Descriptional Complexity of Bounded Regular Languages

Andrea Herrmann, Martin Kutrib, Andreas Malcher, Matthias Wendlandt

Descriptional Complexity of Bounded Regular Languages

Andrea Herrmann, Martin Kutrib, Andreas Malcher, and Matthias Wendlandt

Institut für Informatik, Universität Giessen
Arndtstr. 2, 35392 Giessen, Germany
{kutrib,malcher,matthias.wendlandt}@informatik.uni-giessen.de

Abstract. We investigate the descriptional complexity of the subregular language classes of (strongly) bounded regular languages. In the first part, we study the costs for the determinization of nondeterministic finite automata accepting strongly bounded regular languages. The upper bound for the costs is larger than the costs for determinizing unary regular languages, but lower than the costs for determinizing arbitrary regular languages. In the second part, we study for (strongly) bounded languages the deterministic operational state complexity of the Boolean operations as well as the operations reversal, concatenation, and iteration. In detail, we present upper and lower bounds and we develop for the proof of the lower bounds a tool that exploits the number of different colorings of cycles occurring in deterministic finite automata accepting bounded languages.

1 Introduction

Descriptional complexity is an area of theoretical computer science in which one of the main questions is how succinctly a formal language can be described by a formalism in comparison with other formalisms. A fundamental result is the exponential trade-off between nondeterministic (NFA) and deterministic finite automata (DFA) [16]. A further exponential trade-off is known to exist between unambiguous and deterministic finite automata, whereas the trade-offs between alternating and deterministic finite automata [14] as well as between deterministic pushdown automata and deterministic finite automata [19] are bounded by doubly-exponential functions.

The question of whether the costs for determinization remain exponential even for subclasses of the regular languages, called *subregular* language classes, has been studied in [3, 4] for unary languages and in [18] for finite languages. A systematic study of the problem for subregular language classes is provided in [2]. In this paper, we study with *bounded regular languages* another subregular language class which has not gained much attention yet apart from the fundamental paper [7] in which bounded regular languages are introduced and, for example, characterization theorems are established. In general, a language is called (strongly) bounded if it is a subset of $a_1^* a_2^* \cdots a_k^*$, where $a_1, a_2, \ldots, a_k$ are (pairwise distinct) symbols. Bounded languages have been investigated to a large

extent in the literature. We would like to mention that basic results are summarized in [8] and that there exist strong connections to counter machines which are shown, for example, in [11, 13]. The descriptional complexity of bounded context-free languages has first been studied in [15] and recently in [12].

In this paper, we start to investigate the descriptional complexity of bounded regular languages. We provide the necessary definitions and notions in Section 2. Additionally, we summarize the closure properties for (strongly) bounded regular languages. In Section 3 we compute the costs for determinizing NFAs accepting strongly bounded regular languages. As bounded languages are both an extension of unary languages and a restriction of arbitrary languages, we obtain a 'similar' result for the upper bound of the determinization costs that turns out to be larger than the costs for determinizing unary NFAs, but lower than the costs for determinizing arbitrary NFAs. Finally, we study in Section 4 the deterministic operation problem for bounded regular languages which quantifies the costs (in terms of states of a DFA) of operations on (strongly) bounded regular languages such as union, intersection, concatenation, iteration, and reversal. The deterministic operation problem for regular languages has initially been studied in [20, 21]. Nowadays, there exists a vast literature on the deterministic and nondeterministic operational state complexity of subregular languages, and we refer to the recent survey [6]. Here, we complement these findings with the results for (strongly) bounded regular languages. It should be noted that we devise a new tool to obtain lower bounds for bounded regular languages which may be of interest on its own.

2 Preliminaries and Closure Properties

Let Σ^* denote the set of all words over the finite alphabet Σ. The *empty word* is denoted by λ, and $\Sigma^+ = \Sigma^* \setminus \{\lambda\}$. The *reversal* of a word w is denoted by w^R. For the *length* of w we write $|w|$. For the number of occurrences of a symbol a in w we use the notation $|w|_a$. We denote the powerset of a set S by 2^S. By $\gcd(x_1, x_2, \ldots, x_k)$ we denote the *greatest common divisor* of the integers $x_1, x_2, \ldots, x_k$, and by $lcm(x_1, x_2, \ldots, x_n)$ their *least common multiple*. If two numbers x and y are relatively prime, that is $\gcd(x, y) = 1$, we write $x \perp y$.

A *nondeterministic finite automaton* (NFA) is a system $M = \langle S, \Sigma, \delta, s_0, F \rangle$, where S is the finite set of *internal states*, Σ is the finite set of *input symbols*, $s_0 \in S$ is the *initial state*, $F \subseteq S$ is the set of *accepting states*, and $\delta : S \times \Sigma \to 2^S$ is the partial *transition function*. The *language accepted* by M is $L(M) = \{ w \in \Sigma^* \mid \delta(s_0, w) \cap F \neq \emptyset \}$, where the transition function is recursively extended to $\delta \colon S \times \Sigma^* \to 2^S$.

A finite automaton is *deterministic* (DFA) if and only if $|\delta(s, a)| = 1$, for all $s \in S$ and $a \in \Sigma$. In this case we simply write $\delta(s, a) = p$ for $\delta(s, a) = \{p\}$ assuming that the transition function is a mapping $\delta : S \times \Sigma \to S$. So, any DFA is complete, that is, the transition function is total, whereas for NFAs it is possible that δ maps to the empty set.

A language $L \subseteq \Sigma^*$ is said to be *bounded* if and only if $L \subseteq a_1^* a_2^* \cdots a_k^*$, for $k \geq 1$ and $a_i \in \Sigma$, $1 \leq i \leq k$. It is *strongly bounded* if all letters $a_1, a_2, \ldots, a_k$ are pairwise different. It should be noted that in the literature bounded languages which are defined as above are often called letter-bounded languages. Moreover, if symbols $a_1, a_2, \ldots, a_k$ are replaced by fixed words $w_1, w_2, \ldots, w_k$, a language $L \subseteq w_1^* w_2^* \cdots w_k^*$ is called word-bounded. However, in this paper we confine ourselves to investigating only bounded and strongly bounded languages over symbols.

The closure properties of bounded and strongly bounded languages are summarized in Table 1. Although both language classes are not closed under all operations, it is well known that the regular languages are closed under all operations. This allows to study the deterministic state complexity of all operations for (strongly) bounded regular languages.

	—	$\cup$	$\cap$	R	$\cdot$	$*$
Bounded Regular	no	yes	yes	yes	yes	no
Strongly Bounded Regular	no	no	yes	yes	no	no

Table 1. Summary of closure properties of the language families discussed.

3 Determinization

It is well known that the costs for the simulation of a nondeterministic finite automaton with n states by a deterministic finite automaton can be limited by 2^n many states using the power set construction. On the other hand, several different NFAs are known that reach this bound exactly. In the unary case the upper bound as well as the lower bound collapses to $e^{\Theta(\sqrt{n \cdot \log n})}$. Considering the costs for determinization in the strongly bounded regular case, one may expect that the bounds for the conversion might be strictly in between the bounds for the general and the unary case. In the following, we present an upper bound which is slightly more costly than in the unary case.

Theorem 1. *Let A be an NFA with n states accepting a strongly bounded regular language $L(A)$ over the alphabet Σ and $m = n \cdot |\Sigma|^2 + |\Sigma|$. Then an equivalent DFA A' with at most $|\Sigma|^2 \cdot e^{|\Sigma| \cdot \Theta\left(\sqrt{m \cdot \log(m)}\right)}$ many states can be constructed.*

Proof. Given an NFA $A = \langle S, \Sigma, s_0, \delta, F \rangle$ with n states accepting a strongly bounded regular language $L(A)$ over the alphabet Σ, we will construct an equivalent DFA. We may assume that $\Sigma = \{a_1, a_2, \ldots, a_k\}$ and $L(A) \subseteq a_1^* a_2^* \cdots a_k^*$ with $k \geq 2$ and pairwise distinct $a_i \in \Sigma$ with $1 \leq i \leq k$.

The principal idea of the construction is to divide automaton A into 'unary' sections $S_{a_1}, S_{a_2}, \ldots, S_{a_k}$ according to the read input symbols, to determinize

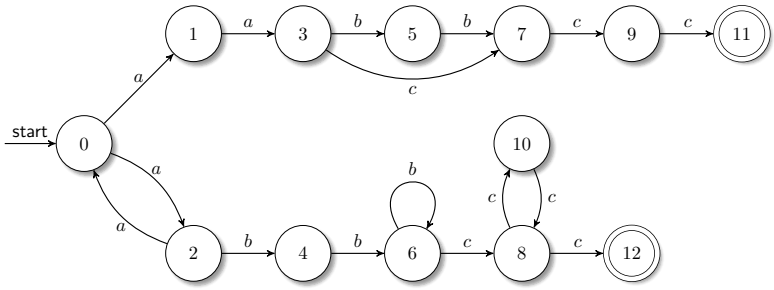

Fig. 1. An NFA A accepting a strongly bounded regular language.

these unary subautomata, and finally to reassemble the different deterministic subautomata to an equivalent DFA. In the first step, we construct an equivalent NFA A' with the property that each state of A' has incoming edges with at most one type of symbol. We define $A' = \langle S', \Sigma, s'_0, \delta', F' \rangle$, where s'_0 is a new state, $S' = \{ s_x \mid s \in S, x \in \Sigma \} \cup \{s'_0\}$ and $F' = \{ s_x \mid s \in F, x \in \Sigma \} \cup \{ s'_0 \mid s_0 \in F \}$. If $s' \in \delta(s, a)$ for some $s, s' \in S$ and $a \in \Sigma$, then define $s'_a \in \delta'(s_x, a)$ for all $x \in \Sigma$. If $s \in \delta(s_0, a)$ for some $s \in S$ and $a \in \Sigma$, then define $s_a \in \delta'(s'_0, a)$. The number of states of A' is at most $n \cdot |\Sigma| + 1$. Now, each state of A' has incoming edges with at most one type of symbol and a state s_a is defined to be in section S_a of A', for $a \in \Sigma$. Furthermore, the initial state s'_0 of A' is only visited in the first computation step and then never again. It is the single state in the special section S_{init}.

The next step is to modify A' in such a way that it has no states having more than one edge to another section labeled with the same symbol. Assume that there is some state $s \in S'$ having $\ell \geq 2$ edges labeled with a leading to section S_a. Then we add a new state s' to section S_a, add for every edge from s labeled with an a to some state s'' in S_a an edge from s' labeled with λ to the state s'', and replace the ℓ old edges by one edge from s to s' labeled with a. These modifications introduce at most $|\Sigma|$ new states for every state as well as λ-moves to the NFA, but preserve the given language. Moreover, for every input symbol $a \in \Sigma$, all nondeterministic moves on a take place inside section S_a. The number of states of A' is now at most $n \cdot |\Sigma|^2 + |\Sigma|$.

The first two steps of the construction based on the example NFA shown in Figure 1 are depicted in Figure 2.

In the following, the sections are successively determinized. We start with the determinization of the first section S_{a_1} having n_1 many states and define the set I_{a_1} of incoming states as the set of all states with incoming edges from other sections. Here, $I_{a_1} = \delta'(s'_0, a_1)$ consists of one state only, since there are no states having more than one edge to another section labeled with the same symbol. Additionally, we define the set O_{a_1} of states with outgoing edges to other sections as $O_{a_1} = \{ s \in S_{a_1} \mid r \in \delta'(s, a_j)$ for some $r \in S'$ and $k \geq j > 1 \}$. For the state $s \in I_{a_1}$ we construct an NFA A_s as subautomaton of A' with

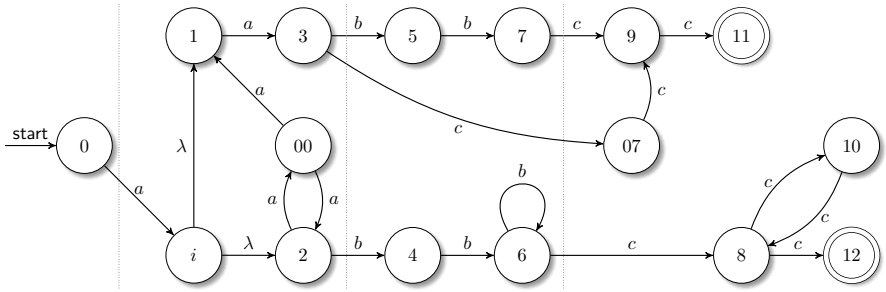

Fig. 2. The first two steps of the construction of A'. States 00 and 07 are added so that the initial state has no incoming edges and every state has only incoming edges with the same label. State i is added to ensure that there are no states having more than one edge to another section labeled with the same symbol.

state set S_{a_1} and s as initial state. Furthermore, all edges labeled with a_i such that $i \neq 1$ are removed. Finally, we eliminate λ-moves applying the construction given in [10] which does not increase the number of states. Additionally, we set all outgoing states from O_{a_1} as accepting. This is done to avoid that states with outgoing edges possibly disappear in the determinization process. Such states will later be rechanged to non-accepting states and completed with the outgoing edges to other sections. Thus, the NFA A_s accepts the unary language $\{ w \in a_1^* \mid \delta'(s, w) \cap (F' \cup O_{a_1}) \neq \emptyset \}$ and has at most n_1 states. Next, we apply the construction given in [3] to obtain a DFA A'_s such that $L(A'_s) = L(A_s)$. According to [3] the costs for determinizing A_s are bounded by $e^{\Theta\left(\sqrt{n_1 \cdot \log(n_1)}\right)}$. Since n_1 is bounded by $m = n \cdot |\Sigma|^2 + |\Sigma|$, we get an upper bound of $\ell_1 = e^{\Theta\left(\sqrt{m \cdot \log(m)}\right)}$ many states. Next, we construct an NFA A'' based on A' by replacing automaton A_s in section S_{a_1} of A' by its deterministic version. Additionally, let s' be the initial state of the DFA A'_s, then all transitions in A'' that link to state s are redirected to the initial state s' of A'_s. As a result of the construction, the DFA A'_s implemented in A'' has two different types of accepting states. The first type are the original accepting states where some input is accepted in A'. The second type are the states where the outgoing edges have to be placed. These have to be changed into non-accepting states and the outgoing edges have to be added to A''.

To find out which states in section S_{a_1} have to be accepting and which states have to be connected with other sections, we do the following considerations. Let B be an NFA accepting a unary language having a single accepting state. Then the lengths of the words of the accepted language $L(B)$ can be described by a finite set $\mathcal{E}$ of equations of the form $g(x) = z_g \cdot x + y_g$, where $x, y_g, z_g \geq 0$ are integers (see, for example, [3]). Looking at the equivalent DFA B', we obtain that for each equation $g \in \mathcal{E}$ there are one or more accepting states in B' indicating the divisibility of the input words according to g.

Now, we consider again for state $s \in I_{a_1}$ the NFA A_s and its equivalent DFA A'_s. First, we set all states in A'_s non-accepting. Second, for every accepting state f in A_s such that $f \in F'$ (thus, being an original accepting state in A'), we consider an NFA $A_{s,f}$ based on A_s where f is the only accepting state and we determine the set of equations $\mathcal{E}$ for $A_{s,f}$. Based on the obtained divisibilities we set the corresponding states in A'_s as accepting. Third, for every outgoing state $o \in O_{a_1}$ linking with input symbol a_j to some state $s_j \in S_j$ for some $k \geq j > 1$, we consider an NFA $A_{s,o}$ based on A_s where o is the only accepting state and we determine the set of equations $\mathcal{E}$ for $A_{s,o}$. Based on the obtained divisibilities we add to the corresponding states in A'_s outgoing edges labeled with a_j to state s_j. We notice that this adding of edges may introduce nondeterminism in A'_s. To remove such possible nondeterministic moves, we do the following: for every state $q \in A'_s$ having more than one outgoing edges labeled by some a_j with $k \geq j > 1$, we introduce a new state p to section S_{a_j}, replace all outgoing a_j-edges from q by outgoing λ-edges from p, and add one a_j-edge from q to p. Note that the removing of nondeterministic moves adds at most ℓ_1 states to each section S_{a_j}. Finally, we rename the NFA A'' with a determinized section S_{a_1} to A' and start the determinization of the next section S_{a_2}.

Again, we define the set I_{a_2} of states with incoming edges from other sections and the set O_{a_2} of states with outgoing edges to other sections. Formally,

$$I_{a_2} = \{\, s \in S_{a_2} \mid s \in \delta'(r, a_2) \text{ for some } r \in S' \text{ and } r \notin \delta'(q, a_2) \text{ for all } q \in S' \,\},$$
$$O_{a_2} = \{\, s \in S_{a_2} \mid r \in \delta'(s, a_j) \text{ for some } r \in S' \text{ and } k \geq j > 2 \,\}.$$

For each state s in I_{a_2} we construct an automaton A_s in a similar way as above. Thus, A_s accepts the unary language $\{\, w \in a_2^* \mid \delta'(s, w) \cap (F' \cup O_{a_2}) \neq \emptyset \,\}$ and has at most $n_2 + 1$ states, if s has been added by removing nondeterministic moves in the previous step, and at most n_2 states otherwise. Next, we determinize A_s and obtain an equivalent DFA A'_s with at most

$$e^{\Theta\left(\sqrt{(m+1)\cdot\log(m+1)}\right)} = e^{\Theta\left(\sqrt{m\cdot\log(m)}\right)} = \ell_1$$

many states. Then we construct an NFA A'' based on A' by replacing automaton A_s in section S_{a_2} of A' by its deterministic version and all transitions in A' that link to state s are redirected in A'' to the initial state of A'_s. Finally, we determine the accepting states of A'_s as well as the connections from outgoing states to other sections, and we remove possibly introduced nondeterminism. Having done this for all $s \in I_{a_2}$ we rename the NFA A'' with determinized sections S_{a_1} and S_{a_2} again to A'. The size of the determinized section S_{a_2} can be calculated as follows: we have at most $m + \ell_1 = e^{\Theta\left(\sqrt{m\cdot\log(m)}\right)} = \ell_1$ states in I_{a_2}. Each determinization costs at most ℓ_1 states. Thus, we obtain $\ell_2 = \ell_1^2$ as an upper bound for the determinization costs of section S_{a_2}. Again, note that the removing of nondeterministic moves adds at most ℓ_2 states to each section S_{a_j} with $k \geq j > 2$.

We continue the construction by determinizing successively the following sections in a similar way as described above. The costs for determinizing section S_{a_i} with $3 \leq i \leq k$ can be calculated as follows. There are at most

$m + \ell_1 + \ell_2 + \cdots + \ell_{i-1}$ states in I_{a_i} and each determinization costs at most ℓ_1 states. By setting $\ell_i = \ell_1^i$, we obtain $i \cdot \ell_i$ as total upper bound.

After determinizing all sections $S_{a_1}, S_{a_2}, \ldots, S_{a_k}$ we obtain a DFA A' being equivalent to A and the number of states of A' is bounded by the function $1 + \ell_1 + 2\ell_2 + \cdots + k \cdot \ell_k \leq k^2 \ell_1^k = |\Sigma|^2 \cdot e^{|\Sigma| \cdot \Theta\left(\sqrt{m \cdot \log(m)}\right)}$. □

4 Deterministic Operational State Complexity

This section is devoted to studying the deterministic operational state complexity of the family of strongly bounded regular languages, that is, the languages are given by DFAs. Clearly, the known upper bounds for general regular languages apply also here. Moreover, every unary language is also (strongly) bounded. So, the known lower bounds for unary languages apply here as well. In [21] it has been shown that the tight bounds for Boolean operations coincide for general regular and unary regular languages. In the unary case the lower bound requires the numbers of states to be relatively prime. In [17] unary regular languages are studied whose deterministic state complexities are not relatively prime. Here we can derive the following corollary for strongly bounded regular languages.

Corollary 2. *For any integers $m, n \geq 1$ let A be an m-state and B be an n-state DFA that accept strongly bounded languages.*

1. *Then m states are sufficient and necessary in the worst case for a DFA to accept the language $\overline{L(A)}$.*
2. *Then $m \cdot n$ states are sufficient for a DFA to accept the language $L(A) \cap L(B)$ (respectively $L(A) \cup L(B)$).*
3. *If $m \perp n$, then there exist a unary m-state DFA A and a unary n-state DFA B (with the same input symbol) such that any DFA accepting $L(A) \cap L(B)$ (respectively $L(A) \cup L(B)$) needs at least $m \cdot n$ states.*

Notice that the languages $L(A) \cup L(B)$ and $\overline{L(A)}$ are not necessarily strongly bounded. However, since they are regular they are accepted by DFAs in any case.

In the following, we turn to the operations reversal, iteration, and concatenation for which the deterministic state complexities of general and unary languages are different (see, for example, the summary in Table 2). So, an immediate question is to what extent the state complexity of strongly bounded languages is strictly in between both cases. Since the deterministic state complexities for unary languages are well known, we suppose that the strongly bounded languages that are investigated in the remainder of this section are defined over an alphabet of size at least two. In other words, we consider strongly bounded languages $L \subseteq a_1^* a_2^* \cdots a_k^*$ such that $k \geq 2$.

4.1 A Tool for Constructing Lower Bound Witnesses

A widely used method to show lower bounds is to define an infinite family of witness languages so that the sizes of the minimal automata accepting them

establish the bound. In order to allow the construction of witnesses as well as to determine the necessary sizes of the automata, lower bound techniques are very helpful. For example, in proofs dealing with the nondeterministic state complexity on regular languages specified by NFAs, the so-called fooling set technique can be used [1, 9].

Here we first present a tool for the definition of lower bound witnesses, that is, for the construction of DFAs accepting (strongly) bounded languages. The idea is based on the number of possibilities to color a cycle of a DFA whose edges are labeled with the same input letter. All cycles in a DFA accepting a (strongly) bounded language have this unary form. We use the two colors *gray* (g) and *white* (w).

Let S_c be a (sub)set of states of a given DFA that build a cycle on some fixed input letter. The *set of all colorings* of S_c with colors from $\{g, w\}$ is $X = \{\, f \mid f : S_c \to \{g, w\} \,\}$. So, there are $|X| = 2^{|S_c|}$ different such colorings.

Example 3. The DFA A depicted in Figure 3 has a cycle on input letter b, where the states of the cycle are $S_c = \{1, 2, 3, 4\}$. The coloring shown at the top of the figure is $f_1 \in X$ with $f_1(1) = g$, $f_1(2) = g$, $f_1(3) = w$, and $f_1(4) = w$. ■

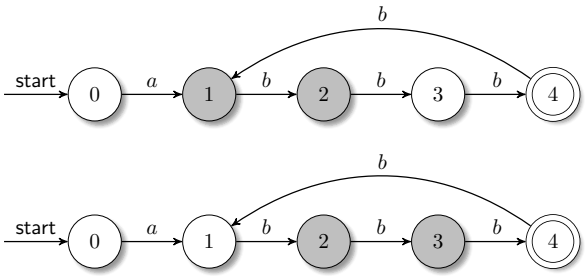

Fig. 3. Coloring of the cycle of DFA A from Example 3. Edges to the rejecting sink state are omitted.

For the clarity of presentation, colorings are written as words. For example, coloring f_1 can be written as $ggww$, and the set of all such colorings is $X = \{wwww, gwww, wgww, \ldots, gggg\}$. Moreover, a coloring can be uniquely identified by the set of states that are colored by w. For example, f_1 is given by $\{3, 4\}$.

Two colorings f_1 and f_2 are said to be equivalent if f_1 can be obtained from f_2 by applying the transition function. More precisely, two colorings f_1 and f_2 are equivalent if and only if there is some $\ell \geq 0$ so that $f_1(i) = f_2(\delta(i, x^\ell))$, for all $i \in S_c$. Here, S_c is the set of cycle states and the cycle is on input symbol x.

Example 4. Let $f_2 \in X$ with $f_2(1) = w$, $f_2(2) = g$, $f_2(3) = g$, and $f_2(4) = w$ be the coloring shown at the bottom of Figure 3. Then f_1 and f_2 are equivalent, since $f_1(i) = f_2(\delta(i, b))$, for all $i \in \{1, 2, 3, 4\}$. ■

Now, we turn to determine the number of possibilities to color a cycle with inequivalent colorings. The number depends only on the number of states in the cycle. For example, for four states we obtain the following equivalence classes $\{wwww\}$, $\{gwww, wgww, wwgw, wwwg\}$, $\{ggww, wggw, wwgg, gwwg\}$, $\{gggg\}$, $\{gggw, wggg, gwgg, ggwg\}$, and $\{gwgw, wgwg\}$, and thus six possibilities.

Now we consider the cyclic group G generated by the cyclic permutation $\langle(12\cdots|S_c|)\rangle$. The group naturally operates on S_c. Moreover, for $\sigma \in G$ and $f \in X$, let $\sigma f \in X$ be defined as $\sigma f(s) = f(\sigma^{-1}(s))$, for all $s \in S_c$. With this operation, G acts on the set X of colorings as well. So, two colorings are equivalent if and only if they are in the same orbit of G. Therefore, the number of possibilities to color a cycle with inequivalent colorings coincides with the number of orbits of G acting on X. This number can be determined by *Polya's enumeration lemma* that is a generalization of the well-known Burnside lemma on the number of orbits of a group action on a set (see, for example, [5, Chapter 8]). In the particular case of a cyclic group generated by a cyclic permutation $\langle(12\cdots n)\rangle$ and two colors, the number or orbits is $\frac{1}{n}\sum_{d|n}\varphi(d)\cdot 2^{\frac{n}{d}}$, where $d|n$ denotes the positive divisors of n and $\varphi(d) = |\{\, 1 \leq k \leq d \mid \gcd(k,d) = 1 \,\}|$ is Euler's function. For example, for $n = 4$ we have $d|n = \{1, 2, 4\}$. Since $\varphi(1) = 1$, $\varphi(2) = 1$, and $\varphi(4) = 2$, the number of orbits and, identically, the number of inequivalent colorings is $\frac{1}{4}(1\cdot 2^4 + 1\cdot 2^{\frac{4}{2}} + 2\cdot 2^{\frac{4}{4}}) = 6$.

4.2 Reversal, Concatenation, and Iteration

The first operation we consider in detail is the reversal. It turns out that the upper bound and lower bound can be described by an exponential function which is slightly smaller than in the case of arbitrary regular languages. On the other hand, in comparison with unary regular languages we obtain an exponential increase. The upper bound in the bounded case is derived from the observation that any DFA accepting some bounded language over an alphabet with at least two elements must have a rejecting sink state.

Theorem 5. *Let $k, n \geq 2$ be two integers and A be an n-state DFA that accepts a (strongly) bounded language $L(A) \subseteq a_1^* a_2^* \cdots a_k^*$. Then 2^{n-1} states are sufficient for a DFA to accept the language $L(A)^R$.*

Proof. Every non-unary DFA $A = \langle S, \Sigma, \delta, s_0, F\rangle$ accepting a (strongly) bounded language necessarily has a rejecting sink state, say $e \in S$. Now an NFA for the reversal of $L(A)$ is constructed by interchanging the initial state with the accepting states and reversing the direction of the transitions. The NFA is determinized which yields a DFA $A' = \langle 2^S, \Sigma, \delta', s_0', F'\rangle$ accepting $L(A)^R$. Since for all states $p, q \in 2^S$ so that $p = q \cup \{e\}$ we have $\delta'(p, v) \in F'$ if and only if $\delta'(q \cup \{e\}, v) \in F'$ if and only if $\delta'(q, v) \in F'$, for all $v \in \Sigma^*$, the states p and q are equivalent. We conclude that A' has at most 2^{n-1} states. □

In order to show the lower bound $2^{n-2}+1$ the coloring of cycles is exploited. Next, the construction of the witness DFAs is given, then we analyze a witness for six states. Finally, the general case is proven.

Let $n > 3$ be an integer. The DFA $A_n = \langle S_n, \Sigma, \delta_n, 0, \{n-2\}\rangle$ is constructed as follows (see Figure 4): $S_n = \{0, 1, \ldots, n-2, e\}$ where e denotes the rejecting sink state, $\Sigma = \{a, a_1, a_2, \ldots, a_k\}$ with $k = \frac{1}{n-2}\left(\sum_{d|n-2} \varphi(d) \cdot 2^{\frac{n-2}{d}}\right) - 1$, and

$$\delta_n(i,a) = \begin{cases} (i+1) \bmod n-2 & \text{for } 0 \le i \le n-3 \\ e & \text{otherwise} \end{cases}.$$

The transition function is still incomplete. Now we consider the colorings of the cycle, that is, of the states $\{0, 1, \ldots, n-3\}$, whereby we disregard $gg\cdots g$. From above it is known that there remain $k = \frac{1}{n-2}\left(\sum_{d|n-2} \varphi(d) \cdot 2^{\frac{n-2}{d}}\right) - 1$ inequivalent colorings. From each equivalence class M_j one element m_j, $1 \le j \le k$, is chosen and identified by the states that are colored white. For example, $wwgw$ is identified by $\{0, 1, 3\}$. Now, the definition of the transition function is completed by setting

$$\delta_n(i,a_j) = \begin{cases} n-2 & \text{if } i \in m_j \\ e & \text{otherwise} \end{cases}$$

for $1 \le j \le k$. The DFA A_n accepts the language

$$L(A_n) = \bigcup_{j=1}^{k} \bigcup_{i \in m_j} (a^{n-2})^* a^i a_j \subseteq a^* a_1^* a_2^* \cdots a_k^*.$$

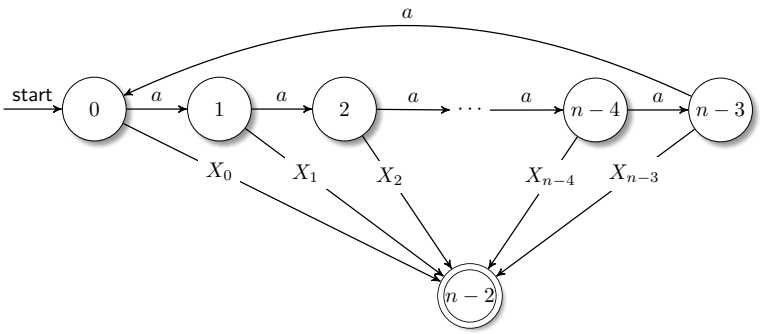

Fig. 4. The witness DFA A_n for reversal. The set of all a_j with $i \in m_j$ are denoted by X_i. Edges to the rejecting sink state are omitted.

Example 6. There are six inequivalent possibilities to color the 4-state cycle of A_6. Disregarding the coloring where all states are gray, the five equivalence

classes in question are

$$M_1 = \{\{0\},\{1\},\{2\},\{3\}\}, \quad M_2 = \{\{0,1\},\{1,2\},\{2,3\},\{3,0\}\},$$
$$M_3 = \{\{0,2\},\{1,3\}\}, \quad M_4 = \{\{0,1,2\},\{1,2,3\},\{2,3,0\},\{3,0,1\}\},$$
$$M_5 = \{\{0,1,2,3\}\}.$$

Choosing $m_1 = \{3\}$, $m_2 = \{2,3\}$, $m_3 = \{1,3\}$, $m_4 = \{1,2,3\}$, $m_5 = \{0,1,2,3\}$ yields $X_0 = \{a_5\}$, $X_1 = \{a_3, a_4, a_5\}$, $X_2 = \{a_2, a_4, a_5\}$, $X_3 = \{a_1, a_2, a_3, a_4, a_5\}$ in Figure 4. ■

Theorem 7. *For any integer $n > 3$, there exists an n-state DFA A that accepts a (strongly) bounded language such that any DFA accepting $L(A)^R$ needs at least $2^{n-2} + 1$ states.*

Proof. We use the DFA $A_n = \langle S_n, \Sigma, \delta_n, 0, \{n-2\}\rangle$ from above as witness. To show that A_n is minimal, consider two states $p, q \in \{0, 1, \ldots, n-3\}$. Let M_r denote the equivalence class with the colorings that color only one state white, and let $m_r = \{s\}$. Then $\delta_n(p, a^x a_r) = n-2$ and $\delta_n(q, a^x a_r) \neq n-2$, for $x = n-2-|p-s|$, since a_r sends only state s to the sole accepting state $n-2$. Clearly, the states $n-2$ and $p \in \{0, 1, \ldots, n-3\}$ are inequivalent.

The NFA $B_n = \langle P_n, \Sigma, \nu_n, p_{n-2}, \{p_0\}\rangle$ with $P_n = \{p_0, p_1, \ldots, p_{n-2}\}$ and

$$\nu_n(p_{n-2}, a_i) = \bigcup_{j \in m_i} p_j, \text{ for } 1 \leq i \leq k, \text{ and}$$
$$\nu_n(p_i, a) = p_j \text{ with } i = (j+1) \bmod (n-2), \text{ for } 0 \leq i \leq n-3,$$

accepts the language $L(A_n)^R$. Notice that $n-2 \notin m_j$ for all $1 \leq j \leq k$.

By applying the powerset construction, the NFA B_n is determinized which yields the DFA $A'_n = \langle S'_n, \Sigma, \delta'_n, p_{n-2}, F'_n\}\rangle$, where $S'_n = 2^{P_n \setminus \{p_{n-2}\}} \cup \{p_{n-2}\}$, $F'_n = \{\, T \in 2^{P_n \setminus \{p_{n-2}\}} \mid T \cap \{p_0\} \neq \emptyset \,\}$, $\delta'_n(\{p_{n-2}\}, a_i) = \bigcup_{j \in m_i} p_j$, for $1 \leq i \leq k$, $\delta'_n(\{p_{n-2}\}, a) = \emptyset$, and $\delta'_n(T, a) = \bigcup_{t \in T} \nu(t, a)$, for $T \in 2^{P_n \setminus \{p_{n-2}\}}$.

The DFA A'_n accepts $L(A)^R$ and has $2^{n-2} + 1$ states. By the construction of A_n and since the equivalence classes of colorings partition the set $2^{P_n \setminus \{p_{n-2}\}}$, all states of A'_n are reachable.

In order to show that A'_n is minimal, first consider the states $\{p_{n-2}\}$ and $R_i \in S'_n$, for $0 \leq i \leq n-3$. For $p_l \in R_i$ we have $\delta'_n(R_i, a^l) \in F'_n$ while $\delta'_n(\{p_{n-2}\}, a^l) \notin F'_n$.

Now let R_i and R_j be two different states from $S'_n \setminus \{p_{n-2}\}$ and let p_l be in their symmetric difference, say, $p_l \in R_i \setminus R_j$. Then $\delta'_n(R_i, a^l) \in F'_n$ while $\delta'_n(R_j, a^l) \notin F'_n$. Therefore, A'_n is minimal. □

Next, we turn to the operation iteration. Here, we will obtain tight upper and lower bounds that lie strictly in between the bounds for unary regular and arbitrary regular languages. Roughly speaking, the bounds for unary regular languages are quadratic and for arbitrary regular languages exponential. The bound for strongly bounded regular languages turns out to be the sum of a quadratic and an exponential function, where the quadratic function depends on the number of states of the first part of the given DFA and the exponential

function depends on the number of remaining states. The partitioning of the state set of a DFA accepting a strongly bounded language $L \subseteq a_1^* a_2^* \cdots a_k^*$ into two sets is done, roughly speaking, as follows: the first set is given by all states that are reachable with words from a_1^*. Since the DFA is deterministic, these states form a line or a line followed by a cycle in the state graph.

More precisely, let $A = \langle S, \Sigma, \delta, s_0, F \rangle$ be a minimal DFA accepting a strongly bounded language $L \subseteq a_1^* a_2^* \cdots a_k^*$ and let e denote the rejecting sink state of A if it exists. In the sequel, the set of states $q \in S$ with $q \neq e$ such that there exists a word v from $a_1^* a_2^* \cdots a_k^* \setminus a_1^*$ with $\delta(s_0, v) = q$ is denoted by S_2. The set $S \setminus (S_2 \cup \{e\})$ is denoted by S_1.

So, all states from S_1 are reachable only by words of the form a_1^*. For $k \geq 2$, we have $S = S_1 \cup S_2 \cup \{e\}$. The next theorem shows the upper bound for the iteration.

Theorem 8. *Let $n_1 \geq 2$ and $n_2 \geq 1$ be two integers and A be an $(n_1 + n_2 + 1)$-state DFA with state set S that accepts a strongly bounded language, so that $S = S_1 \cup S_2 \cup \{e\}$ with $|S_1| = n_1$ and $|S_2| = n_2$. Then $(n_1 - 1)^2 + 2^{n_2} + 2$ states are sufficient for a DFA to accept the language $L(A)^*$.*

In order to show a matching lower bound the coloring of cycles is exploited. Next, the construction of a $(2n+1)$-state witness DFA is given. Let $n \geq 1$ be an integer. The DFA $B_n = \langle S, \Sigma, \delta_n, 0, \{n-1, 2n-1\} \rangle$ is constructed as follows (see Figure 5): $S = \{0, 1, \ldots, 2n-1, e\}$ where e denotes the rejecting sink state, $\Sigma = \{a, b, a_1, a_2, \ldots, a_k\}$ with $k = \frac{1}{n}\left(\sum_{d|n} \varphi(d) \cdot 2^{\frac{n}{d}}\right) - n - 1$, and

$$\delta_n(i,a) = \begin{cases} (i+1) \bmod n & \text{for } 0 \leq i \leq n-1 \\ e & \text{otherwise} \end{cases},$$

$$\delta_n(i,b) = \begin{cases} i+n & \text{for } 0 \leq i \leq n-1 \\ (i+1) \bmod n & \text{for } n \leq i \leq 2n-1 \\ e & \text{otherwise} \end{cases}.$$

In order to complete the definition of the transition function we consider colorings of the cycle on input letter b, that is, of the states $\{n, n+1, \ldots, 2n-1\}$, whereby we disregard $gg\cdots g$, $wgg\cdots g$, $wwgg\cdots g$, $\cdots$, $ww\cdots wg$, and $ww\cdots w$. There remain $k = \frac{1}{n}\left(\sum_{d|n} \varphi(d) \cdot 2^{\frac{n}{d}}\right) - n - 1$ inequivalent colorings. From each equivalence class M_j one element m_j, $1 \leq j \leq k$, is chosen and identified by the states that are colored white. The states in m_j are denoted by $r_{0,j}, r_{1,j}, \ldots, r_{|m_j|-1,j}$. The definition of the transition function is completed by setting

$$\delta_n(i, a_j) = \begin{cases} r_{i,j} & \text{if } 0 \leq i \leq n-1 \text{ and } r_{i,j} \text{ is defined} \\ e & \text{otherwise} \end{cases}$$

for $1 \leq j \leq k$. The DFA B_n accepts the language

$$L(B_n) = (a^n)^* a^{n-1} \cup \bigcup_{j=0}^{n-1} (a^n)^* a^j b^{n-j} (b^n)^* \cup \bigcup_{j=1}^{k} \bigcup_{i=0}^{|m_j|-1} (a^n)^* a^i a_j b^{2n-1-r_{i,j}} (b^n)^*,$$

that is, $L(B_n) \subseteq a^* a_1^* a_2^* \cdots a_k^* b^*$.

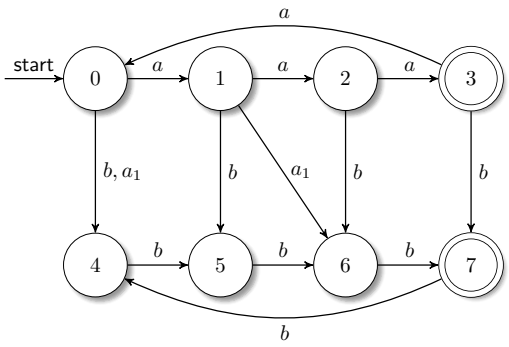

Fig. 5. The witness DFA B_4 for iteration. Edges to the rejecting sink state are omitted.

Example 9. From the six inequivalent possibilities to color the four-state cycle (see Example 6) of B_4 on input letter b only the sole equivalence class $M_1 = \{\{4, 6\}, \{5, 7\}\}$ remains. Choosing $m_1 = \{4, 6\}$ yields $r_{0,1} = 4$ and $r_{1,1} = 6$. So, $\delta_n(0, a_1) = 4$ and $\delta(1, a_1) = 6$ are defined (see Figure 5). ■

Theorem 10. *For any integers $n_1 = n_2 \geq 1$, there exists an $(n_1 + n_2 + 1)$-state DFA A that accepts a (strongly) bounded language such that any DFA accepting $L(A)^*$ needs at least $(n_1 - 1)^2 + 2^{n_2} + 2$ states.*

The final operation we consider is the concatenation. Again, the structure of (strongly) boundedness allows to reduce the descriptional complexity compared with the general case. As for iteration we obtain that the upper bound is described by the sum of a quadratic and an exponential function, where the number of states of the first DFA appear as quadratic resp. linear factor in both addends. As is done for iteration, the states of the second DFA are partitioned into two parts and the number of states of the first part appear as linear factor in the quadratic addend and as exponential factor in the other addend. The proof of the next theorem gives a detailed construction of the upper bound for the concatenation of two strongly bounded regular languages.

Theorem 11. *Let $m, n_1, n_2 \geq 1$ be integers, A be an m-state DFA, and A' be an $(n_1 + n_2 + 1)$-state DFA with state set S' that accept strongly bounded languages, so that $S' = S'_1 \cup S'_2 \cup \{e'\}$ with $|S'_1| = n_1$ and $|S'_2| = n_2$. Then in total $m^2 n_1 + (2m - 1)2^{n_2}$ states are sufficient for a DFA to accept the language $L(A)L(A')$.*

The currently best known lower bound for the concatenation of strongly bounded languages is derived from the concatenation of unary languages. In [21]

it is shown that for any $m, n \geq 1$ with $\gcd(m, n) = 1$ there exist an m-state DFA A and an n-state DFA A' accepting unary (and thus strongly bounded) languages so that any DFA that accepts the concatenation $L(A)L(A')$ has at least mn states. The results on the deterministic state complexity obtained in this section are summarized in Table 2.

	unary regular	bounded regular	regular
$L_1 \cup L_2$	$\leq mn$	$\leq mn$	mn
$L_1 \cap L_2$	$\geq mn$, if $\gcd(m, n) = 1$	$\geq mn$, if $\gcd(m, n) = 1$	
$\overline{L}$	m	m	m
$L_1 L_2$	$\leq mn$	$\leq m^2 n_1 + (2m-1)2^{n_2}$	$(2m-1)2^{n-1}$
	$\geq mn$, if $\gcd(m, n) = 1$	$\geq mn$, if $\gcd(m, n) = 1$	
L^*	$(m-1)^2 + 1$	$(m_1 - 1)^2 + 2^{m_2} + 2$	$2^{m-1} + 2^{m-2}$
L^R	m	$\leq 2^{m-1}$	2^m
		$\geq 2^{m-2} + 1$	

Table 2. Summary of the deterministic state complexity of the operations studied in this section. The upper and lower bounds for bounded regular languages are obtained in this section. The results for unary regular and arbitrary regular languages may be found, for example, in [6, 21].

5 Conclusions

In this paper, we have studied the descriptional complexity of (strongly) bounded regular languages. We have described a procedure for determinizing nondeterministic finite automata accepting strongly bounded regular languages. The obtained upper bound on the number of states is close to the known upper bound for the determinization of unary nondeterministic finite automata. Moreover, we have determined the deterministic state complexity of several operations on strongly bounded regular languages, in particular, of the operations reversal, iteration, and concatenation. The resulting upper and lower bounds are basically strictly in between the known bounds for unary and arbitrary regular languages. As interesting points for further research on the topic we would like to mention the improvement of the lower bound on concatenation, the study of additional operations, and the investigation of the nondeterministic state complexity of operations. Another interesting question is to look more closely at the size of the alphabets of the witness languages for the lower bounds. In the proofs given in this paper, the size is depending on the given number of states. It would clearly be of interest to study fixed alphabets or to consider the size of the alphabet as an additional parameter for upper and lower bounds.

References

1. Birget, J.C.: Intersection and union of regular languages and state complexity. Inform. Process. Lett. 43, 185–190 (1992)
2. Bordihn, H., Holzer, M., Kutrib, M.: Determination of finite automata accepting subregular languages. Theor. Comput. Sci. 410(35), 3209–3222 (2009)
3. Chrobak, M.: Finite automata and unary languages. Theoret. Comput. Sci. 47(2), 149–158 (1986)
4. Chrobak, M.: Errata to "Finite automata and unary languages". Theoret. Comput. Sci. 302, 497–498 (2003)
5. Erickson, M.J.: Introduction to Combinatorics. Wiley (1996)
6. Gao, Y., Moreira, N., Reis, R., Yu, S.: A survey on operational state complexity. CoRR abs/1509.03254 (2015), `http://arxiv.org/abs/1509.03254`
7. Ginsburg, S., Spanier, E.H.: Bounded regular sets. Proc. Amer. Math. Soc. 17(5), 1043–1049 (1966)
8. Ginsburg, S.: The Mathematical Theory of Context-Free Languages. McGraw Hill, New York (1966)
9. Glaister, I., Shallit, J.: A lower bound technique for the size of nondeterministic finite automata. Inform. Process. Lett. 59, 75–77 (1996)
10. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading, Massachusetts (1979)
11. Ibarra, O.H.: Reversal-bounded multicounter machines and their decision problems. J. ACM 25(1), 116–133 (1978)
12. Ibarra, O.H., Ravikumar, B.: On bounded languages and reversal-bounded automata. Inf. Comput. 246, 30–42 (2016)
13. Ibarra, O.H., Seki, S.: Characterizations of bounded semilinear languages by one-way and two-way deterministic machines. Int. J. Found. Comput. Sci. 23(6), 1291–1306 (2012)
14. Leiss, E.L.: Succinct representation of regular languages by Boolean automata. Theoret. Comput. Sci. 13, 323–330 (1981)
15. Malcher, A., Pighizzini, G.: Descriptional complexity of bounded context-free languages. Inf. Comput. 227, 1–20 (2013)
16. Meyer, A.R., Fischer, M.J.: Economy of description by automata, grammars, and formal systems. In: SWAT 1971. pp. 188–191. IEEE (1971)
17. Pighizzini, G., Shallit, J.: Unary language operations, state complexity and Jacobsthal's function. Int. J. Found. Comput. Sci. 13, 145–159 (2002)
18. Salomaa, K., Yu, S.: NFA to DFA transformation for finite languages over arbitrary alphabets. J. Autom. Lang. Comb. 2, 177–186 (1997)
19. Valiant, L.G.: Regularity and related problems for deterministic pushdown automata. J. ACM 22, 1–10 (1975)
20. Yu, S.: State complexity of regular languages. J. Autom., Lang. Comb. 6, 221–234 (2001)
21. Yu, S., Zhuang, Q., Salomaa, K.: The state complexities of some basic operations on regular languages. Theoret. Comput. Sci. 125(2), 315–328 (1994)