More on Deterministic and Nondeterministic Finite Cover Automata* (Extended Abstract)

Hermann Gruber[1], Markus Holzer[2], and Sebastian Jakobi[2]

[1] knowledgepark AG, Leonrodstr. 68,
80636 München, Germany
hermann.gruber@knowledgepark-ag.de
[2] Institut für Informatik, Universität Giessen,
Arndtstr. 2, 35392 Giessen, Germany
{holzer,sebastian.jakobi}@informatik.uni-giessen.de

Abstract. Finite languages are an important sub-regular language family, which were intensively studied during the last two decades in particular from a descriptional complexity perspective. An important contribution to the theory of finite languages are the deterministic and the recently introduced nondeterministic finite *cover* automata (DFCAs and NFCAs, respectively) as an alternative representation of finite languages by ordinary finite automata. We compare these two types of cover automata from a descriptional complexity point of view, showing that these devices have a lot in common with ordinary finite automata. In particular, we study how to adapt lower bound techniques for nondeterministic finite automata to NFCAs such as, e.g., the biclique edge cover technique, solving an open problem from the literature. Moreover, the trade-off of conversions between DFCAs and NFCAs as well as between finite cover automata and ordinary finite automata are investigated. Finally, we present some results on the average size of finite cover automata.

1 Introduction

If one tries to describe formal objects such as, e.g., Boolean functions, graphs, trees, languages, as compact as possible we are faced with the question, which representation to use. This quest for compact representations of formal objects dates back to the early beginnings of theoretical computer science. For instance, one can prove by a *simple* counting argument that most Boolean functions have exponential circuit complexity [27]. For other representations of Boolean functions than circuits, such as formulas, ordered binary decision diagrams, etc. a similar result applies. This incompressibility is inherent in almost all possible representations of formal objects.

* Part of the work was done while the first author was at Institut für Informatik, Ludwig-Maximilians-Universität München, Oettingenstraße 67, 80538 München, Germany and the second author was at Institut für Informatik, Technische Universität München, Boltzmannstraße 3, 85748 Garching bei München, Germany.

When considering formal languages, automata are the preferred choice of representation. In particular, for regular languages and subfamilies one may use deterministic (DFAs) or nondeterministic finite automata (NFAs) or variants thereof to describe these languages. It is well known that these two formalisms are equivalent. The obvious way to obtain a DFA form a given NFA is by applying the *subset* or *power-set construction* [24]. This construction allows to show an upper bound of 2^n states in the DFA obtained from an n-state NFA, and this bound is known to be tight. For finite languages a slightly smaller bound on the determinization problem is given in [25]. Here the tight bound depends on the alphabet size k and reads as $\Theta(k^{\frac{n}{1+\log_2 k}})$. Thus, for a two-letter input alphabet $\Theta(2^{\frac{n}{2}})$ states are sufficient and necessary in the worst case for a DFA to accept a language specified by an n-state NFA. There are a lot of other results known for finite automata accepting finite languages such as, e.g., the maximal number of states of the minimal DFA accepting a subset of Σ^ℓ or $\Sigma^{\leq \ell}$ [5, 12], or the average case size of DFAs and NFAs w.r.t. the number of states and transitions accepting a subset of Σ^ℓ or $\Sigma^{\leq \ell}$ [17].

Since regular languages and finite automata are widely used in applications, and most of them use actually finite languages only, it is worth considering further representations for finite languages that may be more compact, but still bare nice handling in applications. Such a representation is based on finite automata and is known as finite cover automata. The idea is quite simple, namely a finite cover automaton A of a finite language $L \subseteq \Sigma^*$ is a finite automaton that accepts all words in L and possibly other words that are longer than any word in L. Formally, this reads as $L = L(A) \cap \Sigma^{\leq \ell}$, where ℓ is the length of the longest word(s) in L; then we say that A *covers* the finite language L. Originally deterministic finite cover automata (DFCAs) were introduced in [10], where an efficient minimization algorithm for these devices was given. Further results on important aspects of DFCAs can be found in, e.g., [8–11, 21]. Recently, DFCAs were generalized to nondeterministic finite cover automata (NFCAs) in [4] and it was shown that they can even give a more compact representation of finite languages than both NFAs and DFCAs. To our knowledge this was the first systematic study on this subject, although it has been suggested already earlier in a survey paper on cover automata [28].

We further develop the theory of finite cover automata in this paper. At first we introduce the necessary definitions in the next section. Then we briefly recall what is known on lower bound techniques for both types of finite cover automata. In particular, we first reconsider the fooling set techniques known for nondeterministic finite automata (NFAs) and secondly we show how to alter the biclique edge cover technique from [16] to make it applicable for NFCAs, too. This positively answers a question stated in [4], whether the biclique edge cover technique can be used at all to prove lower bounds for NFCAs. As a byproduct we develop a lower bound method for E-equivalent NFAs. This concept was recently introduced in [19]. Two languages are E-equivalent if their symmetric difference lies in the so called *error language* E. Thus, E-equivalence is a generalization of ordinary equivalence and also of cover-automata. In particular, setting $E = \Sigma^{>\ell}$, thus not taking care of words that are too long, we are back to covering languages and cover automata. Section 4 is devoted to conversions between finite automata and finite cover automata. First we provide a large family of languages where cover state complexity meets ordinary state complexity (up to one state for deterministic devices). Hence, for the conversions from finite automata to finite cover automata not much state savings are possible. For the opposite direction we show that an n-state finite cover automaton for a language of order ℓ can be converted to an equivalent finite automaton with about $n \cdot \ell$ states; the exact bounds are shown to be tight for all n and ℓ. In particular, this shows that roughly speaking the number of states of a finite cover automaton is *at least* an ℓ-th fraction of the state size of the equivalent finite automaton. Then we take a closer look on determinizing NFCAs by the well known powerset construction. We show that here the state blow-up heavily depends on the order ℓ of the finite language represented by the NFCA. When the order is large enough, we get a tight exponential blow-up of 2^n, just as in the case of ordinary finite automata. We give a range of conditions that imply sub-exponential, polynomial, and even linear determinization blow-ups. These results are presented in Section 5. In the penultimate section, we perform average case comparisons of the descriptional complexity of finite cover automata. For ordinary finite automata this was already done in, e.g., [17], where it was shown that almost all DFAs accepting finite languages of order ℓ over a binary input alphabet have state complexity $\Theta(2^\ell/\ell)$, while NFAs are shown to perform better, namely the nondeterministic state complexity is in $\Theta(\sqrt{2^\ell})$. Interestingly, in both cases the aforementioned bounds are asymptotically like in the worst case. For finite cover automata exactly the same picture as for ordinary finite automata emerges. Finally, we summarize our results in the conclusions section and state some open problems for future research. Due to space limitations all proofs are omitted.

2 Preliminaries

We recall some definitions on finite automata as contained in [18]. A *nondeterministic finite automaton* (NFA) is a quintuple $A = (Q, \Sigma, \delta, q_0, F)$, where Q is the finite set of *states*, Σ is the finite set of *input symbols*, $q_0 \in Q$ is the *initial state*, $F \subseteq Q$ is the set of *accepting states*, and $\delta\colon Q \times \Sigma \to 2^Q$ is the *transition function.* The *language accepted* by the NFA A is defined as

$$L(A) = \{\, w \in \Sigma^* \mid \delta(q_0, w) \cap F \neq \emptyset \,\},$$

where the transition function is recursively extended to $\delta\colon Q \times \Sigma^* \to 2^Q$. An NFA is *deterministic* (DFA), if and only if $|\delta(q, a)| = 1$, for every $q \in Q$ and $a \in \Sigma$. In this case we simply write $\delta(q, a) = p$ instead of $\delta(q, a) = \{p\}$, assuming that the transition function $\delta\colon Q \times \Sigma \to Q$ is a *total* mapping. Two automata A and B are *equivalent* if they accept the same language, that is, $L(A) = L(B)$. An NFA (DFA, respectively) A is *minimal* if any equivalent NFA (DFA, respectively) needs at least as many states as A. It is a well known fact that minimal DFAs are unique up to isomorphism, while minimal NFAs are *not* necessarily unique in general. Let $\mathsf{nsc}(L)$ ($\mathsf{sc}(L)$, respectively) refer to the number of states a minimal NFA (DFA, respectively) needs to accept the language L. By definition and the seminal result in [24] we have $\mathsf{nsc}(L) \leq \mathsf{sc}(L) \leq 2^{\mathsf{nsc}(L)}$, if L is a language accepted by a finite automaton. Proving lower bounds for $\mathsf{nsc}(L)$ can be done by applying, e.g., the extended fooling set technique, which reads as follows [1]:

Theorem 1. *Let $L \subseteq \Sigma^*$ be a regular language and suppose there exists a set of pairs $S = \{ (x_i, y_i) \mid 1 \leq i \leq n \}$ such that (i) $x_i y_i \in L$, for $1 \leq i \leq n$ and (ii) $i \neq j$ implies $x_i y_j \notin L$ or $x_j y_i \notin L$, for $1 \leq i, j \leq n$. Then any nondeterministic finite automaton for L has at least n states, i.e., $n \leq \mathsf{nsc}(L)$. Here S is called an* extended fooling set *for L.*

A non-empty finite language $L \subseteq \Sigma^*$ is said to be of *order ℓ*, if ℓ is the length of the longest word(s) in the set L, i.e., $L \subseteq \Sigma^{\leq \ell}$, where $\Sigma^{\leq \ell}$ refers to the set $\{ w \in \Sigma^* \mid |w| \leq \ell \}$, where $|w|$ denotes the length of the word w. In particular, the length of the empty word λ is zero. A *deterministic finite cover automaton* (DFCA) for a language $L \subseteq \Sigma^*$ of order ℓ is a DFA A such that $L(A) \cap \Sigma^{\leq \ell} = L$; these devices were introduced in [10]. This definition naturally carries over to NFAs, hence leading to *nondeterministic finite cover automata* (NFCA), which were recently introduced in [4]. Two cover automata A and B are *equivalent* if they cover the same finite language $L \subseteq \Sigma^*$, that is, $L(A) \cap \Sigma^{\leq \ell} = L(B) \cap \Sigma^{\leq \ell}$, where ℓ is the order of L. A DFCA (NFCA, respectively) A for a finite language L is *minimal* if any equivalent automaton of same type needs at least as many states as A. Let $\mathsf{ncsc}(L)$ ($\mathsf{csc}(L)$, respectively) refer to the number of states a minimal NFCA (DFCA, respectively) needs to accept the finite language L. By definition we have $\mathsf{ncsc}(L) \leq \mathsf{csc}(L)$, if L is a finite language. Moreover, since any cover automaton can be at most as large as an ordinary finite automaton of the same type for a finite language L, we have $\mathsf{csc}(L) \leq \mathsf{sc}(L)$ as well as $\mathsf{ncsc}(L) \leq \mathsf{nsc}(L)$. A useful tool for the study of minimal DFCAs is the notion the similarity relation, which plays a similar role as the Myhill-Nerode relation[3] in case of DFAs. For a finite language $L \subseteq \Sigma^*$ of order ℓ the similarity relation $\approx_L$ on words is defined as follows: for $u, v \in \Sigma^*$ let $u \approx_L v$ if and only if we have $uw \in L \iff vw \in L$, for all $w \in \Sigma^*$, whenever $|uw| \leq \ell$ and $|vw| \leq \ell$. Observe, that $\approx_L$ is not a equivalence relation in general. The relation $\approx_L$ can also be defined for states of a DFCA $A = (Q, \Sigma, \delta, q_0, F)$. Two states p and q are *similar*, denoted by $p \approx_L q$, if $\delta(p, w) \in F \iff \delta(q, w) \in F$ holds for all $w \in \Sigma^{\leq \ell - m}$, with $m = \max(lev_A(p), lev_A(q))$—here $lev_A(p) = \min\{ |u| \mid \delta(q_0, u) = p \}$. If $p \not\approx_L q$ then p and q are *dissimilar.* It is known [10] that a DFCA is minimal if all its states are pairwise dissimilar.

3 Lower Bound Techniques For Cover Automata

The problem to estimate the necessary number of states of a minimal NFA accepting a given regular language is complicated. Several authors have introduced

[3] For a language $L \subseteq \Sigma^*$ define the Myhill-Nerode relation $\equiv_L$ on words as follows: for $u, v \in \Sigma^*$ let $u \equiv_L v$ if and only if $uw \in L \iff vw \in L$, for all $w \in \Sigma^*$.

methods for proving lower bounds. The most widely used lower bound techniques for NFAs are the so-called *fooling set* techniques—the fooling set technique [14] and the extended fooling set method [1]. Recently, in [4] both fooling set methods were adapted to work for NFCAs as well. Here we first reconsider the fooling set techniques and then show how to modify yet another lower bound method, the biclique edge cover technique of [16], to work with NFCAs. Whether this latter technique can be generalized to NFCAs was stated as an open problem in [4].

In [4] it was argued that there is no doubt that any fooling set type technique used to prove a lower bound for NFCAs must explicitly consider the order of the language under consideration. In this vein, both fooling set techniques were adapted. In fact, we show that the original fooling set technique of [14] (not the extended version of [1]) already gives a lower bound for NFCAs without modifying the technique to explicitly deal with the order of the language under consideration.

Theorem 2. *Let $L \subseteq \Sigma^*$ be a finite language and suppose there exists a set of pairs $S = \{ (x_i, y_i) \mid 1 \leq i \leq n \}$ such that (i) $x_i y_i \in L$, for $1 \leq i \leq n$, and (ii) $x_i y_j \notin L$, for $1 \leq i, j \leq n$, and $i \neq j$. Then any nondeterministic finite* cover *automaton for L has at least n states, i.e., $n \leq \mathsf{ncsc}(L)$. Here S is called a fooling set for L.* □

In contrast the more powerful extended fooling set technique presented in [1] does not work as a lower bound technique for NFCAs as the following example illustrates, and therefore the modification of this technique presented in [4] is the right generalization.

Example 3. Consider the unary finite language $L = \{a\}^{\leq \ell}$, for $\ell \geq 1$. Clearly, this language can be covered by an NFCA with a single state. However, the set $S = \{ (a^i, a^{\ell - i}) \mid 0 \leq i \leq \ell \}$ is an extended fooling set for L, proving a lower bound of $\ell + 1$ on the nondeterministic state complexity of L. □

In the remainder of this subsection we turn our attention to the biclique edge cover technique from [16]. A central role in this technique plays the notion of the *bipartite dimension* $\dim(G)$ of a bipartite graph G, which is the minimum number of bicliques in G needed to cover all edges of G. The following example shows that this technique cannot be applied to NFCAs without any modification.

Example 4. Let $\ell \geq 1$ and consider the finite language $L = \{a\}^{\leq \ell}$. Clearly the single-state DFA accepting for the language $\{a\}^*$ is a cover automaton for L, hence we have $\mathsf{ncsc}(L) = 1$. However, the bipartite dimension of the graph $G = (X, Y, E)$, with $X = Y = L$ and $E = \{ (x, y) \in X \times Y \mid xy \in L \}$, is $\ell + 1 > 1$. This can be seen as follows. Notice that $(a^i, a^j) \in E$ if and only if $i + j \leq \ell$. In particular, for $0 \leq i \leq \ell$, the edge $e_i = (a^i, a^{\ell - i})$ belongs to E. Therefore, every such e_i has to be covered by some biclique $H_i = (X_i, Y_i, E_i)$ with $a^i \in X_i$, $a^{\ell - i} \in Y_i$, and $E_i = X_i \times Y_i$. Now we see that distinct edges e_i and e_j must be covered by distinct bicliques, that is, $H_i \neq H_j$, for $1 \leq i, j \leq \ell$, with $i \neq j$: if $H_i = H_j$ then we have $a^i, a^j \in X_i$ and $a^{\ell - i}, a^{\ell - j} \in Y_i$, and since H_i is a biclique, its set of edges E_i contains both $(a^i, a^{\ell-j})$ and $(a^j, a^{\ell-i})$. But since $i \neq j$, either $i + \ell - j > \ell$ or $j + \ell - i > \ell$, which means that one of the two edges does not belong to E—a contradiction to $H_0, H_1, \ldots H_\ell$ being a biclique edge cover. This shows that the bipartite dimension of G is at least $\ell + 1$. Equality is witnessed by the bicliques $H_i = (X_i, Y_i, E_i)$ with $X_i = \{a^i\}$, $Y_i = \{a\}^{\leq \ell - i}$, and $E_i = X_i \times Y_i$, for $0 \leq i \leq \ell$. □

In the following we want to generalize the biclique edge cover technique so that it can also be used to prove lower bounds for the size of NFCAs. In fact, we present a generalization that can be used even for the more general notion of E-equivalent automata, which was recently introduced in [19]. In order to avoid confusion with the set of edges of a graph, we use here the term D-equivalence instead of E-equivalence. Let $D \subseteq \Sigma^*$ be some language, the so called *error language.* Two languages L and L' over the alphabet Σ are called *D-equivalent* if they differ only on elements from the error language D, that is, if

$$(L \setminus L') \cup (L' \setminus L) \subseteq D.$$

In this case we write $L \sim_D L'$. Similarly, two automata A and B are D-equivalent, if $L(A) \sim_D L(B)$. The connection between D-equivalence and cover automata is as follows. Assume $L \subseteq \Sigma^{\leq \ell}$ is some finite language of order ℓ. Then a language $L' \subseteq \Sigma^*$ is a cover language for L if and only if $L \sim_D L'$, for the error language $D = \Sigma^{>\ell}$. In other words, any two cover languages L' and L'' for a finite language of order ℓ are D-equivalent, for $D = \Sigma^{>\ell}$.

We now come to our generalization of the biclique edge cover technique. In the original technique we have to find bicliques $H_i = (X_i, Y_i, E_i)$ with $1 \leq i \leq k$, for some k, of a bipartite graph $G = (X, Y, E)$, such that $E = \bigcup_{i=1}^{k} E_i$. In our generalization, we use two sets of edges in the bipartite graph G, namely a set $\underline{E}$ of edges that *must* be covered, and a set $\overline{E}$, with $\underline{E} \subseteq \overline{E}$, of edges that *may* be covered by bicliques. We use the notation $G = (X, Y, \underline{E}, \overline{E})$ to denote such a bipartite graph. Now an *$(\underline{E}, \overline{E})$-approximation* of G is a collection of bicliques $H_i = (X_i, Y_i, E_i)$ of G, with $1 \leq i \leq k$ for some k, such that

$$\underline{E} \subseteq \bigcup_{i=1}^{k} E_i \subseteq \overline{E}.$$

The *$(\underline{E}, \overline{E})$-dimension* of G, denoted by $\dim^*(G)$, is defined as the minimal number of bicliques that constitute an *$(\underline{E}, \overline{E})$-approximation* of G.

Now we are ready to present our lower bound technique for D-equivalent automata. Notice that the sets $\underline{E}$ and $\overline{E}$ of edges of graph G in the following theorem depend on the given language L and error set D by definition.

Theorem 5. *Let L and D be languages over some alphabet Σ. Moreover, let $X, Y \subseteq \Sigma^*$ and $G = (X, Y, \underline{E}, \overline{E})$, with $\underline{E} = \{\, (x, y) \in X \times Y \mid xy \in L \setminus D \,\}$ and $\overline{E} = \{\, (x, y) \in X \times Y \mid xy \in L \cup D \,\}$. Then the number of states of any nondeterministic finite automaton A, with $L(A) \sim_D L$, is at least* $\dim^*(G)$. □

Notice that Theorem 5 yields the original biclique edge cover technique when choosing the error language $D = \emptyset$, that is, when considering the special case of classical language equivalence. Moreover, with the error language $D = \Sigma^{>\ell}$ we obtain the following technique for proving lower bounds on the state complexity of nondeterministic cover automata for finite languages of order ℓ.

Corollary 6. *Let $L \subseteq \Sigma^*$ be some finite language of order ℓ. Moreover, let $X, Y \subseteq \Sigma^*$ and $G = (X, Y, \underline{E}, \overline{E})$, with $\underline{E} = \{\, (x, y) \in X \times Y \mid xy \in L \,\}$ and $\overline{E} = \{\, (x, y) \in X \times Y \mid xy \in L \cup \Sigma^{>\ell}, \}$. Then the number of states of any nondeterministic finite cover automaton for L is at least* $\dim^*(G)$*, that is,* $\dim^*(G) \leq \mathsf{ncsc}(L)$. □

4 Conversions Between Finite Automata and Cover Automata

In this section we compare the descriptional complexity of finite automata and cover automata, by studying the cost of conversions between these models. We consider nondeterministic as well as deterministic automata.

4.1 From Finite Automata to Cover Automata

Clearly, a finite automaton for a finite language L is also a cover automaton for that language. So the bounds $\mathsf{ncsc}(L) \leq \mathsf{nsc}(L)$ and $\mathrm{csc}(L) \leq \mathsf{sc}(L)$ are obvious. However, the question is whether these bounds are tight in the following sense: does there exist, for every integer $n \geq 1$, a regular language L_n that is accepted by a DFA (NFA, respectively) with n states such that the minimal DFCA (NFCA, respectively) needs n states, too? The next result answers this question in the affirmative for nondeterministic automata, while for deterministic devices the bound is off by one.

Theorem 7. *If L is a finite language with all words having the same length ℓ, then $\mathsf{ncsc}(L) = \mathsf{nsc}(L)$ and $\mathsf{csc}(L) = \mathsf{sc}(L) - 1$.* □

From Theorem 7 and the obvious upper bound $\mathsf{ncsc}(L) \leq \mathsf{nsc}(L)$ we obtain the following result. In fact, Theorem 7 provides the lower bound already by *unary* witness languages.

Corollary 8. *Let $n \geq 1$ and L be a finite language accepted by a nondeterministic finite automaton with n states. Then n states are sufficient and necessary in the worst case for a nondeterministic finite cover automaton to accept L. This bound is tight already for a unary alphabet.* □

Next we want to close the gap between the lower and upper bound for the conversion from DFAs to DFCAs.

Theorem 9. *Let L be a finite language accepted by a deterministic finite automaton with n states. If $n = 1$ or $n \geq 4$ then n states are sufficient and necessary in the worst case for a deterministic finite cover automaton to accept L. These bounds are tight already for binary alphabets. If $n \in \{2, 3\}$, or if $n \geq 2$ and L is a unary language, then $n - 1$ states are sufficient and necessary in the worst case.* □

We also note that the conversion from NFAs to DFCAs was investigated already in [6]. They present binary languages L_n that can be accepted by an n-state NFA, while $2^{n-t} - 2^{t-2} + 2^t - 1$ states are necessary, with $t = \lfloor \frac{n}{2} \rfloor$, for a deterministic finite cover automaton to accept L_n. Then they generalize their examples to larger alphabets. The lower bound is known to be tight if n is even, but the tight bound for odd n remains to be determined.

4.2 From Cover Automata to Finite Automata

In the previous subsection we have seen that there are finite languages where the description size cannot be reduced when changing the descriptional model from finite automata to cover automata. In this section we now consider the inverse conversion: given a cover automaton for a finite language, how large can a minimal finite automaton for that language become? In this setting we will see that the number of states of a cover automaton alone is not a fair size measure. In fact, we propose that a reasonable size measure for cover automata must also take the cover length into account: for every integer $\ell \geq 0$ the finite language $\{a\}^{\leq \ell}$ can be covered by a single-state cover automaton, but a NFA for this language has at least $\ell + 1$ states. Therefore, if we start with a cover automaton with n states that describes a finite language of order ℓ, then the number of states of an equivalent finite automaton should be a function in n *and* ℓ.

Since the language L described by a cover automaton A with cover length ℓ satisfies $L = L(A) \cap \Sigma^{\leq \ell}$, a finite automaton for L can be obtained by applying a cross product construction on A and an automaton for $\Sigma^{\leq \ell}$. The states of the constructed automaton are pairs (q, i), where q is a state of A, and i is a counter for the word length. This yields upper the upper bounds $\mathsf{nsc}(L) \leq \mathsf{ncsc}(L) \cdot (\ell + 1)$ and $\mathsf{sc}(L) \leq \mathsf{csc}(L) \cdot (\ell + 2)$ for finite languages L of order ℓ. In the upcoming lemma we show that these bounds can be slightly reduced. In the following we do not consider languages of order $\ell = 0$, because the only such language is $\{\lambda\}$, which is accepted by a single-state NFA and a two-state DFA. Moreover, the case where $\mathsf{ncsc}(L) = 1$ is also omitted—here it is easy to see that the upper bounds $\mathsf{nsc}(L) \leq \ell + 1$ and $\mathsf{sc}(L) \leq \ell + 2$ apply, and optimality is witnessed by the language $L = \Sigma^{\leq \ell}$.

Lemma 10. *Let $n \geq 2$ and A be an n-state nondeterministic cover automaton for a finite language L of order $\ell \geq 1$. Then one can construct a nondeterministic finite automaton for L that has at most $n \cdot (\ell - 1) + 2$ states. If A is deterministic, then one can construct a deterministic finite automaton for L with $n \cdot (\ell - 1) + 3$ states.* □

Next we show that the constructions from Lemma 10 cannot be improved in general by providing a matching lower bounds. Observe that the following lemma even provides a lower bound for the conversion from *deterministic* cover automata to *nondeterministic* finite automata.

Lemma 11. *For every integers $n \geq 2$ and $\ell \geq 1$ there exists a finite language L of order ℓ that is described by a deterministic n-state cover automaton, such that any nondeterministic finite automaton for L needs $n \cdot (\ell - 1) + 2$ states, and any deterministic finite automaton for L needs $n \cdot (\ell - 1) + 3$ states.* □

From Lemmata 10 and 11 we obtain the following result.

Theorem 12. *Let L be a finite language of order $\ell \geq 1$ that is described by a nondeterministic cover automaton A with $n \geq 2$ states. Then $n \cdot (\ell - 1) + 2$ states are sufficient and necessary in the worst case for a nondeterministic finite automaton to accept L. Moreover, if A is a deterministic cover automaton for L, then $n \cdot (\ell - 1) + 3$ states are sufficient and necessary in the worst case for a deterministic finite automaton to accept L.* □

The proof for the lower bound from Lemma 11 uses $2n - 2$ alphabet symbols. In fact, one can also show that the bounds $\mathsf{nsc}(L) \leq \mathsf{ncsc}(L) \cdot (\ell - 1) + 2$ and $\mathsf{sc}(L) \leq \mathsf{csc}(L) \cdot (\ell - 1) + 3$ for the conversions from cover automata to finite automata are *not* tight for languages over an alphabet of constant size. For the deterministic case, this is easy to see: assuming a k-letter alphabet Σ, at most k different states of the form $(q, 1)$ are reachable from the initial state $(q_0, 0)$ in the DFA constructed from a DFCA as shown in the proof of Lemma 10.

Although this argumentation does not hold for nondeterministic automata, where every state of the given NFCA could be reachable in one step from the initial state, the number of states of an equivalent minimal NFA still depends on the number of alphabet symbols: when using the construction from Lemma 10 to obtain an NFA A' for the language $L \subseteq \Sigma^{\leq \ell}$, the automaton A' has a distinguished "last" accepting state $(\bullet, \ell)$, which has no outgoing transitions. This state is only reachable from states of the form $(q, \ell - 1)$, and from such states no other state is reachable. Assume that two such states $(p, \ell - 1)$ and $(q, \ell - 1)$ go to state $(\bullet, \ell)$ on the same set of input letters. If additionally p and q are of same acceptance value, then clearly they can be merged into a single state. Since a k-letter alphabet Σ has $2^k - 1$ non-empty subsets, the number of accepting states of the form $(q, \ell - 1)$ can always be reduced to $2^k - 1$, and similarly for the non-accepting states. So in total there are at most $2 \cdot (2^k - 1)$ states of the form $(q, \ell - 1)$, which may be large compared to k, but it is still a constant.

5 Determinization of Finite Cover Automata

In this section we continue our descriptional complexity studies of cover automata: we investigate the cost of determinization, that is, the conversion from a nondeterministic to a deterministic cover automaton. A classical result in the theory of finite automata is that every n-state NFA can be converted by the so-called power-set construction to an equivalent DFA with at most 2^n states [24]. Moreover, it is known that this bound is tight in the sense that for every $n \geq 1$ there exists a language accepted by a minimal n-state NFA, and for which the minimal DFA needs exactly 2^n states [23]. Now the question is to which extent these results carry over to cover automata. Clearly, since the power-set construction for finite automata preserves the accepted language, it can be used to convert an NFCA into an equivalent DFCA. Thus, the following is immediate.

Lemma 13. *Let L be a finite language described by a nondeterministic cover automaton with $n \geq 1$ states. Then one can construct a deterministic cover automaton for L that has at most 2^n states.* □

Our next goal is to prove a matching lower bound of 2^n states for the determinization of n-state NFCAs. The next fact we present is useful to show that a number of worst case results known for the state complexity of deterministic finite automata carry over to the setting of cover automata.

Theorem 14. *Assume L is a regular language over Σ with $\mathsf{sc}(L) = n$, and let $L' = L \cap \Sigma^{\leq n+2^n}$. Then $\mathrm{csc}(L') = n$.* □

Theorem 14 implies that if the order of the language is large compared to the size of the NFA, then determinization of cover automata is as expensive as for usual finite automata. In particular, classical examples for finite automata [23] show that the full blow-up from n states to 2^n states may be necessary for converting an NFCA into an equivalent DFCA. Together with Lemma 13 we obtain the following result.

Corollary 15. *Let L be a finite language that is described by a nondeterministic cover automaton with $n \geq 1$ states. Then 2^n states are sufficient and necessary in the worst case for a deterministic cover automaton to accept L.* □

A natural question is now whether the full blow-up can be reached if the order of the described language is small compared to the number of states in the given NFCA. First, recall that every finite language L of order ℓ over a k-letter alphabet satisfies $\mathsf{sc}(L) \leq (1 + \mathrm{o}(1))\frac{k^{\ell+2}}{d_k \ell}$ with $d_k = (k-1)^2 \log k$; see [5]. This shows that the full blow-up cannot be reached if ℓ is too small compared to n. From that result and the fact that $\mathsf{csc}(L) \leq \mathsf{sc}(L)$, the following bounds for the size of a deterministic cover automaton can be derived. In fact, since the proof of the next result only uses the above bound on $\mathsf{sc}(L)$, the statements also hold for the determinization of finite automata.

Theorem 16. *Let L be a finite language of order ℓ over a k-letter alphabet Σ and assume L is described by a nondeterministic finite cover automaton with n states.*

1. *If $(\ell + 2) \cdot \log k - \log \ell + 1 < n$, then $\mathsf{csc}(L) < 2^n$, for large enough n.*
2. *if $\ell \in \mathrm{o}(n)$, then $\mathsf{csc}(L) \in 2^{\mathrm{o}(n)}$,*

3. *if $\ell \in \mathrm{O}(\log n)$, then $\mathsf{csc}(L) \in n^{\mathrm{O}(1)}$,*
4. *if $(\ell + 2) \cdot \log k - \log \ell + 1 < \log n$, then $\mathsf{csc}(L) < n$, for large enough n.* □

The fourth statement in the above theorem is of particular practical relevance: in this case, the given n-state NFCA is not minimal, and determinization followed by minimization yields a smaller cover automaton. In contrast to languages of order less than n, where the blow-up of 2^n states cannot be achieved, there are quite natural examples reaching the full blow-up already for order linear in the number of states of the NFCA. The example used in the following proof is essentially due to [22, Lemma 2]:

Theorem 17. *Let $L_n = \big(a + (a \cdot b^*)^{n-1} \cdot a\big)^* \cap \Sigma^{\leq 5n-2}$. Then L_k can be covered by an n-state nondeterministic cover automaton, but the smallest deterministic cover automaton for L_n has at least 2^n states.* □

6 Average Size Comparisons of Finite Cover Automata

This section is devoted to the average case state complexity of DFCAs and NFCAs, when choosing a finite language of a certain "size" ℓ uniformly at random from all finite languages of that particular size. Here size means that all words of the language are either of the same length ℓ, or of length at most ℓ. This model was used in [17] to compare the number of states or transitions of ordinary finite automata on average. There it is shown that almost all DFAs accepting finite languages over a binary input alphabet have state complexity $\Theta(2^\ell/\ell)$, while NFAs are shown to perform better, namely the nondeterministic state complexity is in $\Theta(\sqrt{2^\ell})$. Interestingly, in both cases the aforementioned bounds are asymptotically like in the worst case. As we will see, a similar situation emerges for finite cover automata as well. The first theorem gives us the expected number of states a DFCA has on average, if we assume that all finite languages from $\mathfrak{P}(\Sigma^{\leq \ell})$, that is, the power-set of $\Sigma^{\leq \ell}$, are equiprobable.

Theorem 18. *Let Σ be an alphabet of size k and $c_k = (k-1)\log k$. Then $\mathbb{E}[\mathrm{csc}(L)] \geq (1 - \mathrm{o}(1)) \frac{k^\ell}{c_k \ell}$, if L is a language drawn uniformly at random from the power-set of $\Sigma^{\leq \ell}$.* □

Regarding an upper bound, it is known from [5] that $\mathsf{sc}(L) \leq (1 + \mathrm{o}(1)) \frac{k^{\ell+2}}{d_k \ell}$, as ℓ tends to infinity, with $d_k = (k-1)^2 \log_2 k$, for languages $L \subseteq \Sigma^{\leq \ell}$ and alphabet size k. This generalized a previous result of [12]. Recall that the size of a minimum DFA for a finite language is an upper bound for the size of a minimum DFCA; and the state complexity in the worst case is of course an upper bound for the average state complexity. So the above average case result is tight up to a factor of at most $(1 + \mathrm{o}(1)) \frac{k^2}{(k-1)}$. Next we turn our attention to the average state complexity of NFCAs.

Theorem 19. *Let Σ be an alphabet of size k. Then for large enough ℓ we have $\mathbb{E}[\mathsf{ncsc}(L)] > k^{\frac{\ell}{2}-1}$, if L is a language drawn uniformly at random from the power-set of $\Sigma^{\leq \ell}$.* □

A worst case upper bound for the nondeterministic state complexity of subsets of $\Sigma^{\leq \ell}$ is given in [17] for binary alphabets. Generalizing this result to cover automata and larger alphabets, the bound reads as follows:

Theorem 20. *Let Σ be an alphabet of size k. Then $\mathsf{ncsc}(L) \leq \mathsf{nsc}(L) < \frac{3}{k-1}\sqrt{k^{\ell}}$, if L is any subset of $\Sigma^{\leq \ell}$, i.e., $L \subseteq \Sigma^{\leq \ell}$.* □

7 Conclusions

We completed the picture of lower bound techniques for nondeterministic finite cover automata, and solved the problems left open in [4]. Then we determined the precise best-case and worst-case bounds for conversions between DFCAs and DFAs, as well as between NFCAs and NFAs. In [6], almost tight bounds for the conversion between NFAs and DFCAs were given. Determining the precise bound in this case remains an open problem.

When the length ℓ of the longest word is much smaller than the number n of states in a minimal cover automaton, then the succinctness gain offered by finite cover automata over finite automata is very modest, even in the best case. We note that this is the case in the area of natural language processing: in [20] they construct a minimum 29317-state DFA accepting 81142 English words. Of course, almost all common English words have $\ell < 20$. Similarly, in [26] they construct an NFA accepting roughly 230000 Greek words.

Finally, a recent experimental study [7] showed that for binary finite languages, the expected reduction in the number of states provided by DFCAs is negligible. Our analysis of the average case provides a theoretical underpinning of their observations. One may study further random models of finite languages, e.g., a Bernoulli-type model [17], and one based on the sum of word lengths [2].

References

1. Birget, J.C.: Intersection and union of regular languages and state complexity. Inform. Process. Lett. 43, 185–190 (1992)
2. Bassino, F., Giambruno, L., Nicaud, C.: The average state complexity of rational operations on finite languages. Internat. J. Found. Comput. Sci. 21(4): 495–516 (2010)
3. Brzozowski, J.A.: Canonical regular expressions and minimal state graphs for definite events. In: Mathematical Theory of Automata, MRI Symposia Series, vol. 12, pp. 529–561. Polytechnic Press, NY (1962)
4. Câmpeanu, C.: Non-deterministic finite cover automata. Scientific Annals of Computer Science 25(1), 3–28 (2015)
5. Câmpeanu, C., Ho, W.H.: The maximum state complexity for finite languages. J. Autom., Lang. Comb. 9(2–3), 189–202 (2004)
6. Câmpeanu, C., Kari, L., Păun, A.: Results on Transforming NFA into DFCA. Fundam. Inform. 64(1–4), 53–63 (2005)
7. Câmpeanu, C., Moreira, N., Reis, R.: Expected compression ratio for DFCA: experimental average case analysis. Universidade do Porto (2011), Technical Report DCC-2011-07

8. Câmpeanu, C., Păun, A., Smith, J.R.: Tight bounds for the state complexity of deterministic cover automata. In: Leung, H., Pighizzini, G. (eds.) Proceedings of the 8th Workshop on Descriptional Complexity of Formal Systems. pp. 223–231. Las Cruces, New Mexico, USA (2006), Computer Science Technical Report NMSU-CS-2006-001
9. Câmpeanu, C., Păun, A., Yu, S.: An efficient algorithm for constructing minimal cover automata for finite languages. Internat. J. Found. Comput. Sci. 13(1), 83–97 (2002)
10. Câmpeanu, C., Sântean, N., Yu, S.: Minimal cover-automata for finite languages. Theoret. Comput. Sci. 267(1–2), 3–16 (2001)
11. Champarnaud, J.M., Guingne, F., Hansel, G.: Similarity relations and cover automata. RAIRO–Informatique théorique et Applications / Theoretical Informatics and Applications 39(1), 115–123 (January–March 2005)
12. Champarnaud, J.M., Pin, J.E.: A maxmin problem on finite automata. Discrete Appl. Math. 23, 91–96 (1989)
13. Domaratzki, M., Kisman, D., Shallit, J.: On the number of distinct languages accepted by finite automata with n states. J. Autom., Lang. Comb. 7(4), 469–486 (2002)
14. Glaister, I., Shallit, J.: A lower bound technique for the size of nondeterministic finite automata. Inform. Process. Lett. 59, 75–77 (1996)
15. Gramlich, G., Schnitger, G.: Minimizing nfa's and regular expressions. J. Comput. System Sci. 73(6), 908–923 (2007)
16. Gruber, H., Holzer, M.: Finding lower bounds for nondeterministic state complexity is hard (extended abstract). In: Ibarra, O.H., Dang, Z. (eds.) Proceedings of the 10th International Conference on Developments in Language Theory. pp. 363–374. No. 4036 in LNCS, Springer, Santa Barbara, California, USA (2006)
17. Gruber, H., Holzer, M.: Results on the average state and transition complexity of finite automata accepting finite languages. Theoret. Comput. Sci. 387(2), 155–166 (2007)
18. Harrison, M.A.: Introduction to Formal Language Theory. Addison-Wesley (1978)
19. Holzer, M., Jakobi, S.: From equivalence to almost-equivalence, and beyond: Minimizing automata with errors. Internat. J. Found. Comput. Sci. 24(7), 1083–1134 (2013)
20. Lucchesi, C.L., Kowaltowski, T.: Applications of Finite Automata Representing Large Vocabularies. Softw., Pract. Exper. 23(1), 15–30 (1993)
21. Körner, H.: A time and space efficient algorithm for minimizing cover automata for finite languages. Internat. J. Found. Comput. Sci. 14(6), 1071–1086 (2003)
22. Leung, H.: Separating exponentially ambiguous finite automata from polynomially ambiguous finite automata. SIAM J. Comput. 27(4), 1073–1082 (1998)
23. Lupanov, O.B.: Über den Vergleich zweier Typen endlicher Quellen. Probleme der Kybernetik 6, 328–335 (1966)
24. Rabin, M.O., Scott, D.: Finite automata and their decision problems. IBM J. Res. Dev. 3, 114–125 (1959)
25. Salomaa, K., Yu, S.: NFA to DFA transformation for finite language over arbitrary alphabets. J. Autom., Lang. Comb. 2(3), 177–186 (1997)
26. Sgarbas, K.N., Fakotakis, N.D., Kokkinakis, G.K.: Incremental construction of compact acyclic NFAs, 39th Annual Meeting of the Association for Computational Linguistics. pp. 482–489. Association for Computational Linguistics (2001)
27. Shannon, C.E.: The synthesis of two-terminal switching circuits. Bell Systems Technical Journal 28(1), 59–98 (1949)
28. Yu, S.: Cover automata for finite language. Bull. EATCS 92, 65–74 (2007)