delta, q_0^k, F)$ . Then  $T(M_1) \cup T(M_2) \cup \dots \cup T(M_k) = T(M)$  and we construct a DFA  $\hat{M}$  as the Cartesian product of  $M_1, M_2, \dots, M_k$  accepting  $T(M_1) \cup \dots \cup T(M_k)$  in the usual way. A DFA  $\hat{M}'$  can be constructed from  $M'$  analogously. The time complexity of the inequivalence problem of two DFAs is in  $\text{NLOGSPACE} \subseteq \text{P}$  [9]. Hence  $T(\hat{M}) = T(\hat{M}')$  can be tested in polynomial time.

The NP-hardness of the problem will be shown by reduction from the Minimum Inferred DFA problem. 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  $\Sigma$ , two finite subsets  $S, T \subset \Sigma^*$ , integer  $l$ .

**QUESTION** Is there an  $l$ -state DFA that accepts a language  $L$  such that

$$S \subseteq L \text{ and } T \subseteq \Sigma^* \setminus L?$$

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 consistent with  $S$  and  $T$  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 consistent with  $S$  and  $T$  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  $\mathcal{L}$  be symbols not in  $\Sigma$ . Let  $\Sigma' = \Sigma \cup \{\#, \$, \mathcal{L}\}$ ,  $m = l + \text{size}(\overline{T} \cap \overline{S})$ , and  $t = \max(k, m)$ .

$$\begin{aligned} L_1 &= \overline{T}, \\ L_2 &= \overline{T} \cap \overline{S}, \\ L_3 &= \{\$, \mathcal{L}\} \#^t L_2 \#^m (\mathcal{L} \#^t L_2 \#^m)^*, \\ L_4 &= \$ \#^t L_1 \#^m, \\ L_5 &= L_3 \cup L_4. \end{aligned}$$

Following [8], it is easy to show the following lemma:

**Lemma 1.** *Let  $L$  be regular and  $M'$  a DFA consistent with  $S$  and  $T$ .*

1.  $\text{size}(\$ \#^t L \#^m) = \text{size}((\$ \#^t L \#^m)^+) = t + m + 1 + \text{size}(L)$
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.* The 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.  $\square$

We now present the reduction. 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  $\{q_0^3, \dots, q_0^k\}$  not in  $Q_1 \cup Q_2$ . Then we can construct a  $k$ -MDFA  $M = (Q_1 \cup Q_2 \cup \{q_0^3, \dots, q_0^k\}, \Sigma', \delta, \{q_0^1, q_0^2, \dots, q_0^k\}, F_1 \cup F_2)$ . 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  $i \in \{3, \dots, k\}$ . 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  $m'$ . We next show the correctness of the reduction.

Claim: 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  $|M''| \leq l$ . 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  $|M_1| = t + m + 1 + \text{size}(L_2) = t + 2m + 1 - l$ ,  $|M_2| \leq t + m + l + 1$  and therefore  $|M_1| + |M_2| \leq 3m + 2t + 2$ . Considering the two symbols  $\$, \mathcal{L}$  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  $|M'| \leq 3m + 2t + k$  and  $T(M') = L_5$ .

“ $\Leftarrow$ ”: (sketch)

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  $|M| \leq 3m + 2t + k$ . 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  $|M_1| + |M_2| \leq 3m$  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.

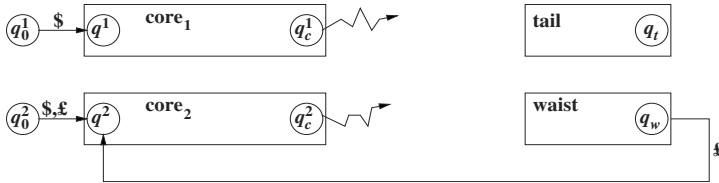


Fig. 1. The modified  $k$ -MDFA  $M$

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  $w'_1, w'_2 \in \mathcal{L} \#^t L_2 \#^m$  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. 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  $\mathcal{L}$ -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 non-accepting and has outgoing edges, but no outgoing  $\mathcal{L}$ -edge. It can be shown that  $M$  contains exactly one waist, one tail and two distinct cores.

W.l.o.g. we may assume that  $w$  will be accepted from  $q_0^i$  passing through core<sub>1</sub> and the tail and  $w'$  will be accepted from  $q_0^j$  passing through core<sub>2</sub> 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  $q^2 = \delta(q_0^j, \mathcal{L})$  denote the first states of core<sub>1</sub> and core<sub>2</sub>. By  $q_c^1 = \delta(q_0^i, \$\#^t)$  and  $q_c^2 = \delta(q_0^j, \mathcal{L}\#^t)$  we denote the last states of core<sub>1</sub> and core<sub>2</sub>. Since  $w$  is accepted passing through core<sub>1</sub>, we can conclude that  $q^2 = \delta(q_w, \mathcal{L})$  is the starting state of the loop.

4. It can be shown that all initial states have no incoming edges.
5. It can be shown that  $\delta(q^1, \#^t S \#^m) \subseteq F$  and  $\delta(q^2, \#^t S \#^m) \cap F = \emptyset$ .
6. It can be shown that  $\delta(q^2, \#^t L_2 \#^m (\mathcal{L} \#^t L_2 \#^m)^*) \subseteq F$ .
7.  $M$  can be modified to the form depicted in Fig. 1 on page 390. (The initial states  $q_0^3, \dots, q_0^k$  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:  $q_0^1 \xrightarrow{\$} q^1$ ,  $q_0^2 \xrightarrow{\$, \mathcal{L}} q^2$ , and  $q_0^i \xrightarrow{\$, \mathcal{L}} q^2$  for  $i \in \{3, \dots, k\}$ . We observe that due to 4., 5., and 6. 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  $q_0^1$ .

8. 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:  $\mathcal{E}(q_0^1) = \{q \in Q \mid \exists x, x' \in (\Sigma')^* : \delta(q_0^1, x) = q \wedge \delta(q, x') \in F\}$ .  $\mathcal{E}(q_0^2)$  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  $\delta(p, x') \in F$ . Due to 7., we may assume that  $x$  starts with  $\mathcal{L}$ . 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  $\mathcal{L}$ -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 6. and 7., 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  $\mathcal{E}(q_0^1) \cap \mathcal{E}(q_0^2) = \emptyset$ .
9. Since the sets of reachable states are distinct, we 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 5. and 7., we know that  $L_4 \supseteq T(M'_1) \supseteq \$ \#^t S \#^m$  and  $T(M'_2) = L_3$ . Furthermore,  $|M'_1| + |M'_2| \leq 3m + 2t + 2$ , since  $Q'_1 \cap Q'_2 = \emptyset$ .
  10. Starting from  $M'_1$  we define another DFA  $M_1$  by removing  $q_0^1$  and the first  $t$  states of  $\text{core}_1$ . 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  $\text{core}_2$ . We define  $q_c^2$  as new initial state. The  $\mathcal{L}$ -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  $\sigma$ -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,  $|M_1| + |M_2| \leq 3m$ .
  11. We have  $|M_2| = \text{size}((L_2 \#^m)^+) = m + \text{size}(\overline{T} \cap \overline{S}) = 2m - l$  and therefore  $|M_1| \leq 3m - |M_2| = 3m - 2m + l = m + l$ . Removing the tail in  $M_1$  yields an  $l$ -state DFA  $M'$  consistent with  $S$  and  $T$ .  $\square$

**Corollary 1.** *Let  $k, k' \geq 2$  be two constant numbers. Then  $\text{DFA} \rightarrow k\text{-MDFA}$  and  $k\text{-MDFA} \rightarrow k'\text{-MDFA}$  are NP-complete.*

The following theorem is a simple observation of the fact that  $k$ -MDFAs can be efficiently converted to DFAs whose size is bounded by a polynomial in  $k$ , and that the below-mentioned decidability questions are efficiently solvable for DFAs.

**Theorem 2.** *Let  $M$  be a  $k$ -MDFA and  $M'$  be a  $k'$ -MDFA. Then the following problems are solvable in polynomial time. Is  $T(M) = T(M')$ ? Is  $T(M) \subseteq T(M')$ ? Is  $T(M) \subset T(M')$ ? Is  $T(M) = \Sigma^*$ ?*

## 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  $\beta = k$  ( $\text{NFA}(\beta = k)$ ) is NP-complete for  $k \geq 3$ .

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

*Proof.* Let  $i$  be an integer. We consider the language

$$T_i(M) = \{w \in \Sigma^* \mid \exists \text{ accepting computation } \pi \text{ of } M \text{ of } w \text{ with } \beta(\pi) \leq i\}.$$

In [2] it is shown that a DFA  $M_i$  accepting  $T_i(M)$  can be effectively constructed. It can be observed that the construction can be done in time polynomially bounded in  $|M|$  and that the resulting DFA has size  $O(|M|^k)$ .

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

$$\beta_M \leq k \Leftrightarrow T(M) \setminus T_k(M) = \emptyset \Leftrightarrow T(M) \cap \overline{T_k(M)} = \emptyset \Leftrightarrow T(M) \cap T(M'_k) = \emptyset$$

Since  $M$  is an NFA and  $M'_k$  is a DFA, we can construct, in polynomial time, an NFA  $\hat{M}$  of size  $O(|M| \cdot |M'|^k)$  as the Cartesian product of  $M$  and  $M'_k$  accepting  $T(M) \cap T(M'_k)$ . The non-emptiness of  $T(\hat{M})$  can be tested in  $\text{NLOGSPACE} \subseteq \text{P}$  [10]. If  $T(\hat{M}) \neq \emptyset$ , then  $\beta_M > k$ . If  $T(\hat{M}) = \emptyset$ , then we know that  $\beta_M \leq k$ . To find out whether  $\beta_M = k$ , we construct  $T_{k-1}(M)$  if  $k-1 \geq 1$ . 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  $\beta_M < k$ .  $\square$

**Theorem 3.**  $\text{NFA}(\beta = k) \longrightarrow \text{NFA}(\beta = k)$  is NP-complete for  $k \geq 3$ .

*Proof.* We first show that the problem is in NP. To this end we determine non-deterministically an NFA  $M'$  with  $|M'| \leq l$ . Due to Lemma 2, we can test whether  $M'$  has branching  $k$  in polynomial time. We next convert  $M$  and  $M'$  to  $k$ -MDFAs  $\hat{M}$  and  $\hat{M}'$  with at most  $k|M| + 1$  and  $k|M'| + 1$  states applying the construction presented in [11]. The equality of  $T(\hat{M})$  and  $T(\hat{M}')$  can then be tested in polynomial time analogous to the considerations of Theorem 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  $L_5$ . Due to Theorem 1, we then can construct an appropriate DFA consistent with  $S$  and  $T$ . An obvious, but essential observation for NFAs with finite branching is that a move with a branching  $\beta > 1$  cannot be located within a loop, since 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  $n = 5m + 1$ . In addition to the previous definitions we define:

$$\begin{aligned} L'_3 &= \{\$, \mathcal{L}\} \#^m (\#^{m+1})^* L_2 \#^m (\mathcal{L} \#^m (\#^{m+1})^* L_2 \#^m)^*, \\ L'_4 &= \$ \#^m (\#^{m+1})^* L_1 \#^m, \\ L^i &= \{(\$ \#^{in^k-1})^+\} \quad (1 \leq i \leq k-2), \\ L'_5 &= L^1 \cup L^2 \cup \dots \cup L^{k-2}, \\ L'_6 &= L'_3 \cup L'_4 \cup L'_5. \end{aligned}$$

**Lemma 3.** Let  $L$  be regular and  $M'$  a DFA consistent with  $S$  and  $T$ .

$$1. \text{size}(\$ \#^m (\#^{m+1})^* L \#^m) = \text{size}((\$ \#^m (\#^{m+1})^* L \#^m)^+) = 2m + 1 + \text{size}(L)$$



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.  $\text{size}(\{\$ \#^m (\#^{m+1})^*\} \cup L'_5) \geq n^k + 2n^k + \dots + (k-2)n^k + (k-2)n^k + 1 + (m+1)$

*Proof.* The claims 1., 2., and 3. can be shown analogously to those of Lemma 1. Claim 4. is obvious. The proof of 5. is not difficult, but lengthy and is omitted here.  $\square$

We now present the reduction. 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, let  $M_i = (Q_i, \Sigma', \delta_i, q_0^i, F_i)$ ,  $3 \leq i \leq k$  be  $k-2$  minimal DFAs accepting  $L^1, L^2, \dots, L^{k-2}$ . W.l.o.g. we may assume that  $Q_1, Q_2, \dots$ , and  $Q_k$  are pairwise distinct. We observe that for  $3 \leq i \leq k$  the states  $q_0^i$  have no incoming edges and only one outgoing edge to a non-trap state, namely a  $\$$ -edge. Moreover,  $q_0^1$  has no incoming edges and only two outgoing edges to non-trap states, namely a  $\$$ -edge and a  $\mathcal{L}$ -edge. We remove  $q_0^1$  from  $M_1$  and  $q_0^i$  from  $M_i$  for  $3 \leq i \leq k$  and construct an NFA  $M = ((Q_1 \setminus \{q_0^1\}) \cup Q_2 \cup (Q_3 \setminus \{q_0^3\}) \cup \dots \cup (Q_k \setminus \{q_0^k\}), \Sigma', \delta, q_0^2, F_1 \cup F_2 \cup \dots \cup F_k)$ . For  $\sigma \in \Sigma'$  and  $1 \leq i \leq k$  we define  $\delta(q, \sigma) = \delta_i(q, \sigma)$  if  $q \in Q_i$ . Furthermore,  $\delta(q_0^2, \$) = \delta_1(q_0^1, \$)$ ,  $\delta(q_0^2, \mathcal{L}) = \delta_1(q_0^1, \mathcal{L})$ , and  $\delta(q_0^2, \$) = \delta_i(q_0^i, \$)$  for  $3 \leq i \leq k$ . 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  $m'$ . We next show the correctness of the reduction.

Claim: 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.

“ $\Rightarrow$ ”:

Let  $M''$  be a DFA consistent with  $S$  and  $T$  and  $|M''| \leq l$ . 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,  $M_3, \dots, M_k$  are minimal DFAs accepting  $L^1, \dots, L^{k-2}$ . Analogous to the proof of Theorem 1 and the above considerations we can construct an NFA  $M'$  with  $\beta_{M'} = k$  such that  $T(M') = L'_6$  and  $|M'| \leq 5m + 1 + \sum_{i=1}^{k-2} in^k$ .

“ $\Leftarrow$ ”: (sketch)

Let  $M = (Q, \Sigma', \delta, q_0, F)$  be an NFA with branching  $\beta_M = k$  such that  $T(M) = L'_6$  and  $|M| \leq 5m + 1 + \sum_{i=1}^{k-2} in^k$ . 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  $|M''| \leq 3m + 2t + 2$  (setting  $t = m$ ) and  $T(M'') = L_5$ . Due to the proof of Theorem 1, 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  $w'_1, w'_2 \in \mathcal{L} \#^m L_2 \#^m$  be two words in  $L'_6$ .

2. A loop-core is defined as a sequence of states  $q_1, q_2, \dots, q_m, q_{m+1}$  such that  $\delta(q_i, \#) = q_{i+1}$  for all  $i \in \{1, 2, \dots, m\}$  and  $q_{m+1}$  is non-accepting, has outgoing edges, in particular a  $\#$ -edge to  $q_1$ , but no outgoing  $\mathcal{L}$ -edge. A  $\$$ - $\#$ -loop of length  $jn^k$  with  $1 \leq j \leq k-2$  is defined as a sequence of states  $q_1, q_2, \dots, q_{jn^k}$  such that  $\delta(q_i, \#) = q_{i+1}$  for all  $i \in \{1, 2, \dots, jn^k - 1\}$  and  $q_{jn^k}$  is accepting and has an outgoing  $\mathcal{L}$ -edge to  $q_1$ .

It can be shown that  $M$  contains exactly one waist, one tail, two distinct loop-cores, and  $k-2$   $\$$ - $\#$ -loops of length  $n^k, 2n^k, \dots, (k-2)n^k$ .

W.l.o.g. we may assume that  $w$  will be accepted passing through loop-core<sub>1</sub> and the tail and  $w'$  will be accepted passing through loop-core<sub>2</sub> 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  $\mathcal{L}$  when  $M$  passes through the accepting computations of  $w$  and  $w'_1$ . Since  $w$  is accepted passing through loop-core<sub>1</sub>, we can conclude that  $\{q^2\} = \delta(q_w, \mathcal{L})$  is the starting state of the loop in  $L'_3$ .

3. All computations starting in  $q^2 \in \delta(q_0, \mathcal{L})$  and leading to an accepting state, thus computations of words in  $\mathcal{L}\#^m(\#^{m+1})^*L_2\#^m(\mathcal{L}\#^m(\#^{m+1})^*L_2\#^m)^*$ , have branching 1. 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  $\mathcal{L}$ -edge from  $q_w$  to  $q^2$ .
4. The loop-cores and the  $\$$ - $\#$ -loops contain no moves with branching greater than one, since due to their loops there would be computations with infinite branching.
5. 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  $\$ \#^m(\#^{m+1})^*L_2\#^m(\mathcal{L}\#^m(\#^{m+1})^*L_2\#^m)^*$ , have branching 1.

Due to 3. and 4. 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  $\mathcal{L}$ -edges from  $q_0$  and insert  $k-2$   $\mathcal{L}$ -edges to the first states of the  $\$$ - $\#$ -loops and two  $\mathcal{L}$ -edges to loop-core<sub>1</sub> and loop-core<sub>2</sub>. 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  $\lfloor \frac{k}{2} \rfloor$ . Due to Lemma 3 we know that a DFA for  $L$  needs at least  $n^k + 2n^k + \dots + (k-2)n^k + (k-2)n^k + 1 + (m+1)$  states. Analogous to the (here omitted) considerations in 2., one can show that every NFA accepting  $L$  with finite branching contains  $k-2$  different  $\$$ - $\#$ -loops of length  $n^k, 2n^k, \dots, (k-2)n^k$ , 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  $n^k + 2n^k + \dots + (k-2)n^k + (k-2)n^k + 1 + (m+1)$ , there remain  $(k-2)n^k + 1$  states. In [2] 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  $i$ -th root of the size of the DFA. Hence an NFA accepting  $L$  with branching  $\lfloor \frac{k}{2} \rfloor$  has at least  $n^k + 2n^k + \dots + (k-2)n^k + (m+1) + ((k-2)n^k + 1)^{1/\lfloor \frac{k}{2} \rfloor}$  states. Since  $((k-2)n^k + 1)^{1/\lfloor \frac{k}{2} \rfloor} \geq n^2$ , we have that  $|M| \geq \sum_{i=1}^{k-2} in^k + n^2$  which is a contradiction to  $|M| \leq 5m + 1 + \sum_{i=1}^{k-2} in^k = n + \sum_{i=1}^{k-2} in^k$ .

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 a  $\$$ -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  $L_5$ . Due to Theorem 1 we can construct an  $l$ -state DFA  $M'$  consistent with  $S$  and  $T$ .  $\square$

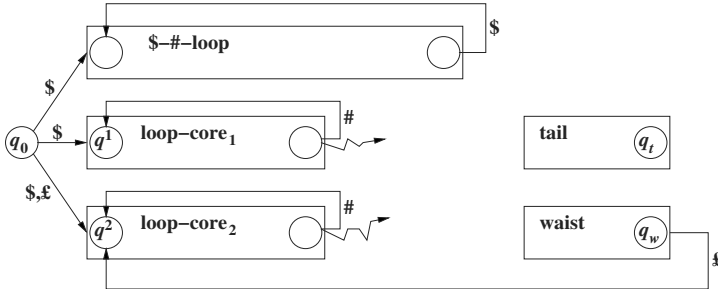


Fig. 2. The NFA( $\beta = 3$ )  $M$  accepting  $L'_6$ .

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

**Theorem 4.** 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.* The proof of 1. is similar to the proof of Theorem 2. To prove 2., we observe that  $T(M) \subseteq T(M') \Leftrightarrow T(M) \cap \overline{T(M')} = \emptyset$ . The claim can then be shown using a similar argumentation as in Lemma 2.  $\square$

## 5 Conclusions

We have shown that the minimization of finite automata equipped with a very small and fixed amount of nondeterminism is computationally hard. Hence 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  $\text{NFA}(\beta = k) \rightarrow \text{NFA}(\beta = 2)$  remains open. Moreover, the two constructions in Theorem 1 and Theorem 3 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. Gold, E.M.: Complexity of automaton identification from given data. *Information and Control* **37:3** (1978) 302–320
2. Goldstine, J., Kintala, C.M.R., Wotschke, D.: On measuring nondeterminism in regular languages. *Information and Computation* **86:2** (1990) 179–194
3. Holzer, M., Salomaa, K., Yu, S.: On the state complexity of  $k$ -entry deterministic finite automata. *Journal of Automata, Languages and Combinatorics* **6:4** (2001) 453–466
4. Garey, M.R., Johnson, D.S.: *Computers and Intractability*. W.H. Freeman and Co., San Francisco (1979)
5. Gill, A., Kou, L.: Multiple-entry finite automata. *Journal of Computer and System Sciences* **9** (1974) 1–19
6. Hopcroft, J.E.: An  $n \log n$  algorithm for minimizing states in a finite automaton. In: Kohavi, Z. (ed.): *Theory of machines and computations*. Academic Press, New York (1971) 189–196
7. Hopcroft, J.E., Ullman, J.D.: *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading MA (1979)
8. Jiang, T., Ravikumar, B.: Minimal NFA problems are hard. *SIAM Journal on Computing* **22:6** (1993) 1117–1141
9. Jones, N.D., Lien, E.Y., Laaser, W.T.: New problems complete for nondeterministic log space. *Mathematical Systems Theory* **10:1** (1976) 1–17
10. Jones, N.D.: Space-bounded reducibility among combinatorial problems. *Journal of Computer and System Sciences* **11:1** (1975) 68–85
11. Kappes, M.: Descriptive complexity of deterministic finite automata with multiple initial states. *Journal of Automata, Languages and Combinatorics* **5:3** (2000) 269–278
12. Kintala, C.M.R., Wotschke, D.: Amounts of nondeterminism in finite automata. *Acta Informatica* **13:2** (1980) 199–204
13. Meyer, A.R., Fischer, M.J.: Economy of descriptions by automata, grammars, and formal systems. *IEEE Symposium on Foundations of Computer Science* (1971) 188–191
14. Stockmeyer, L., Meyer, A.R.: Word problems requiring exponential time: preliminary report. *Fifth Annual ACM Symposium on Theory of Computing* (1973) 1–9
15. Yu, S.: Regular languages. In: Rozenberg, G., Salomaa, A. (eds.): *Handbook of Formal Languages Vol. 1*. Springer-Verlag, Berlin Heidelberg (1997) 41–110