Finding Lower Bounds for Nondeterministic State Complexity is Hard

Hermann Gruber and Markus Holzer

Institut für Informatik, Technische Universität München,
Boltzmannstraße 3, D-85748 Garching bei München, Germany
email: {gruberh,holzer}@informatik.tu-muenchen.de

Abstract. We investigate the following lower bound methods for regular languages: The fooling set technique, the extended fooling set technique, and the biclique edge cover technique. It is shown that the maximal attainable lower bound for each of the above mentioned techniques can be algorithmically deduced from a canonical finite graph, the so called *dependency graph* of a regular language. This graph is very helpful when comparing the techniques with each other and with nondeterministic state complexity. In most cases it is shown that for any two techniques the gap between the best bounds can be arbitrarily large. The only exception is the biclique edge cover technique which is always as good as the logarithm of the deterministic or nondeterministic state complexity. Moreover, we show that deciding whether a certain lower bound w.r.t. one of the investigated techniques can be achieved is in most cases computationally hard, i.e., PSPACE-complete and hence is as hard as minimizing nondeterministic finite automata.

1 Introduction

Finite automata are one of the oldest and most intensely investigated computational models. It is well known that deterministic and nondeterministic finite automata are computationally equivalent, and that nondeterministic finite automata can offer exponential state savings compared to deterministic ones [19]. Nevertheless, some challenging problems of finite automata are still open. For instance, to estimate the size, in terms of the number of states, of a minimal nondeterministic finite automaton for a regular language is stated as an open problem in [1] and [12]. This is contrary to the deterministic case, where for a given n-state deterministic automaton the minimal automaton can be efficiently computed in $O(n \log n)$ time. Observe that computing a state minimal nondeterministic finite automaton is known to be PSPACE-complete [15]. Moreover, it has been shown that upper or lower bounds on the state size of minimal nondeterministic automata with a guaranteed relative error better than $\frac{\sqrt{n}}{\mathrm{poly}(\log(n))}$ cannot be obtained in polynomial time, provided some cryptographic assumption holds [8].

Several authors have introduced communication complexity methods for proving such lower bounds; see, e.g., [3, 7, 11]. Although the bounds provided by these techniques are not always tight and in fact can be arbitrarily worse compared to the nondeterministic state complexity, they give good results in many cases. In this paper we investigate the fooling set technique [7], the extended fooling set technique [3, 11], and the biclique edge cover technique. Note that the latter method is an alternative representation of the nondeterministic message complexity [11]. One drawback of all these methods is that getting such a good estimate seems to require conscious thought and "clever guessing." However, we show for the considered techniques that this is in fact *not* the case. In order to achieve this goal, we present a unified view of these techniques in terms of bipartite graphs. This setup allows us to show that there is a canonical bipartite graph for each regular language, which is independent of the considered method, such that the best attainable lower bound can be determined algorithmically for each method. This canonical bipartite graph is called the *dependency graph* of the language.

The dependency graph is a tool that allows us to compare the relative strength of the methods, and to determine whether they provide a guaranteed relative error w.r.t. the nondeterministic state complexity. Following [1], no lower bound technique is known to have such a bounded error, but a lower bound can be obtained by noticing that the numbers of states in minimal deterministic automata and in minimal nondeterministic automata are at most exponentially apart from each other. We are able to prove that the biclique edge cover technique always gives an estimate at least as good as this trivial lower bound, whereas the other methods cannot provide any guaranteed relative error. On the other hand, we give evidence that the guarantee for the biclique edge cover technique is essentially optimal. In turn we improve a result of [13, 16] on the gap between nondeterministic message complexity and nondeterministic state complexity.

Finally, we also address computational complexity issues and show that deciding whether a certain lower bound w.r.t. one of the investigated techniques can be achieved is in most cases computationally hard, i.e, PSPACE-complete and hence these problems are as hard as minimizing nondeterministic finite automata. Here it is worth mentioning that the presented algorithms for the upper bounds also rely on the dependency graph, whose vertices are the equivalence classes of the Myhill-Nerode relation for the language L and its reversal L^R. Hence, doing the computation on this object in a straightforward manner would result in an exponential time algorithm. This is due to the fact that the index of the Myhill-Nerode equivalence relation for L^R can be exponential in terms of the index of the Myhill-Nerode relation for L, or equivalently to the size of the minimal deterministic finite automaton accepting L. Nevertheless, by cleverly encoding the equivalence classes we succeed in implicitly representing the dependency graph, which finally results in PSPACE-algorithms for the problems under consideration.

The paper is organized as follows: In the next section we define the basic notions. Section 3 introduces the three lower bound techniques we are interested in. Then the dependency graph is defined in Section 4 and based on this graph the question "How good is a lower bound induced by one of these lower bound techniques?" is answered in Section 5. The last section is devoted to computational complexity considerations on how to compute witnesses (fooling sets, extended fooling sets, biclique edge covers) for a certain lower bound technique.

2 Definitions

We assume the reader to be familiar with the basic notations in formal language and automata theory as contained in [10]. In particular, let Σ be an alphabet and Σ^* the set of all words over the alphabet Σ, including the empty word λ. The length of a word w is denoted by $|w|$, where $|\lambda| = 0$. The reversal of a word w is denoted by w^R and the reversal of a language $L \subseteq \Sigma^*$ by L^R, which equals the set $\{ w^R \mid w \in L \}$.

A *nondeterministic finite automaton* is a 5-tuple $A = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, Σ is a finite set of input symbols, $\delta : Q \times \Sigma \to 2^Q$ is the transition function, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of accepting states. The transition function δ is extended to a function from $\delta : Q \times \Sigma^* \to 2^Q$ in the natural way, i.e., $\delta(q, \lambda) = \{q\}$ and $\delta(q, aw) = \bigcup_{q' \in \delta(q,a)} \delta(q', w)$, for $q \in Q$, $a \in \Sigma$, and $w \in \Sigma^*$. The *language accepted* by A is

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

Two automata are equivalent if they accept the same language.

A nondeterministic finite automaton $A = (Q, \Sigma, \delta, q_0, F)$ is *deterministic* 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 $\delta(q, a) = \{p\}$. By the powerset construction one can show that every nondeterministic finite automaton can be converted into an equivalent deterministic finite automaton by increasing the number of states from n to 2^n; this bound is known to be sharp [18]. Thus, deterministic and nondeterministic finite automata are equally powerful.

For a regular language L, the deterministic (nondeterministic, respectively) state complexity of L, denoted by $\mathrm{sc}(L)$ ($\mathrm{nsc}(L)$, respectively) is the minimal number of states needed by a deterministic (nondeterministic, respectively) finite automaton accepting L. Observe that the minimal deterministic finite automaton is isomorphic to the deterministic finite automaton induced by the Myhill-Nerode equivalence relation $\equiv_L$, which is defined 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 $w \in \Sigma^*$. Hence, the number of states of the minimal deterministic finite automaton accepting the language $L \subseteq \Sigma^*$ equals the index, i.e., the cardinality of the set of equivalence classes, of the Myhill-Nerode equivalence relation $\equiv_L$. The set of all equivalence classes w.r.t $\equiv_L$ is referred to $\Sigma^*/\equiv_L$ and we denote the equivalence class of a word u w.r.t. the relation $\equiv_L$ by $[u]_L$. Moreover, we define the relation ${}_L\equiv$ as follows: For $u, v \in \Sigma^*$ let $u_L \equiv$ if and only if $wu \in L \iff wv \in L$, for all $w \in \Sigma^*$. The set of all equivalence classes w.r.t. ${}_L\equiv$ is referred to $\Sigma^*/_L\equiv$ and we denote the equivalence class of a word u w.r.t. the relation ${}_L\equiv$ by ${}_L[u]$.

Finally, we recall two remarkably simple lower bound techniques for the nondeterministic state complexity of regular languages. Both methods are commonly called *fooling set* techniques and were introduced in [3] and [7]. Although the difference in both theorems look quite harmless, the two techniques are essentially different.

Theorem 1 (Fooling Set and Extended Fooling Set Technique). *Let $L \subseteq \Sigma^*$ be a regular language and suppose there exists a set of pairs $S = \{ (x_i, y_i) \mid 1 \leq i \leq n \}$ with the following properties:*

1. *If (i) $x_i y_i \in L$ for $1 \leq i \leq n$, (ii) $x_i y_j \notin L$, for $1 \leq i, j \leq n$, and $i \neq j$, then any nondeterministic finite automaton accepting L has at least n states, i.e., $nsc(L) \geq n$. Here S is called a* fooling set *for L.*
2. *If (i) $x_i y_i \in L$ for $1 \leq i \leq n$, (ii) and $i \neq j$ 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 accepting L has at least n states, i.e., $nsc(L) \geq n$. Here S is called an* extended fooling set *for L.*

Note that the lower bounds provided by these techniques are not always tight and in fact can be arbitrarily bad compared to the nondeterministic state complexity. Nevertheless, they give good results in many cases—for the fooling set technique see the examples provided in [7].

3 Lower Bound Techniques and Bipartite Graphs

In this section we develop a unified view of fooling sets and extended fooling sets in terms of bipartite graphs and introduce a technique that leverages the shortcomings of the fooling set techniques. We need some notations from graph theory.

A *bipartite graph* is a 3-tuple $G = (X, Y, E)$, where X and Y are the (not necessarily finite, or disjoint) sets of vertices, and $E \subseteq X \times Y$ is the set of edges. A bipartite graph $H = (X', Y', E')$ is a *subgraph* of G if $X' \subseteq X$, $Y' \subseteq Y$, and $E' \subseteq E$. The subgraph H' is *induced* if $E' = (X' \times Y') \cap E$. Given a set of edges E', the *subgraph induced* by E' w.r.t. E is the smallest induced subgraph containing all edges in E'.

The relation between fooling sets and graphs is quite natural, because a (extended) fooling set S can be interpreted as the edge set of a bipartite graph $G = (X, Y, S)$ with $X = \{ x \mid$ there is a y such that $(x, y) \in S \}$ and $Y = \{ y \mid$ there is a x such that $(x, y) \in S \}$. In case S is a fooling set, the induced bipartite graph is nothing other than a ladder, i.e., a collection of pairwise vertex-disjoint edges. More generally, the notation of (extended) fooling sets carries over to bipartite graphs as follows: Let $G = (X, Y, E)$ be a bipartite graph.

1. Then a set $S \subseteq E$ is a fooling set for G, if for every two different edges e_1 and e_2 in S, the subgraph *induced* by the edges e_1 and e_2 w.r.t. E is the rightmost graph of Figure 1,
2. and a set $S \subseteq E$ is an extended fooling set for G, if for every two different edges e_1 and e_2 in S, the subgraph *induced* by the edges e_1 and e_2 w.r.t. E is one of the graphs depicted in Figure 1.

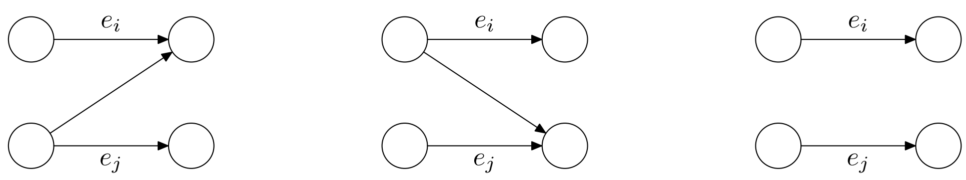

Fig. 1. Three important bipartite (sub)graphs.

Now let us associate to any language $L \subseteq \Sigma^*$ and sets $X, Y \subseteq \Sigma^*$ a bipartite graph $G = (X, Y, E_L)$, where $(x, y) \in E_L$ if and only if $xy \in L$, for every $x \in X$ and $y \in Y$. Then it is easy to see that the following statement holds—we omit the straight forward proof.

Theorem 2. *Let $L \subseteq \Sigma^*$ be a regular language. Then the set S is a (extended, respectively) fooling set for L if and only if the edge set $S \subseteq E_L$ is a (extended, respectively) fooling set for the bipartite graph $G = (\Sigma^*, \Sigma^*, E_L)$.* □

For the lower bound technique to come we need the notion of a biclique edge cover for bipartite graphs. Let $G = (X, Y, E)$ be a bipartite graph. A set $C = \{H_1, H_2, \ldots\}$ of non-empty bipartite subgraphs of G is an *edge cover* of G if every edge in G is present in at least one subgraph. An edge cover C of the bipartite graph G is a *biclique edge cover* if every subgraph in C is a biclique, where a *biclique* is a bipartite graph $H = (X, Y, E)$ satisfying $E = X \times Y$. The *bipartite dimension* of G is denoted $d(G)$ and is defined to be the size of the smallest biclique edge cover of G if it exists and is infinite otherwise. Then the biclique edge cover technique reads as follows—this technique is a reformulation of the nondeterministic message complexity method [11] in terms of graphs:

Theorem 3 (Biclique Edge Cover Technique). *Let $L \subseteq \Sigma^*$ be a regular language and suppose there exists a bipartite graph $G = (X, Y, E_L)$ with $X, Y \subseteq \Sigma^*$ (not necessarily finite) for the language L. Then any nondeterministic finite automaton accepting L has at least as many states as the bipartite dimension of G, i.e., $nsc(L) \geq d(G)$.*

Proof. Let $A = (Q, \Sigma, \delta, q_0, F)$ be any nondeterministic finite automaton accepting L. We show that every finite automaton induces a finite size biclique edge cover of the bipartite graph G. For each state $q \in Q$ let $H_q = (X_q, Y_q, E_q)$ with $X_q = X \cap \{\, w \in \Sigma^* \mid \delta(q_0, w) \ni q \,\}$, $Y_q = Y \cap \{\, w \in \Sigma^* \mid \delta(q, w) \cap F \neq \emptyset \,\}$, and $E_q = X_q \times Y_q$. We claim that $C = \{\, H_q \mid q \in Q \,\}$ is a biclique edge cover for G. By definition each H_q, for $q \in Q$, is a biclique. Moreover, each bipartite graph H_q is a subgraph of G. Since by construction $X_q \subseteq X$ and $Y_q \subseteq Y$ it remains to show that $E_q \subseteq E$. To this end assume that $x \in X_q$ and $y \in Y_q$. Then the word xy belongs to the language L because $q \in \delta(q_0, x)$ and $\delta(q, y) \cap F \neq \emptyset$. But then (x, y) is an edge of G. Finally, we must prove that C is an edge cover. Let (x, y) be an edge in G, for $x \in X$ and

$y \in Y$. Then the word xy is in L and since the nondeterministic finite automaton A accepts the language L, there is a state q in Q such that $q \in \delta(q_0, x)$ and $\delta(q, y) \cap F \neq 0$. Therefore $x \in X_q$ and $y \in Y_q$ and moreover (x, y) is an edge in E_q, because H_q is a biclique. This proves that C is a biclique edge cover of G.

Now assume that there is a nondeterministic finite automaton accepting L which number of states is strictly less than the bipartite dimension of G. Then this automaton induces a biclique edge cover C of G, which size is bounded by the number of states and thus is also strictly less than the bipartite dimension of G. This is a contradiction because the bipartite dimension is defined to be the size of the smallest biclique edge cover. Therefore any nondeterministic finite automaton accepting L has at least the bipartite dimension of G number of states. □

By the previous theorem we obtain the following corollary.

Corollary 4. *Let L be a regular language over the alphabet Σ. Then the bipartite graph $G = (\Sigma^*, \Sigma^*, E_L)$ has finite bipartite dimension.* □

4 The Dependency Graph of a Language

In applying the lower bound theorems from the previous section to any particular language it is necessary to choose pairs (x_i, y_i) or sets X and Y appropriately. For fooling sets a heuristic,[1] which of course also applies to the other techniques, was proposed in [7] and seems to work well in most cases. In fact, we show that such a heuristic is *not* needed. To this end we define the following bipartite graph:

Definition 5. *Let $L \subseteq \Sigma^*$. Then the* dependency graph *for the language L is defined to be the bipartite graph $G_L = (X, Y, E_L)$, where $X = \Sigma^*/{\equiv_L}$ and $Y = \Sigma^*/{}_L{\equiv}$ and $([x]_L, {}_L[y]) \in E_L$ if and only if $xy \in L$.*

It is easy to see that the dependency graph G_L for a language L is independent of the chosen representation of the equivalence classes. Hence all these graphs are isomorphic to each other. Moreover, it is worth mentioning that the dependency graph of a language was implicitly defined in [17]. Now we are ready to state the main lemma of this section.

Lemma 6. *Let $L \subseteq \Sigma^*$ be a regular language and $G = (\Sigma^*, \Sigma^*, E_L)$ its associated bipartite graph.*

1. *The maximum size of a (extended, respectively) fooling set for G is n if and only if the maximum size of a (extended, respectively) fooling set for the dependency graph G_L equals n.*
2. *The bipartite dimension of G is n if and only if the bipartite dimension of the dependency graph G_L equals n.*

Proof. We only prove the first statement. The second statement can be shown with similar arguments.

Let $S = \{ (u_i, v_i) \mid 1 \leq i \leq n \}$ be a (extended) fooling set for the bipartite graph $G = (\Sigma^*, \Sigma^*, E_L)$. By definition any two different edges in S are vertex-disjoint, if S is interpreted as a subset of E_L. Moreover, we find that any two different edges (u_i, v_i) and (u_j, v_j) obey $u_i \not\equiv_L u_j$ and $v_i \,{}_L{\not\equiv}\, v_j$. Otherwise the (extended) fooling set property is not

[1] In [7] the following heuristic is proposed: "Construct a nondeterministic finite automaton $A = (Q, \Sigma, \delta, q_0, F)$ accepting L, and for each state q in Q let x_q be the shortest string such that $\delta(q_0, x_q) = q$, and let y_q be the shortest string such that $\delta(q, y_q) \cap F \neq \emptyset$. Then choose the set S to be some *appropriate subset* of the pairs $\{ (x_q, y_q) \mid q \in Q \}$."

satisfied. Thus, the idea to obtain the finite bipartite graph that mirrors all relevant properties of G is to replace the vertex sets by the corresponding equivalence classes.

The construction is done in two steps. Any edge (u_i, v_i) in G can be replaced by (u'_i, v_i) whenever $u_i \equiv_L u'_i$. Thus, the "left vertices" in G can be replaced by an essential set of words x_i pairwise nonequivalent with respect to $\equiv_L$. Since L is regular, this set is finite. To conclude the first step, the bipartite graph G' is defined as the subgraph induced by the vertex set (X, Σ^*), where $X = \{ x_i \mid 1 \leq i \leq m \}$ and m is the index of $\Sigma^*/\equiv_L$. The (extended) fooling set S is updated accordingly. We denote this (extended) fooling set by S'. Note that S and S' are of same size. For the second step we argue as follows: Define the equivalence relation $\sim_X$ on Σ^* by $v \sim_X v'$ if and only if $xv \in L \iff xv' \in L$, for all $x \in X$. We show that this relation is the same as the relation ${}_L\equiv$. By definition $v \sim_X v'$ implies $v \,{}_L\equiv v'$. Conversely, let $v \,{}_L\equiv v'$. For each $u \in \Sigma^*$ we have $uv \in L \iff [u]_L v \subseteq L$. Thus we conclude $[u]_L v \subseteq L$ if and only if $uv \in L$ iff $uv' \in L$ if and only if $[u]_L v' \subseteq L$. Hence $v \sim_X v'$. This shows that $\sim_X$ is just an alternative formulation of ${}_L\equiv$, and we can apply a similar replacement procedure as in the first step, now for the "right vertices" in G' using the relation $\sim_X$. This results in a bipartite graph G'', which is defined as the subgraph induced by the vertex set (X, Y), where Y is chosen in a similar way as the x_i's above, but now w.r.t. the equivalence relation $\Sigma^*/{}_L\equiv$. Similarly we modify the (extended) fooling set S' and obtain the set S''. It is easy to see that S'' is in fact a (extended) fooling set for G'', and that it is of the same size as the original (extended) fooling set S. This completes the construction. □

An immediate consequence of the previous lemma is that finding the best possible lower bound for the technique under consideration is indeed solvable in an algorithmic manner. For instance, a fooling set corresponds to an *induced matching* [5] in G_L, and an extended fooling set to a *cross-free matching* [6] in G_L, and *vice versa*. The drawback of the dependency graph G_L is that its size can be exponential in terms of the state complexity of the deterministic finite automaton for the language [20].

5 How Good are the Lower Bounds Induced by These Techniques?

We compare the introduced techniques with each other w.r.t. the lower bounds that can be obtained in the best case and to the nondeterministic state complexity. The first theorem shows that the bound based on the biclique edge cover technique can be seen as a generalization of the extended fooling set technique. Due to the lack of space we omit the proof and refer to the Appendix—see also Example 15.

Theorem 7. *Let L be a regular language. Then the bipartite dimension of the dependency graph G_L is equal to or greater than the maximum size of an extended fooling set for L.* □

When comparing the techniques under consideration we obtain the following result—see the Appendix for further details again.

Theorem 8. *There is a sequence of languages $(L_n)_{n\geq 1}$ such that the nondeterministic state complexity of L_n is at least n, i.e., $nsc(L_n) \geq n$, but any fooling set for L has size at most c, for some constant c. An analogous statement holds for extended fooling sets versus fooling sets, nondeterministic state complexity versus extended fooling sets, and the bipartite dimension versus extended fooling sets.*

As the reader may have noticed, the comparison between bipartite dimension and nondeterministic state complexity is missing in the previous theorem. The following theorem shows that the bipartite dimension of a regular language is a measure of descriptional complexity.

Theorem 9. *Let $L \subseteq \Sigma^*$ be a regular language and G_L the dependency graph for L. Then $2^{d(G_L)}$ is greater or equal to the* deterministic *state complexity of L, i.e., $2^{d(G_L)} \geq sc(L)$.*

Proof. Let $G_L = (X, Y, E_L)$ and assume that the bipartite dimension of G_L equals k. Then the edge set of G_L can be covered by a set of bicliques $C = \{H_1, H_2, \ldots, H_k\}$. For $x \in \Sigma^*$, let $B(x) \subseteq C$ be the set of bicliques where x occurs as a "left vertex." We claim that $B(x) = B(x')$ implies $x \equiv_L x'$, for all $x, x' \in \Sigma^*$. Suppose that $y \in \Sigma^*$ occurs as a "right vertex" in some biclique in $B(x)$. If $B(x) = B(x')$, then both $(x, y) \in E_L$ and $(x', y) \in E_L$. The other possibility is that y does not occur as a right vertex in any biclique from $B(x)$. Then $B(x) = B(x')$ implies that $(x, y) \notin E_L$ and $(x', y) \notin E_L$. By definition of G_L we have $(x, y) \in E_L$ if and only if $xy \in L$. To conclude, if $B(x) = B(x')$, then $xy \in L \iff x'y \in L$, for all $y \in \Sigma^*$, which is the definition of the Myhill-Nerode equivalence $\equiv_L$ of L. Then define $x \sim x'$ with $x, x' \in \Sigma^*$ if and only if $B(x) = B(x')$. This equivalence relation induces $2^{|C|}$ equivalence classes, and is a refinement of the Myhill-Nerode relation. Thus we have shown that $2^{|C|}$ is greater or equal to the deterministic state complexity of L. □

Hence, $d(G_L) \geq \log \mathrm{sc}(L)$ and $d(G_L) \geq \log \mathrm{nsc}(L)$. By Corollary 4 and Theorem 9 we obtain a characterization of regular languages in terms of bipartite dimension.

Corollary 10. *Let $L \subseteq \Sigma^*$ be an arbitrary language and $G = (\Sigma^*, \Sigma^*, E_L)$ the bipartite graph associated with L. Then L is a regular language if and only if $d(G)$ is finite.* □

The above result is essentially optimal. In [13, 16] it was shown that the nondeterministic state complexity can be $\Omega(2^{\sqrt{d}})$, where d is the bipartite dimension of the dependency graph. We improve on this result showing that this gap can be actually even larger using the languages $L_n = \{\, w \in 0^* \mid |w| \neq 0 \mod n\}$. The proof can be found in the Appendix.

Theorem 11. *There is a sequence of languages $(L_n)_{n\geq 1}$ over a one letter alphabet such that $nsc(L_n) = \Omega\left(d_n^{-1/2} \cdot 2^{d_n}\right)$, where d_n is the bipartite dimension of G_{L_n}.* □

6 Computational Complexity of Lower Bound Techniques

To determine the nondeterministic state complexity of a regular language is known to be a computationally hard task, namely PSPACE-complete [15]. In this section we consider three decision problems based on the lower bound techniques presented so far. The *fooling set problem* is defined as follows:

– Given a deterministic finite automaton A and a natural number k in binary, i.e., an encoding $\langle A, k\rangle$.
– Is there a fooling set S for the language $L(A)$ of size at least k?

The *extended fooling set* and the *biclique edge cover problem* are analogously defined. We start our investigations with the fooling set problem.

Theorem 12. *The fooling set problem is NP-hard and contained in PSPACE.*

Proof. For the NP-hardness we reduce the NP-complete induced matching problem on bipartite graphs [5] to the problem under consideration. The induced matching problem on bipartite graphs is defined as follows: Given a bipartite graph $G = (X, Y, E)$ and an integer k encoded in binary, does E contain an induced matching of size at least k?

Let $\langle G, k\rangle$ be an instance of the induced matching problem for bipartite graphs. Assume $G = (X, Y, E)$ with $X = \{1, 2, \ldots, n\}$ and $Y = \{1, 2, \ldots, m\}$. Then we define the regular language $L_G = \{ a_i b_j \mid (i,j) \in E \}$ over the alphabet $\Sigma = \{a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m\}$. It is easy to see that there is a deterministic finite automaton A for L_G of size polynomial in n and m. Then one can easily verify that there is an induced matching for G of size at least k if and only if there is a fooling set for L_G of size at least $k+2$. Note that if M is an induced matching for G, then $S = \{ (a_i, b_j) \mid (i,j) \in M\} \cup \{(\lambda, w), (w, \lambda)\}$ is a maximum fooling set for L_G, where w is any word in L_G. Hence the induced matching problem for bipartite graphs reduces to the fooling set problem.

It remains to prove the containment within PSPACE. Let $\langle A, k\rangle$ be the instance of the fooling set problem, where $A = (Q, \Sigma, \delta, q_0, F)$ is a deterministic finite automaton and k an integer. If S is a fooling set, then one can assume w.l.o.g. that for every $(x,y) \in S$ we have $|x| \leq |Q|$ and $|y| \leq 2^{|Q|}$. Moreover we note that the size of S cannot exceed $|Q|$. This gives the idea to the following algorithm: A polynomially space bounded nondeterministic Turing machine can guess k words x_i with $|x_i| \leq |Q|$ and store the states $q_i = \delta(q_0, x_i)$ in a k-vector. Then the Turing machine guesses words y_i of length at most $2^{|Q|}$ in sequence, for $1 \leq i \leq k$, and verifies that the fooling set property is satisfied. Thus, for the word y_i the Turing machine checks whether $\delta(q_i, y_i) \in F$ and $\delta(q_j, y_i) \notin F$, if $i \neq j$. Of course, due to the space bounds, the machine cannot remember the whole of the word y_i, but it suffices to guess the word letter by letter and to update the k-state vector accordingly. This shows containment in PSPACE. □

Note that the upper bound for fooling sets as shown in Theorem 12 easily transfers to extended fooling sets. The PSPACE-hardness is shown by a reduction is from the PSPACE-complete deterministic finite automaton union universality problem, and closely follows that of the PSPACE-hardness of the nondeterministic finite automaton minimization problem [15]; see the Appendix for further details. Thus, we have:

Theorem 13. *The extended fooling set problem is PSPACE-complete.* □

Finally, we show that that deciding the biclique edge cover problem is also PSPACE-complete, although the dependency graph of the given language can be of exponential size in terms of the input.

Theorem 14. *The biclique edge cover problem is PSPACE-complete.*

Proof. The PSPACE-hardness follows along the lines of the proof for the PSPACE-completeness of the extended fooling set problem and the fact that the bipartite dimension is sandwiched between the maximum size of an extended fooling set and the nondeterministic state complexity.

The containment in PSPACE is seen as follows: We present a PSPACE algorithm deciding on input $\langle A, k\rangle$ whether there is a biclique edge cover of size at most k for $G_{L(A)}$. Since PSPACE is closed under complement, this routine can also be used to decide whether there is no biclique edge cover of size at most $k-1$, and moreover that the bipartite dimension of the graph is at least k.

Due to the space constraints, keeping the dependency graph $G_{L(A)}$ in memory is ruled out, since the index of ${}_{L(A)}{\equiv}$ can be exponential in the size of the given deterministic finite automaton. Recall that the vertex sets of $G_{L(A)}$ can be chosen to correspond to the equivalence classes of $\equiv_{L(A)}$ and ${}_{L(A)}\equiv$. So the first vertex set is in one-to-one correspondence with the state set Q of the automaton A, while by Brzozowski's theorem [4], the second vertex set corresponds one-to-one to a certain subset of 2^Q. Namely, for $A = (Q, \Sigma, \delta, q_0, F)$ let $A^R = (Q, \Sigma, \delta^R, F, \{q_0\})$, where $p \in \delta^R(q, a)$ if and only if $\delta(p, a) = q$, be a finite automaton with multiple initial states, the so called reversed automaton of A. Moreover, let $D(A^R)$ be the automaton obtained by applying the "lazy" subset construction to the automaton A^R, that is we generate only the subsets reachable from the set of start states of the finite state automaton A^R. Then these subsets of Q correspond to the equivalence classes of ${}_{L(A)}\equiv$. Since this automaton can be of size exponential in $|Q|$, however, it cannot be kept in the working memory, too. Nevertheless, assuming $Q = \{q_0, q_1, \dots q_{n-1}\}$, we can represent the subsets of Q as binary string of length n in a natural fashion. By these mappings, we may assume now that $G_{L(A)} = (X, Y, E_{L(A)})$ with $X = Q$, $Y = \{0,1\}^n$, and the suitably induced edge relation $E_{L(A)}$. Thus, we have established a compact representation of the vertices in the dependency graph. Next, we need a routine to decide membership in the edge set of $G_{L(A)}$.

Given the implicit representation of $G_{L(A)}$ in terms of a n-state deterministic finite automaton $A = (Q, \Sigma, \delta, q_0, F)$, there is a PSPACE algorithm deciding, given a state q of A and a subset address $s = a_0 a_1 \dots a_{n-1}$, whether $(q, s) \in E_{L(A)}$. Assume x to be a word satisfying $\delta(q_0, x) = q$. As $|x| \leq n$, it can be determined and stored to the work tape without affecting the space bounds. If s corresponds to a reachable subset M in $D(A^R)$, then we can guess on the fly a word y of length at most 2^n, and verify that M is reached in $D(A^R)$ by reading y. Now, (q, s) is an edge in $G_{L(A)}$ if and only if xy^R is in $L(A)$. This is the case if and only if $(xy^R)^R = yx^R$ is accepted by $D(A^R)$. Recall that the word y may be of exponential length and cannot be directly stored on the work tape. But $D(A^R)$ is in the state set M after reading y, and we only have to verify that we reach an accepting state if we continue by reading x^R. This is the desired subroutine for deciding whether $(q, s) \in E_{L(A)}$, which runs in (nondeterministic) polynomial space.

The next obstacle is that, although there surely exists a biclique edge cover of cardinality at most n for $G_{L(A)}$, a single biclique in this cover can be of exponential size. Thus we have to reformulate the biclique edge cover problem in a suitable manner. Let $G = (X, Y, E)$ be a bipartite graph, and for $y \in Y$ define $\Gamma(y) = \{\, x \in X \mid (x, y) \in E \,\}$. Then the formula

$$\exists C \subseteq 2^X : |C| \leq k \wedge \big(\forall (x, y) \in E : \exists c \in C : x \in c \wedge c \subseteq \Gamma(y)\big) \qquad (1)$$

is a statement equivalent to the biclique edge cover problem. This is seen as follows: Assume C is a set of at most k subsets of X satisfying the above conditions. We construct a set of $|C|$ bicliques covering all edges in G. For $c \in C$, let c' be the set of vertices in Y such that $\Gamma(y) \supseteq c$. Then (c, c') induces a biclique in G, since every vertex in c is adjacent to all vertices in c'. Furthermore, the condition on C ensures that every edge is a member of least one such biclique, and we have obtained a biclique edge cover of size at most k. Conversely, assume that $\{H_1, H_2, \dots, H_k\}$ is a biclique edge cover of size k for G, where $H_i = (c_i, c_i', c_i \times c_i')$ for $1 \leq i \leq k$. We set $C = \{c_1, c_2, \dots c_k\}$. Then for every edge (x, y) in G, there is a $c \in C$ such that $x \in c$ and $c \subseteq \Gamma(y)$. If H_i is a biclique covering of the edge (x, y), then obviously $x \in c_i$ and y is adjacent to all vertices in c_i. This proves the stated claim on Equation (1).

Now let us come back to the input $\langle A, k\rangle$, where $A = (Q, \Sigma, \delta, q_0, F)$. The reformulated statement can be checked in PSPACE by guessing a set C of at most k subsets of Q, and then the Turing machine checks the following for each pair $(x, y) \in X \times Y$, where X and Y is chosen as described above: If $(x, y) \notin E_{L(A)}$ it goes to the next pair. Otherwise, it guesses a subset $c \in C$ and verifies that both $x \in C$ and that for every $x' \in c$, $(x', y) \in E_{L(A)}$. By our previous investigations it is easy to see that this algorithm can be implemented on a nondeterministic polynomial space bounded Turing machine. This proves that the biclique edge cover problem belongs to PSPACE. □

Finally, let us mention that the complexity of the fooling set and the extended fooling set problem does not increase if the regular language is specified as a nondeterministic finite automaton. The proofs for the upper bounds on the complexity carry over to this setup with minor modifications. Currently, we do not know whether this also true for the biclique edge cover technique, if the regular language is given as a nondeterministic finite automaton. The best upper bound we are aware of is co-NEXPTIME, obtained by explicit construction of G_L and verifying that there is no biclique edge cover of size at most k.

Acknowledgments Thanks to Martin Kutrib for some discussion on the subject during the early stages of the paper.

References

1. H. N. Adorna. Some descriptional complexity problems in finite automata theory. In R. P. Salde na and C. Chua, editors, *Proceedings of the 5th Philippine Computing Science Congress*, pages 27–32, Cebu City, Philippines, March 2005. Computing Society of the Philippines.
2. S. Bezrukov, D. Frončck, S. J. Rosenberg, and P. Kovář. On biclique coverings. Preprint, 2005.
3. J.-C. Birget. Intersection and union of regular languages and state complexity. *Information Processing Letters*, 43:185–190, 1992.
4. J. A. Brzozowski. Mathematical theory of automata. In *Canonical Regular Expressions and Minimal State Graphs for Definite Events*, volume 12 of *MRI Symposia Series*, pages 529–561. Polytechnic Press, NY, 1962.
5. K. Cameron. Induced matchings. *Discrete Applied Mathematics*, 24:97–102, 1989.
6. M. Dawande. A notion of cross-perfect bipartite graphs. *Information Processing Letters*, 88:143–147, 2003.
7. I. Glaister and J. Shallit. A lower bound technique for the size of nondeterministic finite automata. *Information Processing Letters*, 59:75–77, 1996.
8. G. Gramlich and G. Schnitger. Minimizing NFA's and regular expressions. In V. Diekert and B. Durand, editors, *Proceedings of the 22nd Annual Symposium on Theoretical Aspects of Computer Science*, number 3404 in LNCS, pages 399–411, Stuttgart, Germany, February 2005. Springer.
9. P. L. Hammer and P. C. Fishburn. Bipartite dimension and bipartite degrees of graphs. *Discrete Applied Mathematics*, 160:127–148, 1996.
10. J. E. Hopcroft and J. D. Ullman. *Formal Languages and Their Relation to Automata.* Addison-Wesley, 1968.
11. J. Hromkovič. *Communication Complexity and Parallel Computing.* Springer, 1997.
12. J. Hromkovič. Descriptional complexity of finite automata: Concepts and open problems. *Journal of Automata, Languages and Combinatorics*, 7(4):519–531, 2002.
13. J. Hromkovič, J. Karhumäki, H. Klauck, G. Schnitger, and S. Seibert. Measures of nondeterminism in finite automata. Report TR00-076, Electronic Colloquium on Computational Complexity (ECCC), 2000.
14. T. Jiang, E. McDowell, and B. Ravikumar. The structure and complexity of minimal NFAs over unary alphabet. *International Journal of Foundations of Computer Science*, 2(2):163–182, June 1991.
15. T. Jiang and B. Ravikumar. Minimal NFA problems are hard. *SIAM Journal on Computing*, 22(6):1117–1141, December 1993.
16. G. Jirásková. Note on minimal automata and uniform communication protocols. In C. Martín-Vide and V. Mitrana, editors, *Grammars and Automata for String Processing*, volume 9 of *Topics in Computer Mathematics*, pages 163–170. Taylor and Francis, 2003.
17. T. Kameda and P. Weiner. On the state minimization of nondeterministic finite automata. *IEEE Transactions on Computers*, C-19(7):617–627, 1970.
18. F. R. Moore. On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata. *IEEE Transaction on Computing*, C-20:1211–1219, 1971.
19. M. O. Rabin and D. Scott. Finite automata and their decision problems. *IBM Journal*, 3:114, 1959.
20. A. Salomaa, D. Wood, and S. Yu. On the state complexity of reversals of regular languages. *Theoretical Computer Science*, 320(2–3):315–329, 2004.

Appendix

Proof (of Theorem 7). The proof of this fact is entirely graph theoretic. We need some further notations from graph theory: An *undirected simple graph* is a tuple $\Gamma = (V, E)$, where V is the set of vertices and $E \subseteq \{ \{u, v\} \mid u, v \in V \text{ and } u \neq v \}$ the set of edges. A set $C \subseteq V$ of vertices is a *clique*, if $\{v, v'\} \in E$ for all vertices $v, v' \in C$. The *clique number* of G, denoted by $\omega(\Gamma)$, is the maximum size of a clique in Γ. A coloring of the vertex set is an assignment of a color to each vertex in a way such that each pair of vertices sharing an edge receives a different color. The *chromatic number* $\chi(\Gamma)$ is then the least number of colors needed in order to color the vertex set.

As it turns out, the bipartite dimension of a bipartite graph G can be determined in an associated undirected simple graph *via* the following result, which is due to [9]. Let $G = (X, Y, E)$ a bipartite graph and Γ_G be its associated undirected simple graph whose vertex set is the edge set of G, and for each pair of vertex-disjoint edges $e_i = (x_i, y_i)$ and $e_j = (x_j, y_j)$ in G, let $\{e_i, e_j\}$ be an edge in Γ_G if and only if the subgraph induced by $(\{u_i, u_j\}, \{v_i, v_j\})$ is one of the constellations shown in Figure 1. Then the result in [9] reads as follows:

> Let $G = (X, Y, E)$ be a bipartite graph and Γ_G be its associated undirected simple graph. Then the bipartite dimension of G equals the chromatic number of Γ_G, i.e., $d(G) = \chi(\Gamma_G)$.

Next we show that a extended fooling sets correspond to cliques in the graph Γ_G, and thus are related to the clique number $\omega(\Gamma_G)$.

> Let $G = (X, Y, E)$ be a bipartite graph and $S \subseteq E$ a set of edges. Then S is an extended fooling set for G if and only if S is a clique in Γ_G.

We argue as follows: If S is an extended fooling set for G, then every pair of distinct edges $e_i, e_j \in S$ forms a constellation as in Figure 1, each giving rise to an edge $\{e_i, e_j\}$ in Γ_G. Thus, when S is seen as a set of vertices in Γ_G, then all members in S are pairwise connected by an edge in Γ_G. Thus, S is a clique in Γ_G. Conversely, assume S is not an extended fooling set for G. Then S contains a pair of edges $e_i = (x_i, y_i)$ and $e_j = (x_j, y_j)$ such that either (i) $x_i = x_j$ or $y_i = y_j$, or (ii) $(x_i, y_j) \in E$ and $(x_j, y_i) \in E$. In both cases, $\{e_i, e_j\}$ is not an edge in Γ_G, and S is not a clique in Γ_G. We immediately obtain the following relation:

> Let L be a regular language and G_L the dependency graph of L. Then the maximum size extended fooling set for L has size $\omega(\Gamma_{G_L})$.

The relation $\chi(\Gamma) \geq \omega(\Gamma)$ holds for any graph Γ because all vertices in a clique are mutually connected by edges, and thus indeed n colors are needed to color a clique of size n. Therefore by [9] the bipartite dimension of a bipartite graph is always equal to or greater than the maximum possible size of an extended fooling set. This completes the proof of the stated claim. □

Example 15. Consider the finite language $L = \{ab, ac, bc, ba, ca, cb\}$, which has nondeterministic state complexity five—see Figure 2. Then one can easily verify that, for instance,

$$S = \{(\lambda, ab), (ba, \lambda)\} \cup \{(a, b), (b, a)\}$$

is a fooling set and

$$S' = \{(\lambda, ab), (ba, \lambda)\} \cup \{(a, b), (b, c), (c, a)\}$$

an extended fooling set for L. Note that the size of S' exactly matches the nondeterministic state complexity of L, while S is one element off the optimum, but best possible w.r.t. the fooling set condition.

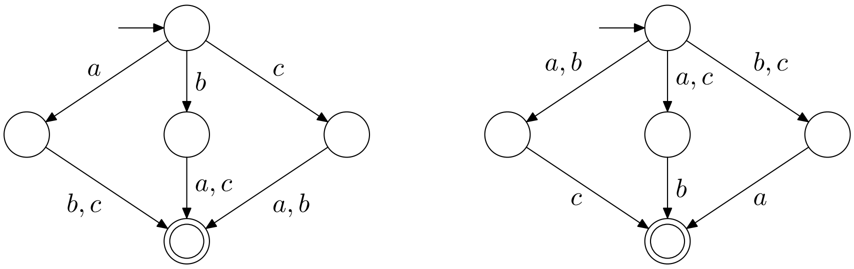

Fig. 2. Two non-isomorphic minimal nondeterministic finite automata for the finite language $L = \{ab, ac, bc, ba, ca, cb\}$.

For the latter statement it remains to be shown that there is no larger fooling set than S for L. To this end we argue as follows: (i) Any element of a fooling set for L is obviously of the form (λ, y), (x, λ) with $|x| = |y| = 2$, or (x, y) with $|x| = |y| = 1$. (ii) No three or more different pairs of the form (λ, y) or (x, λ) with $|x| = |y| = 2$ can be present simultaneously in any fooling set for L. Assume to the contrary that there are at least three pairs of this form. Then with loss of generality there are two pairs with first component λ. This contradicts the obvious fact that in any fooling set no two elements can be present with the same first or second component. (iii) No three or more different pairs of the form (x, y) with $|x| = |y| = 1$ can be present simultaneously in any fooling set for L. Assume to the contrary that there are at least three pairs of this form. Let (x_i, y_i) with $i \geq 3$ be these pairs satisfying $x_i y_i \in L$, for $|x_i| = |y_i| = 1$. Then from the fooling set property we conclude $x_1 y_3 \notin L$ and $x_2 y_3 \notin L$, and therefore $x_1 = y_3$ and $x_2 = y_3$. Thus we obtain $x_1 = x_2$, a contradiction, by similar reasons as above. This proves that the fooling set S is already of maximal size.

For the biclique edge cover technique we define the bipartite graph $G = (X, Y, E_L)$ for L with $X = \{\lambda, a, b, c, ab\}$, $Y = \{\lambda, a, b, c, ab\}$. The edge set E_L is depicted in Figure 3. This

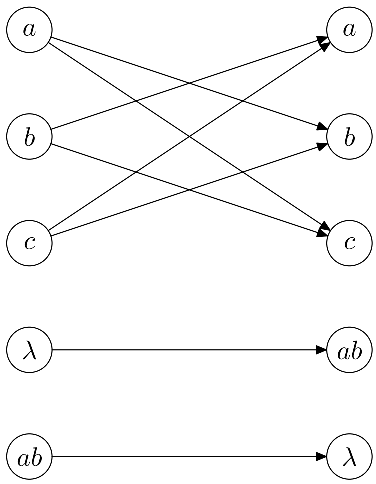

Fig. 3. The dependency graph G_L (only vertices are shown that are connected by edges) of the finite language $L = \{ab, ac, bc, ba, ca, cb\}$.

graph is the edge-disjoint union of two graphs G_1 and G_2, where G_1 is a 2-ladder and G_2 is the bi-complement of a 3-ladder. Clearly, the bipartite dimension of G_1 equals 2, and by [2] follows that the bipartite dimension of G_2 equals 3. Hence the bipartite dimension of $G = (X, Y, E_L)$ equals 5, which is optimal.

Figure 4 is an eye-catching proof of the fact that the finite language L has an extended fooling set of size 5 using the graph Γ_{G_L}—the 5-clique is outlined in boldface. The reader is invited to check that the depicted graph is indeed Γ_{G_L}, and to find a 5-coloring for this graph.

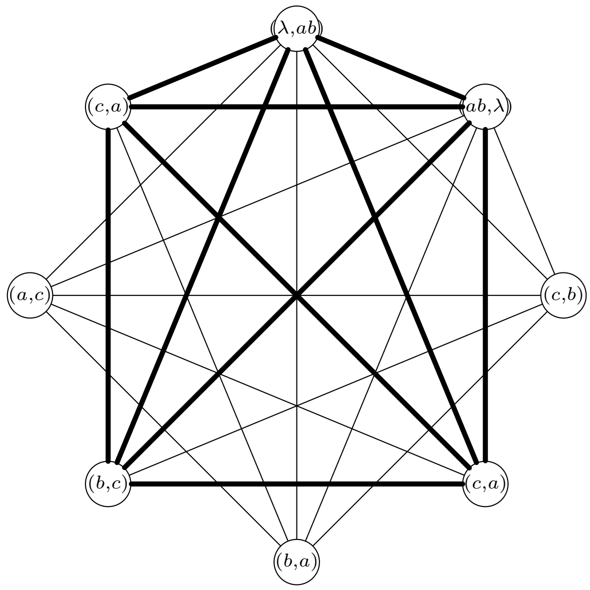

Fig. 4. The undirected simple graph Γ_{G_L} for the finite language $L = \{ab, ac, bc, ba, ca, cb\}$.

Proof (of Theorem 8).

Nondeterministic state complexity *versus* fooling sets: The statement is due to [7] and uses as witness languages $L_n = \{\, w \in 0^* \mid |w| = 0 \text{ or } |w| \neq 0 \mod n \,\}$.

Extended fooling sets *versus* fooling sets: Let $\Sigma = \{\, a_i \mid 1 \leq i \leq n \,\}$. Consider the finite language $L_n = \{\, a_i a_j \mid 1 \leq i \leq j \leq n \,\}$. It is easy to see that $S_n = \{\, (a_i, a_i) \mid 1 \leq i \leq n \,\} \cup \{(\lambda, a_1 a_1), (a_1 a_1, \lambda)\}$ is an extended fooling set for L_n of size at least $n + 2$. Then the analysis that any fooling set for L_n has size at most 3 goes as follows: (i) First one observes, that any fooling set can only contain at most one pair of the form (λ, w) or (w, λ). Then (ii) no two pairs of the form (a_i, a_j) and (a_k, a_ℓ) with $1 \leq i \leq j \leq n$ and $1 \leq k \leq \ell \leq n$ can be members of any fooling set for L_n. W.l.o.g. assume that $i \leq k$, then $a_i a_\ell \in L_n$, which contradicts the fooling set property. Hence, any fooling set for L_n can have at most 3 elements. This bound is tight, which is seen by the fooling set $S = \{(\lambda, a_1 a_1), (a_1 a_1, \lambda), (a_1, a_1)\}$ for the language L_n.

Nondeterministic state complexity *versus* extended fooling sets: Consider the unary language $L_n = \{\, w \in 0^* \mid |w| = 0 \text{ or } |w| \neq 0 \mod n \,\}$. We will show that there is no extended fooling set of size four for L_n. As any subset of a extended fooling set is again an extended fooling set, there cannot be an extended fooling set of larger cardinality either. The idea of the proof is that any subgraph of G_{L_n} induced by a set of four vertex-disjoint edges contains too many edges for being an extended fooling set for L_n.

For an arbitrary bipartite graph G, let $S = \{ (x_i, y_i) \mid 1 \leq i \leq 4 \}$ be an extended fooling set for G. Then for any pair (x_i, y_i) and (x_j, y_j) in S with $i \neq j$, there is at least one non-edge in the corresponding induced subgraph of G, namely (x_i, y_j) or (x_j, y_i). There are $\binom{4}{2} = 6$ such pairings, so the subgraph induced by S can have at most $4 \cdot 4 - 6 = 10$ edges.

Now let us turn to the dependency graph G_{L_n}. The cases $n \leq 3$ are trivial. Now assume $n \geq 4$. For the graph $G_{L_n} = (X, Y, E_L)$, we choose the representation

$$X = \{ 0^k \mid 1 \leq k \leq n \},$$
$$Y = \{ 0^{n-k} \mid 0 \leq k \leq n-1 \},$$

and define $(0^i, 0^j) \in E_L$, if $i + j \neq n$. Let $S = \{ (x_i, y_i) \mid 1 \leq i \leq 4 \}$ be a set of pairwise vertex-disjoint edges in G_{L_n}. Set $U = \{ x_i \mid 1 \leq i \leq 4 \}$ and $V = \{ y_i \mid 1 \leq i \leq 4 \}$, and let G the bipartite subgraph induced by (U, V). Out of the maximally 16 edges from U to V, at most four pairs (x_i, y_j) can be non-edges: Assume $|x_i| = k \mod n$. Then $(x_i, y_j) \notin E_L$ implies $|y_j| = (n-k) \mod n$. Two words of the same length modulo n are equivalent w.r.t. $\equiv_L$. Hence V contains only one word with this property. We conclude that for each element in U, there is at most one member y_j in V such that $(x_i, y_j) \notin E_L$, and the bipartite graph G has at least 12 edges. This contradicts our previous result that S induces a subgraph of at most 10 edges.

Finally it is easy to see that the nondeterministic state complexity of L_n grows with n and cannot be bounded by any constant.

Bipartite dimension *versus* extended fooling sets: Consider the language $L_n = \{ w \in 0^* \mid |w| = 0 \text{ or } |w| \neq 0 \mod n \}$ used in the proof above, where it was shown that any extended fooling for L_n has size at most 4. In order to prove the above stated result it suffices to show that the bipartite dimension of G_{L_n} grows with n and cannot be bounded by any constant.

For the dependency graph $G_{L_n} = (X, Y, E_L)$ we chose the same representation as in the proof above. Let $\overline{G}_{L_n} := (X, Y, (X \times Y) \setminus E_L)$ be the bi-complement of G_{L_n} w.r.t. the edge set E_L. Observe, that $\overline{G}_{L_n}$ is an induced matching with n edges, i.e., an n-ladder. The bipartite dimension of graphs with the property that their complement w.r.t. the edge set is an induced matching with n edges was determined in [2]. It was shown that it equals k, where k is the smallest integer such that $n \leq \binom{k}{\lfloor \frac{k}{2} \rfloor}$. So k grows with n, and cannot be bounded by any constant as n tends to infinity.

Proof (of Theorem 11). Let $L_n = \{ w \in 0^* \mid |w| \neq 0 \mod n \}$. As in the proof of Theorem 8 one can show that the bipartite dimension of the dependency graph G_{L_n} is the unique integer k such that

$$\binom{k-1}{\lfloor \frac{k-1}{2} \rfloor} < n \leq \binom{k}{\lfloor \frac{k}{2} \rfloor}.$$

By Stirling's approximation of the factorial $\binom{k-1}{\lfloor \frac{k-1}{2} \rfloor} = \Omega\left(2^k/\sqrt{k}\right)$ and we conclude that

$$n = \Omega\left(\frac{2^{d(G_{L_n})}}{\sqrt{d(G_{L_n})}}\right).$$

It remains to be shown that there are infinitely many n such that $\text{nsc}(L_n) \geq n$. We show that this is the case, whenever n is a prime number and thus taking the sequence $(L_{p_i})_{i \geq 1}$, where p_i is the ith prime number, will prove the stated result. To this end we argue as follows:

An unary language L is called *n-cyclic,* if $0^i \in L \iff 0^{i+n} \in L$, for every $i \geq 0$. Moreover, language L is *minimally n-cyclic,* if L is n-cyclic, but not m-cyclic for any $m < n$. In [14, Corollary 2.1] it was shown that if L is a minimally p-cyclic unary language, where p is prime, then $\mathrm{nsc}(L) = p$. We show that L_n is minimally n-cyclic. It can be readily seen that L_n is n-cyclic. Assume to the contrary that L_n is also m-cyclic with $m < n$. Then $\lambda \notin L_n$ implies $0^m \notin L_n$. But the shortest nonempty word not in the language has length n, a contradiction. Thus, the stated claim follows. □

Proof (of Theorem 13). The PSPACE-complete deterministic finite automaton union universality problem is defined as follows:

- Given a list of deterministic finite automata $A_1, A_2, \ldots, A_n$ over a common alphabet Σ.
- Is $\bigcup_{i=1}^{n} L(A_i) = \Sigma^*$?

Obviously, the deterministic finite automata union universality problem remains PSPACE-complete, if for given deterministic finite automata $A_1, A_2, \ldots, A_n$, all words of length at most one are in $\bigcup_{i=1}^{n} L(A_i)$.

The given construction relies on a definition of a special language L commonly specified by the multiple deterministic finite automata—recall the construction given in [15]. Let $\langle A_1, A_2, \ldots, A_n \rangle$ be the instance of the union universality problem for deterministic finite automata, where $A_i = (Q_i, \Sigma, \delta_i, q_{i1}, F_i)$, for $1 \leq i \leq n$ is a deterministic finite automaton with state set $Q_i = \{q_{i1}, q_{i,2}, \ldots q_{i,t_i}\}$. We assume that $Q_i \cap Q_j = \emptyset$ for $i \neq j$. The language $P(i,j)$ is defined as the set of words which could be accepted if q_{ij} was redefined as the only accepting state, that is $P(i,j) = \{\, w \in \Sigma^* \mid \delta(q_{i1}, w) = q_{ij} \,\}$. We introduce a new symbol a_i for each automaton A_i, and a new symbol b_{ij} for each state q_{ij} in $\bigcup_{i=1}^{n} Q_i$. In addition, we have new symbols c, d and f. Define the language $P(i)$ as a marked version of the language accepted by A_i:

$$P(i) = \bigcup_{j=1}^{t_i} [a_i \cdot P(i,j) \cdot b_{ij}].$$

The language $Q(i)$ consists of short prefixes of words in $L(A_i)$, which are marked at the end:

$$Q(i) = \{\, w b_{ij} \mid w \in (\Sigma \cup \lambda) \text{ and } \delta(q_{i1}, w) = q_{ij} \,\}.$$

Let B be the set of symbols b_{ij} introduced above. Then the auxiliary language R is given by

$$R = (\{c\} \cup \Sigma)(d \cup \Sigma)\Sigma^*(\{f\} \cup B).$$

Lastly, let

$$L = \bigcup_{i=1}^{n} [P(i) \cup a_i L(A_i) \cup Q(i)] \cup R \cup \Sigma^*. \tag{2}$$

Given $A_1, A_2, \ldots, A_n$, it is easy to construct in polynomial time

- a deterministic finite automaton with a single state accepting Σ^*,
- a deterministic finite automaton with four states accepting R,
- a deterministic finite automaton with $|\Sigma| + 2$ states accepting the language $\bigcup_{i=1}^{n} Q(i)$, and
- a deterministic finite automaton with $2 + \sum_{i=1}^{n} |Q_i|$ states accepting $\bigcup_{i=1}^{n} [P(i) \cup a_i L(M_i)]$.

By the well-known product construction, a deterministic finite automaton accepting the union of these four languages can be obtained in polynomial time, and the union of these languages equals L.

We show that the size of a maximum extended fooling set for L depends on whether $\bigcup_{i=1}^n L(A_i)$ equals Σ^*. Let $k = 4 + \sum_{i=1}^n |Q_i|$. Then size of a maximum extended fooling set for L equals k, if the union of deterministic finite automata languages under consideration is universal, and equals $k+1$ otherwise. To this end we argue as follows: Define the set of pairs $S = S' \cup S''$ with

$$S' = \{ (a_i w_{ij}, b_{ij}) \mid 1 \leq i \leq n \text{ and } 1 \leq j \leq t_i \}$$

and

$$S'' = \{(\lambda, a_1 b_{11}), (a_1 b_{11}, \lambda), (c, df), (cd, f)\},$$

where w_{ij} is any word in $P(i,j)$, for each $1 \leq i \leq n$ and $1 \leq j \leq t_i$. We claim that S is an extended fooling set for L.

It is readily observed that $xy \in L$ for all $(x,y) \in S$. Next, we note that the word $a_i w_{ij} b_{i\ell}$ is in L only if $j = \ell$. Of course, if $j = \ell$ then $a_i w_{ij} b_{i\ell} \in L$. Assume now $i \neq \ell$. Since the word begins with a_i and ends with $b_{i\ell}$, it is not in L, or it is in $P(i) \cup a_i \cdot L(A_i)$. It is clear that $w_{ij} \in P(i,j)$. Any word in $P(i)$ ending with $b_{i\ell}$ is in $a_i \cdot P(i,\ell) b_{i\ell}$, so $w_{ij} \in P(i,j) \cap P(i,\ell)$. But automaton A_i is deterministic, so $P(i,j) \cap P(i,\ell) = \emptyset$ if $j \neq \ell$, and thus $a_i w_{ij} b_{i\ell} \notin L$. Thus, all elements in S' obey the extended fooling set property. We turn to the elements in S'': Obviously, $a_1 b_{11} \in L$. But neither any of the words $a_i w_{ij} b_{ij} a_1 b_{1,1}$ nor any of $a_1 b_{1,1} a_i w_{ij} b_{ij}$ are in L. Therefore S' can be augmented by adding the elements $(\lambda, a_1 b_{11})$ and $(a_1 b_{11}, \lambda)$. Similar, none of the words cb_{ij}, $ca_{11}b_{11}$, c, and $cddf$ are in L, so the element (c, df) can be added to the set without altering this property. And finally, none of the words cdb_{ij}, $cda_{1,1}b_{1,1}$, and cd is in L. Therefore, S is in fact an extended fooling set as claimed above.

The rest of the proof consists in showing that there is an extended fooling set of cardinality at least $k+1$ if and only if $\bigcup_{i=1}^n L(A_i) \neq \Sigma^*$. In the case the language in question is universal, an explicit construction shows that the nondeterministic state complexity is at most k—for details of this construction we refer to [15, Claim 3.2]. Hence, there cannot be an extended fooling set of size $k+1$. Conversely, let $w \in \Sigma^*$ be a word not in $\bigcup_{i=1}^n L(A_i)$. Since $|w| \geq 2$, we can write $w = xy$ with $|x| \geq 1$ and $|y| \geq 1$. We claim that $S \cup \{(x,y)\}$ (the union is disjoint) is also a larger extended fooling set for L: Assume this is not the case. Then there is $(x', y') \in S$ such that xy' and $x'y$ are both in L. We first rule out the case that (x', y') is in S''. Then $xa_1 b_{1,1} \notin L$, if $|x| \geq 1$, and $a_1 b_{1,1} y \notin L$, if $|y| \geq 1$. Any word in L beginning with c ends either with f, or b_{ij}, for some i, j. Hence, neither cy nor cdy is in L. So (x', y') must be in S' and of the form $(a_i w_{ij}, b_{ij})$. Then both $a_i w_{ij} y$ and xb_{ij} are in L. We can deduce that $w_{ij} y \in L(A_i)$, since the word $a_i w_{ij} y$ begins with a_i. And $x \in P(i,j)$, since the word xb_{ij} ends with b_{ij}. Since A_i is deterministic and w_{ij} is also in $P(i,j)$, we have $w_{ij} \equiv_{L(A_i)} x$, where $\equiv_{L(A_i)}$ is the Myhill-Nerode equivalence relation for $L(A_i)$. But $w_{ij} y \in L(A_i)$ implies, by definition of the equivalence relation, that $xy \in L(A_i)$, contradicting $xy = w \in \Sigma^* \setminus \bigcup_{i=1}^n L(A_i)$. We conclude there is an extended fooling set of size $k+1$ in this case.

One can also obtain a PSPACE-hardness result for the biclique edge cover problem. In the case where L, the union of the deterministic finite automata languages, is not universal, there is an extended fooling set of size $k+1$, and since the bipartite dimension cannot be lower we have $d(G_L) \geq k+1$. In the other case, the nondeterministic state complexity equals k [15, Claim 3.2], and matches the size of a the extended fooling set S for L. But the bipartite dimension of the graph G_L is sandwiched between both measures. Thus, $d(G_L) \leq k$. □