A Multivariate Analysis of Some DFA Problems*

Henning Fernau[1], Pinar Heggernes[2], and Yngve Villanger[2]

[1] Department of Informatics, University of Trier, Germany
fernau@uni-trier.de
[2] Department of Informatics, University of Bergen, Norway
{pinar.heggernes,yngve.villanger}@ii.uib.no

Abstract. We initiate a multivariate analysis of two well-known NP-hard decision problems on DFAs: the problem of finding a short synchronizing word and that of finding a DFA on few states consistent with a given sample of the intended language and its complement. For both problems, we study natural parameterizations and classify them with the tools provided by Parameterized Complexity. Somewhat surprisingly, in both cases, rather simple FPT algorithms can be shown to be optimal, mostly assuming the (Strong) Exponential Time Hypothesis.

Keywords: Deterministic finite automata, NP-hard decision problems, synchronizing word, consistency problem.

1 Introduction

Multivariate analysis of computationally hard problems [16] tries to answer the question what actually makes a problem hard by systematically considering so-called natural parameters that can be singled out in an instance or in some target structure. In problems dealing with finite automata, such parameters could be the size of the input alphabet, or the number of states. For instance, if some hardness reduction produces or requires automata with large input alphabets, then this proof does not reveal much if only binary input alphabets are of interest. Parameterized Complexity offers tools to tell if hardness result could be expected when fixing, say, the alphabet size. In other words, we target the question what aspects of our problem cause it to become hard. As the possible choices of parameters are very abundant, we consider our paper rather as the starting point of this line of research within the theory of finite automata. Only limited multivariate analysis research has been undertaken so far on finite automata problems, NFA minimization being one exception [5], Mealy machines with census requirements another one [18], offering ample ground to work on.

In this paper, we study the parameterized complexity of two problems related to finite automata: the problem of finding a shortest synchronizing word in a deterministic finite automaton (DFA) and that of finding the smallest DFA consistent with a given sample consisting of positive and of negative examples of the intended language. Both problems have a long history, dating back to the very

* This work is supported by the Norwegian and the German Research Councils.

A.-H. Dediu, C. Martín-Vide, and B. Truthe (Eds.): LATA 2013, LNCS 7810, pp. 275–286, 2013.

beginning of automata theory, and both questions have found many practical applications.

The Synchronizing Word (SW) problem is the following one: Given a DFA $A = (S, I, \delta, s_0, F)$ with state set S, input alphabet I, transition function $\delta : S \times I \to S$, initial state s_0 and set of final states F, together with some integer $k \geq 0$, decide if there exists a synchronizing word of length at most k for A. Here, a *synchronizing word* is a string $x \in I^*$ such that there exists some state $s_f \in S$ such that, for any start state $s \in S$, $\delta^*(s, x) = s_f$. Hence, a synchronizing word enables to reset an automaton to some well-defined state, wherever it may start. Therefore, it is also known as a *reset sequence* and also under many other different names. This notion and several related ones that we are going to discuss draw their practical motivation from testing circuits and automata, cf. [26,27,29].

Eppstein showed [15] that SW is NP-complete. Later, Berlinkov proved in [4] that the related optimization problem MIN-SW cannot belong to APX under some complexity assumptions. Walker [33] observed that Eppstein's reduction not only works when starting from 3-SAT, but also when using SAT. This will be useful for our purposes.

We show that SW is W[2]-hard when parameterized by the natural parameter k. This provides an alternative proof of the mentioned NP-hardness result. As our reduction is from Hitting Set, it also shows that MIN-SW cannot be approximated even up to some logarithmic factor depending on the size of the input alphabet [27,1]. This is not the same as Berlinkov's result, as he focuses on small alphabet sizes. It would be interesting to know if SW with parameter k actually belongs to W[2]. Otherwise, it might be one of the few natural problems known to be placed in higher levels of the W-hierarchy, cf. the discussions in [8].

The related combinatorial questions are nicely reported and reviewed in [29,32]. The most important question in that area is settling or disproving Černý's conjecture [7] that each t-state DFA that has a synchronizing word has also one of length at most $\frac{t(t-1)}{2}$. Currently best upper bounds are of size cubic in t, the record holder being [31].

As our second problem on automata, we consider the DFA Consistency problem. Here, the input consists in an alphabet I, two finite disjoint sets of words $X^+, X^- \subseteq I^*$, and an integer t. The question is to decide if there exists some DFA A with at most t states such that $X^+ \subseteq L(A)$ and $X^- \cap L(A) = \emptyset$. This problem was extensively studied in the area of Algorithmic Learning Theory, especially in Grammatical Inference, as it is central to the model of *Learning in the Limit*, initiated by Gold's seminal work [21]. As the problem was shown to be NP-hard [2,22,23] and even hard to approximate [28], several heuristics and translations to other problems were proposed. In our context, reductions to coloring problems seem to be most relevant, see [10]. We can underline our approach with a quotation from [23, p. 139]: *Alternative proofs of the hardness of the consistency problem would be of help, in order to better understand what is really hard.* This can be seen as a quest for a multivariate analysis for DFA Consistency, as we commence in this paper.

Notice that DFA Consistency can be seen as an implementation of Occam's razor in the sense that the shortest explanation (in terms of DFA) for the given sample is aimed at. This principle can also lead to computationally hard problems in the context of regular languages if only positive examples are given, taking into account questions concerning the representability or coding of the sample words. This kind of question was investigated in [17] from the viewpoint of Parameterized Complexity. Clearly, the consistency problem can be also asked for other types of automata and grammars. As long as the universal language I^* has a simple representation in the corresponding class, the consistency problem is trivial if $X^- = \emptyset$. However, there are also interesting classes of languages where this is not the case. For instance, it has been shown in [9] that this type of consistency problem is W[2]-hard for a whole range of categorial grammar families. Further Parameterized Complexity results for algorithmic learning problems can be found in [3,13,30]. There are also not so many papers dealing with Parameterized Complexity classification of questions on finite automata, cf. [18,34] in this context.

In this extended abstract, we had to omit most of the proofs. A long version of the paper can be obtained from the authors on request.

2 Preliminaries

A graph $G = (V, E)$ is undirected and unweighted, with vertex set V and edge set E. Given a subset $X \subseteq V$, the subgraph of G induced by X is denoted by $G[X]$. A vertex subset X is an *independent set* if $G[X]$ has no edges. A partition of V into $V_1, V_2, \ldots, V_r$ is called a *proper r-coloring* if $G[V_i]$ is an independent set for $1 \leq i \leq r$.

A *deterministic finite automaton (DFA) A* is a tuple (S, I, δ, s_0, F), where S is the set of states, I is the input alphabet, $\delta : S \times I \to S$ is the transition function, s_0 is the initial state, and F is the set of final states. We will use $t = |S|$.

For an introduction to the by now well established field of Parameterized Complexity, we refer the reader to the textbook [14]. Here we give a short and informal overview. A decision problem is said to be a *parameterized* problem if its input can be partitioned into a *main part J* and a *parameter P*. A parameterized problem with main input size $|J|$ and parameter size $|P|$ is said to be *fixed-parameter tractable* (FPT) if it can be solved by an algorithm with running time $O^*(f(|P|))$, where f is a computable function depending only on P and not on J, and the O^*-notation suppresses all factors that are polynomial in $|J|$. It is well-known that a parameterized problem is FPT if and only if it has a *kernel*, meaning that there is a polynomial time algorithm that produces an equivalent instance J' of size $|J'| \in O(g(|P|))$, where g is again a function depending only on P. If g is a polynomial function, then the problem is said to admit a *polynomial kernel*. Whether or not a fixed-parameter tractable problem admits a polynomial kernel is a broad subfield of Parameterized Complexity.

In the same way as NP-hardness of a decision problem indicates that we cannot expect a polynomial time algorithm, there exists a hierarchy of complexity classes above FPT, and showing that a parameterized problem is hard for any of these classes makes it unlikely to be FPT. The main classes are: FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \ldots \subseteq$ W[P] $\subset$ XP, where XP is the class of parameterized problems that are solvable in time $O(|J|^{h(|P|)})$ for some function h. Consequently, if the problem is NP-hard when the parameter size is bounded by a constant, then it is not even likely to belong to XP.

Next we define some well-known problems and complexity theoretical assumptions, which will be useful for proving hardness results and lower bounds.

Problem: r-SAT
Input: A boolean CNF formula ϕ on n variables and m clauses, where each clause contains at most r literals.
Question: Is there a truth assignment that satisfies ϕ?

If there is no bound on the number of literals that a clause can contain, then we simply refer to the problem as SAT.

One common way to argue for the unlikeness of a subexponential algorithm is to use the *Exponential Time Hypothesis* (ETH). By the observation that each variable is used at least once, and by the Sparsification Lemma [25], ETH can be expanded to:

Exponential Time Hypothesis (ETH)[25]: There is a positive real s such that 3-SAT instances on n variables and m clauses cannot be solved in time $2^{sn}(n+m)^{O(1)}$.
Most useful is the following corollary: There is a real $s' > 0$ such that 3-SAT instances on m clauses cannot be solved in time $2^{s'm}(n+m)^{O(1)}$.

A slightly stronger assumption is the following one.
Strong Exponential Time Hypothesis (SETH)[25][6]: SAT cannot be solved in time $2^{sn}(n+m)^{O(1)}$ for $s < 1$.

In order to argue for the unlikeliness of polynomial kernels we use:

Proposition 1 ([19]). SAT*, parameterized by the number* n *of variables, does not have a kernel that is polynomial in* n *unless* NP $\subseteq$ coNP/poly*.*

We make use of the following NP-complete problems in our reductions [14,20].

Problem: r-Coloring
Input: A graph G on n vertices and m edges.
Question: Is there a proper r-coloring of G?

Problem: Hitting set
Input: A family $\mathcal{F}$ of sets over a universe $\mathcal{U}$ and an integer k.
Parameter: k
Question: Does there exist a set $\mathcal{S} \subset \mathcal{U}$ such that $|\mathcal{S}| \leq k$ and $F \cap \mathcal{S} \neq \emptyset$ for each $F \in \mathcal{F}$?

3 Shortest Synchronizing Word

We consider different parameterizations of the following problem.

Problem: Synchronizing Word (SW)
Input: A DFA $A = (S, I, \delta, s_0, F)$ and an integer k.
Question: Is there a k-synchronizing word for A?

The most natural parameter is of course k. As we will show, the problem is W[2]-hard with this parameter. We will consider other natural parameters as well; these are $t = |S|$ and $|I|$. Note that s_0 and F are simply irrelevant for SW, and $|\delta|$ is simply t times $|I|$. Table 1 summarizes the results of this section.

Table 1. A summary of the results of this section. In addition, we show that the two given running time upper bounds are tight in the sense that we cannot expect to solve SW in time $O^*(2^{o(t)})$ or in time $O^*((|I| - \epsilon)^k)$ for any $\epsilon > 0$.

Parameter	Parameterized Complexity	Polynomial Kernel
k	W[2]-hard	—
$\lvert I\rvert$	NP-complete for $\lvert I\rvert = 2$	—
k and $\lvert I\rvert$	FPT with running time $O^*(\lvert I\rvert^k)$	Not unless NP $\subseteq$ coNP/poly
t	FPT with running time $O^*(2^t)$	Open

For the first hardness result, we first need the following lemma.

Lemma 2. *Given a* Hitting Set *instance with family $\mathcal{F}$ and universe $\mathcal{U}$, a DFA $A = (S, I, \delta, s_0, F)$ can be constructed in time $O(|\mathcal{F}||\mathcal{U}|)$, such that $|S| = |\mathcal{F}| + k + 1$, $|I| = |\mathcal{U}|$, and A has a k-synchronizing word iff $\mathcal{F}$ has a hitting set of size k.*

Theorem 3. Synchronizing Word *is W[2]-hard, parameterized with k.*

If we instead parameterize SW with $|I|$, we obtain that the problem is not even in XP. For that result, we first need the following.

Proposition 4 ([15],[33]). *Given a* SAT *formula ϕ with n variables and m clauses, a DFA $A = (S, I, \delta, s_0, F)$ can be constructed in $O(nm)$ time, such that $|S| = nm + m + 1$, $|I| = 2$, and A has an n-synchronizing word if and only if ϕ has a satisfying truth assignment.*

Theorem 5. Synchronizing Word *is NP-complete when $|I| = 2$.*

As neither parameter k nor parameter $|I|$ is useful for fixed-parameter tractability, a natural next step is to use both k and $|I|$ as a combined parameter.

Theorem 6. Synchronizing Word *is FPT when parameterized with $|I|$ and k; it can be solved in time $O^*(|I|^k)$.*

The above result is straight-forward, and one could hope for an improvement or a polynomial kernel for SW when parameterized with both k and $|I|$. Interestingly, no such improvement seems likely, as we show next.

Lemma 7. *Given a CNF formula ϕ with n variables and m clauses, a DFA $A = (S, I, \delta, s_0, F)$ can be constructed in $O(nm)$ time, such that $|S| = n+m+1$, $|I| = 2n$, and A has an n-synchronizing word if and only if ϕ is satisfiable.*

Proof. Let $V = \{x_1, \ldots, x_n\}$ be a set of variables in ϕ. Let $C = \{c_1, \ldots, c_m\}$ be the set of clauses in ϕ. We assume, w.l.o.g., that no variable occurs twice in any clause. The alphabet I contains $2n$ symbols x_i and $\overline{x_i}$ for $1 \leq i \leq n$, corresponding to the literals in the formula. We have the following $n + m + 1$ states:
Variable states q_i, $1 \leq i \leq n$; clause states c_j , $1 \leq j \leq m$; one sink state s.
The transitions are as follows:

1. $\delta(q_i, x_i) = \delta(q_i, \overline{x_i}) = q_{i+1}$, with $q_{n+1} = s$; $\delta(q_i, x_j) = \delta(q_i, \overline{x_j}) = q_i$ if $j \neq i$.
2. $\delta(c_j, l) = c_j$ for literal l if $l \notin c_j$. $\delta(c_j, l) = s$ for literal l if $l \in c_j$.
3. $\delta(s, l) = s$ for any literal l.

Notice that, as there are no transitions leading from the sink state to any other state, the state in which the synchronizing word (if it exists) must end is clear, it must be s. Any synchronizing word must be of length at least n as this is the length of the shortest path from q_1 to s. More precisely, any synchronizing word must be of the form described by the following regular expression:

$$(x_1 \cup \overline{x_1})^+ (x_2 \cup \overline{x_2})^+ \cdots (x_v \cup \overline{x_n})^+ (x_1 \cup \overline{x_1} \cup x_2 \cup \overline{x_2} \cup \cdots \cup x_n \cup \overline{x_n})^*.$$

This word should reflect the variable assignment. Namely, if there is a synchronizing word $\sigma = l_1 \cdots l_n$ of length n, then we can read off a variable assignment $\Phi : V \to \{0, 1\}$ as follows:

$$\Phi(x_i) = \begin{cases} 1, & \text{if } l_i = x_i \\ 0, & \text{if } l_i = \overline{x_i} \end{cases}$$

As σ leads into s in particular for each state c_j, this means that each clause c_j is satisfied by construction. The converse is similarly seen. □

Theorem 8. Synchronizing Word *cannot be solved in time $O^*(2^{o(t)})$ unless ETH fails.*

Proof. We start by using Lemma 7 to reduce a 3-SAT instance on n variables and m clauses to a SW instance $(A = (S, I, \delta, s_0, F), k)$ where $t = n + m + 1$, $|I| = 2n$, and $k = n$. If an algorithm existed that solved any SW instance in $O^*(2^{o(t)})$, then it would also solve 3-SAT in $O^*(2^{o(n+m)})$ time, contradicting ETH. □

Theorem 9. Synchronizing Word *does not have a polynomial kernel when parameterized with both k and $|I|$ unless* NP $\subseteq$ coNP/poly.

Proof. By Proposition 4 any CNF formula can be reduced to a SW instance $(A = (S, I, \delta, s_0, F), k)$ with $|I| = 2$ and $k = n$. If there existed a polynomial algorithm that produced an equivalent instance of size polynomial in $k, |I|$, this would mean that the number of states is reduced. As SW is NP-complete, there exists a polynomial time reduction back to a CNF formula with n' variables and m' clauses where $n' + m'$ is polynomial in $k, |I|$. This would imply that the number of clauses in this new CNF formula is bounded by a polynomial in n, and it is thus a polynomial kernel for SAT when parameterized by the number of variables n. By Proposition 1 this implies that $NP \subseteq coNP/poly$. □

Finally we turn our attention to parameter t. Again, there is a straight-forward FPT algorithm which is best possible.

Theorem 10 ([29]). Synchronizing Word *is FPT when parameterized with t; it can be solved in time $O^*(2^t)$.*

Theorem 11. Synchronizing Word *cannot be solved in time $O^*((|I| - \epsilon)^k)$ for any $\epsilon > 0$ unless SETH fails.*

Table 2. The table summarizes the results of this section. In addition we show that the parameter combination $(t, |I|)$ does not admit a polynomial kernel.

Parameter	Parameterized Complexity	Running time lower bound
t	NP-complete for $t = 2$	-
ℓ	NP-complete for $\ell = 2$	-
$\|I\|$	NP-complete for $\|I\| = 2$	-
c	Open	-
t, ℓ	NP-complete for $t\ell = 6$	-
$t, \|I\|$	FPT, running time $O^*(t^{t\|I\|})$	No $O^*(t^{o(t\|I\|)})$-time algorithm under ETH
t, c	Open	-
t, c, ℓ	FPT, running time $O^*(t^{c\ell})$	No $O^*(t^{o(c\ell)})$-time algorithm under ETH

4 DFA Consistency

In this section, we consider various parameterizations of the following problem:

Problem: DFA Consistency
Input: An alphabet I, two finite disjoint sets $X^+, X^- \subseteq I^*$, and an integer t
Question: Is there a DFA A with at most t states such that X^+ is accepted by A and X^- is rejected by A?

The natural parameters we work with here are the number of states t in the target DFA, the alphabet size $|I|$, the number of words $c = |X^+ \cup X^-|$, and the maximum length ℓ of any of the words in $X^+ \cup X^-$, i.e., $\max\{|\sigma| \mid \sigma \in X^+ \cup X^-\}$. The results of this section are summarized in Table 2. Notice that for the special case where $c = 2$ this problem is called the Separating Word Problem (for DFA), a recent overview can be found in [12].

Lemma 12. *Given a CNF formula φ with n variables and m clauses, an instance of the* DFA CONSISTENCY *problem can be constructed in time $O(nm)$, where $t = 2$, $|I| = 3n + 1$, $\ell = 2n$, and $c = 6n + m + 3$, such that there is a DFA on these parameters that distinguishes X^+ and X^- iff φ is satisfiable.*

A similar statement was shown by D. Angluin in 1989, but never published.

Lemma 13. *Let $G = (V, E)$ on n vertices and m edges be an instance of* 3-COLORING. *Then an instance of* DFA CONSISTENCY *can be constructed in time $O(n + m)$, where $t = 3$, $|I| = n + m$, $\ell = 2$, $c = 2m$, and such that there exists a DFA on these parameters that distinguishes X^+ and X^- iff G is 3-colorable.*

Proof. Let us first construct the DFA CONSISTENCY instance and then argue that it is a YES instance if and only if the 3-COLORING instance is a YES instance. We start by setting $t = 3$ and $I = V \cup E$, which leaves the definition of X^+ and X^-. Let $v_1, \ldots, v_n$ be an arbitrary numbering of the vertices in V. The sets of words X^+ and X^- are now constructed as follows:

- $X^+ = \{v_i e \mid e = v_i v_j \in E, i < j\}$;
- $X^- = \{v_j e \mid e = v_i v_j \in E, i < j\}$.

This completes the construction of the DFA CONSISTENCY instance.

Let us now argue for the equivalence of the two instances. For the first direction we assume that there exists a DFA $A = (S, I, \delta, s_0, F)$ on three states and alphabet $I = V \cup E$ that accept X^+ and rejects X^-. Let s_0, s_1, s_2 be the states of A and let v_i be contained in V_q for $0 \leq q \leq 2$ if $\delta(s_1, v_i) = s_q$. This gives us a partitioning V_0, V_1, V_2 of V. Our objective will now be to argue that V_q is an independent set in G for $0 \leq q \leq 2$. On the contrary, let $e = v_i v_j \in E$ where $i < j$ be an edge such that $v_i, v_j \in V_q$. From the construction of X^+ and X^- it is clear that set X^+ contains word $v_i e$ and set X^- contains $v_j e$. As the only difference between these two words is the first symbol and one word is accepted and the other one is rejected, it is clear that different states are reached by reading v_i and v_j from the start state s_0. Thus, either v_i or v_j is not contained in V_q and the contradiction is obtained.

For the second direction assume that there is a partitioning V_0, V_1, V_2 of V such that V_q is an independent set for $0 \leq q \leq 2$. Name the three states s_0, s_1, s_2 and let s_0 be the start state and s_1 the only accepting state. Function δ is now defined as follows:

1. $\delta(s_1, v_i) = s_q$, for $v_i \in V_q$ where $0 \leq q \leq 2$;
2. $\delta(s_q, v_i v_j) = s_1$ for $0 \leq q \leq 2$ and $i < j$;
3. $\delta(s_q, v_i v_j) = s_2$ for $0 \leq q \leq 2$ and $i > j$;

It is not hard to verify that all words in X^+ are accepted and all words in X^- are rejected. □

Lemma 14. *(also see [23]) Given a CNF formula φ with n variables and m clauses, an instance of the* DFA CONSISTENCY *problem can be constructed in* *time $O((n+m)^2)$, where $t = n+m+1$, $|I| = 2$, $\ell = m+n$, and $c = 5m+n+4$, such that there exists a DFA on these parameters that distinguishes X^+ and X^- iff φ is satisfiable.*

Theorem 15. DFA CONSISTENCY *cannot be solved in time $O^*(t^{o(t|I|)})$ unless ETH fails.*

Proof. Through the standard reduction from 3-SAT to 3-COLORING it follows that that 3-COLORING instance on n vertices and m edges can not be solved in time $O^*(2^{o(n+m)})$ unless ETH fails.

By the reduction of Lemma 13 we get an instance of DFA CONSISTENCY where $t = 3, |I| = n+m, \ell = 2$, and $c = 2m$. Any algorithm for DFA CONSISTENCY solving it in $O^*(t^{o(t|I|)})$ time will also solve 3-COLORING in $O^*(2^{o(n+m)})$ and ETH will fail. □

Theorem 16. DFA CONSISTENCY *does not have a polynomial kernel when parameterized with both t and $|I|$ unless* NP $\subseteq$ coNP/poly.

Proof. By Lemma 12 any CNF formula can be reduced to a a DFA CONSISTENCY instance where $t = 2$, $|I| = 3n+1$, $\ell = 2n$, and $c = 6n+m+3$ in polynomial time. If there existed a polynomial algorithm that produced an equivalent instance of size polynomial in $t, |I|$, this would mean that the number of words in $X^+ \cup X^-$ is reduced. As DFA CONSISTENCY is NP-complete, there exists a polynomial time reduction back to a SAT instance with n' variables and m' clauses where $n'+m'$ is polynomial in $t, |I|$. This would imply that the number of clauses in this CNF formula is bounded by a polynomial in n, and it is thus a polynomial kernel for SAT when parameterized by the number of variables. By Proposition 1 this implies that $NP \subseteq coNP/poly$. □

Next we turn to parameter combination (t, c, ℓ), which again gives a trivial FPT algorithm whose running time seems unlikely to be improvable.

Theorem 17. DFA CONSISTENCY *is FPT when parameterized with t, c, and ℓ; it can be solved in time $O^*(t^{c\ell})$.*

Theorem 18. DFA CONSISTENCY *cannot be solved in time $O^*(t^{o(c\ell)})$ unless ETH fails.*

Proof. Through the standard reduction from 3-SAT to 3-COLORING it follows that that 3-COLORING instance on n vertices and m edges can not be solved in time $O^*(2^{o(n+m)})$ unless ETH fails. By the reduction of Lemma 13 we get an instance of the DFA CONSISTENCY problem where $t = 3, |I| = n+m, \ell = 2$, and $c = 2m$. Any algorithm for the DFA CONSISTENCY problem solving the problem in $O^*(t^{o(c\ell)})$ time will also solve the 3-COLORING problem in $O^*(2^{o(m)})$ and ETH will fail. □

We end this section by turning our attention to parameter c. Could it be that DFA CONSISTENCY is NP-hard when $t = 2$ and c is bounded by a constant? We are able to answer this question partially with the below positive result.

Theorem 19. DFA CONSISTENCY *can be solved in polynomial time when $t = 2$ and $c = 2$.*

Table 3. A summary of the results on Q-SYNCHRONIZING WORD

Parameter	Parameterized Complexity
t	FPT with running time $O^*(2^t)$
$\|I\|$	PSPACE-complete for $\|I\| = 2$
k	W[2]-hard
k and $\|I\|$	FPT with running time $O^*(\|I\|^k)$
$\|Q\|$ and k	W[1]-hard
$\|Q\|$ and $\|I\|$	W[t]-hard for all t

5 Other Related Problems

The two core problems we investigated so far have quite a number of interesting variants for which several of our results carry over. We focus on SW-variants, often computationally harder than SW.

Based on the assumption that some partial information on the current state of a DFA might be known, formalized by a set of states Q, the Q-SYNCHRONIZING WORD (Q-SW) problem was introduced. In this problem we are only interested in finding a word x, $|x| \leq k$, that synchronizes all states from Q, i.e., $|\delta^*(Q, x)| = 1$. From [34] and the reduction from DFA INTERSECTION NONEMPTINESS given in [29] that shows PSPACE-hardness of this problem, we can immediately deduce the last two rows of Table 3. The only technical problem is that in the parameterized analogue DFA INTERSECTION, the length parameter m is an exact bound, while the length parameter k is an upper bound. However, by adding a sequence of m "new" states starting from some Q-state s_0, we can enforce the constructed DFA to have a word of length at least $m + 1$ as its shortest Q-synchronizing word. The reduction given in [29] will increase the word length by one.

Our parameterized complexity results for parameters t, $|I|$, and k transfer from SYNCHRONIZING WORD to this more general setting. Table 3 summarizes our results.

We also considered related problems on Mealy machines. For reasons of space, we only mention that finding short homing sequences leads to complexity results similar to synchronizing words, while finding short distinguishing sequences is more complex, simmilar to Q-synchronizing words.

6 Conclusion and Questions for Future Research

With this paper, we started some first steps in the multivariate analysis of several DFA (and Mealy machine) problems. Several questions emerge.

- Does SYNCHRONIZING WORD have a polynomial kernel with parameter t?
- Is DFA CONSISTENCY FPT when parameterized with c or with c and t?
- Does DFA CONSISTENCY have a polynomial kernel with parameter (t, c, ℓ)?

- There are other natural variants of DFA CONSISTENCY. Angluin showed [2] that REGULAR EXPRESSION CONSISTENCY is even hard for regular expressions of a very simple structure, without any nested Kleene stars, which sits very low in the famous star height hierarchy, see [11]. In view of the fact that for many applications, regular expressions are considered as important as DFAs, this could give an interesting line of research.
- What could be further natural parameters for problems on regular languages? Discovering these as possible sources of hardness could be a very fruitful line of research for both problem classes that we considered in this paper. Thoughts from the classical theory of Formal Languages could become very helpful, for instance, from Descriptional Complexity [24].

References

1. Alon, N., Moshkovitz, D., Safra, S.: Algorithmic construction of sets for k-restrictions. ACM Transactions on Algorithms 2(2), 153–177 (2006)
2. Angluin, D.: On the complexity of minimum inference of regular sets. Information and Control (now Information and Computation) 39, 337–350 (1978)
3. Arvind, V., Köbler, J., Lindner, W.: Parameterized learnability of juntas. Theoretical Computer Science 410(47-49), 4928–4936 (2009)
4. Berlinkov, M.V.: Approximating the Minimum Length of Synchronizing Words Is Hard. In: Ablayev, F., Mayr, E.W. (eds.) CSR 2010. LNCS, vol. 6072, pp. 37–47. Springer, Heidelberg (2010)
5. Björklund, H., Martens, W.: The tractability frontier for NFA minimization. Journal of Computer and System Sciences 78(1), 198–210 (2012)
6. Calabro, C., Impagliazzo, R., Paturi, R.: The Complexity of Satisfiability of Small Depth Circuits. In: Chen, J., Fomin, F.V. (eds.) IWPEC 2009. LNCS, vol. 5917, pp. 75–85. Springer, Heidelberg (2009)
7. Černý, J.: Poznámka k homogénnym experimentom s konečnými automatmi. Matematicko-fyzikálny Časopis 14(3), 208–216 (1964)
8. Chen, J., Zhang, F.: On product covering in 3-tier supply chain models: Natural complete problems for $W[3]$ and $W[4]$. Theoretical Computer Science 363(3), 278–288 (2006)
9. Costa Florêncio, C., Fernau, H.: On families of categorial grammars of bounded value, their learnability and related complexity questions. Theoretical Computer Science 452, 21–38 (2012)
10. Costa Florêncio, C., Verwer, S.: Regular Inference as Vertex Coloring. In: Bshouty, N.H., Stoltz, G., Vayatis, N., Zeugmann, T. (eds.) ALT 2012. LNCS (LNAI), vol. 7568, pp. 81–95. Springer, Heidelberg (2012)
11. Dejean, F., Schützenberger, M.P.: On a question of Eggan. Information and Control (now Information and Computation) 9(1), 23–25 (1966)
12. Demaine, E.D., Eisenstat, S., Shallit, J., Wilson, D.A.: Remarks on Separating Words. In: Holzer, M. (ed.) DCFS 2011. LNCS, vol. 6808, pp. 147–157. Springer, Heidelberg (2011)
13. Downey, R.G., Evans, P.A., Fellows, M.R.: Parameterized learning complexity. In: Proc. Sixth Annual ACM Conference on Computational Learning Theory, COLT, pp. 51–57. ACM Press (1993)
14. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer (1999)

15. Eppstein, D.: Reset sequences for monotonic automata. SIAM Journal on Computing 19(3), 500–510 (1990)
16. Fellows, M.: Towards Fully Multivariate Algorithmics: Some New Results and Directions in Parameter Ecology. In: Fiala, J., Kratochvíl, J., Miller, M. (eds.) IWOCA 2009. LNCS, vol. 5874, pp. 2–10. Springer, Heidelberg (2009)
17. Fellows, M.R., Fernau, H.: Facility location problems: A parameterized view. Discrete Applied Mathematics 159, 1118–1130 (2011)
18. Fellows, M.R., Gaspers, S., Rosamond, F.A.: Parameterizing by the number of numbers. Theory of Computing Systems 50(4), 675–693 (2012)
19. Fortnow, L., Santhanam, R.: Infeasibility of instance compression and succinct PCPs for NP. In: Dwork, C. (ed.) ACM Symposium on Theory of Computing, STOC, pp. 133–142. ACM (2008)
20. Garey, M.R., Johnson, D.S.: Computers and Intractability. Freeman, New York (1979)
21. Gold, E.M.: Language identification in the limit. Information and Control (now Information and Computation) 10, 447–474 (1967)
22. Gold, E.M.: Complexity of automaton identification from given data. Information and Control (now Information and Computation) 37, 302–320 (1978)
23. de la Higuera, C.: Grammatical inference. Learning automata and grammars. Cambridge University Press (2010)
24. Holzer, M., Kutrib, M.: Descriptional and computational complexity of finite automata - a survey. Information and Computation 209(3), 456–470 (2011)
25. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? Journal of Computer and System Sciences 63(4), 512–530 (2001)
26. Kohavi, Z.: Switching and Finite Automata Theory. McGraw-Hill (1970)
27. Lee, D., Yannakakis, M.: Testing finite state machines: State identification and verification. IEEE Transactions on Computers 43, 306–320 (1994)
28. Pitt, L., Warmuth, M.K.: The minimum consistent DFA problem cannot be approximated within any polynomial. Journal of the ACM 40, 95–142 (1993)
29. Sandberg, S.: Homing and Synchronizing Sequences. In: Broy, M., Jonsson, B., Katoen, J.-P., Leucker, M., Pretschner, A. (eds.) Model-Based Testing of Reactive Systems. LNCS, vol. 3472, pp. 5–33. Springer, Heidelberg (2005)
30. Stephan, F., Yoshinaka, R., Zeugmann, T.: On the parameterised complexity of learning patterns. In: Gelenbe, E., Lent, R., Sakellari, G. (eds.) Computer and Information Sciences II - 26th International Symposium on Computer and Information Sciences, ISCIS, pp. 277–281. Springer (2011)
31. Trahtman, A.N.: Modifying the Upper Bound on the Length of Minimal Synchronizing Word. In: Owe, O., Steffen, M., Telle, J.A. (eds.) FCT 2011. LNCS, vol. 6914, pp. 173–180. Springer, Heidelberg (2011)
32. Volkov, M.V.: Synchronizing Automata and the Černý Conjecture. In: Martín-Vide, C., Otto, F., Fernau, H. (eds.) LATA 2008. LNCS, vol. 5196, pp. 11–27. Springer, Heidelberg (2008)
33. Walker, P.J.: Synchronizing Automata and a Conjecture of Černý. Ph.D. thesis, University of Manchester, UK (2008)
34. Wareham, H.T.: The Parameterized Complexity of Intersection and Composition Operations on Sets of Finite-State Automata. In: Yu, S., Păun, A. (eds.) CIAA 2000. LNCS, vol. 2088, pp. 302–310. Springer, Heidelberg (2001)