Square, Power, Positive Closure, and Complementation on Star-Free Languages

Sylvie Davies, Michal Hospodár

Square, Power, Positive Closure, and Complementation on Star-Free Languages

Sylvie Davies[1,⋆] and Michal Hospodár[2,⋆⋆(✉)]

[1] David R. Cheriton School of Computer Science
University of Waterloo, Waterloo, ON, Canada N2L 3G1
sylvie.davies@uwaterloo.ca
[2] Mathematical Institute, Slovak Academy of Sciences
Grešákova 6, 040 01 Košice, Slovakia
hosmich@gmail.com

Abstract. We examine the deterministic and nondeterministic state complexity of square, power, positive closure, and complementation on star-free languages. For the state complexity of square, we get a non-trivial upper bound $(n-1)2^n-2(n-2)$ and a lower bound of order $\Theta(2^n)$. For the state complexity of the k-th power in the unary case, we get the tight upper bound $k(n-1)+1$. Next, we show that the upper bound kn on the nondeterministic state complexity of the k-th power is met by a binary star-free language, while in the unary case, we have a lower bound $k(n-1)+1$. For the positive closure, we show that the deterministic upper bound $2^{n-1}+2^{n-2}-1$ as well as the nondeterministic upper bound n, can be met by star-free languages. We also show that in the unary case, the state complexity of positive closure is $n^2-7n+13$, and the nondeterministic state complexity of complementation is between $(n-1)^2+1$ and n^2-2.

1 Introduction

The state complexity of a regular language L, $\mathrm{sc}(L)$, is the smallest number of states in any deterministic finite automaton (DFA) recognizing the language L. The state complexity of a unary regular operation $\circ$ is the function from $\mathbb{N}$ to $\mathbb{N}$ given by $n \mapsto \max\{\mathrm{sc}(L^\circ) \mid \mathrm{sc}(L) \leq n\}$. The nondeterministic state complexity of a regular language or a unary regular operation is defined analogously using the representation of languages by nondeterministic finite automata (NFAs).

The (nondeterministic) state complexity of star, reversal, square, power, and complementation was examined in [4, 6, 10–12, 14, 18]. Researchers also investigated the complexity of operations on some subregular classes. For example, Câmpeanu et al. [3] considered finite languages, and Pighizzini and Shallit [13] studied operational state complexity on unary regular languages.

⋆ Research supported by the Natural Sciences and Engineering Research Council of Canada under grant No. OGP0000871.
⋆⋆ Research supported by VEGA grant 2/0132/19 and grant APVV-15-0091.

The class of *star-free languages* is the smallest class containing finite languages and closed under complementation, union, and concatenation. In 1965, Marcel-Paul Schützenberger [15] proved that a language is star-free if and only if its syntactic monoid is group-free, that is, it has only trivial subgroups. An equivalent condition is that the minimal DFA of a star-free language is permutation-free, that is, there is no string that induces a non-trivial permutation on any subset of the set of states.

The operational state complexity of basic regular operations on star-free languages represented by DFAs was examined by Brzozowski and Liu [1], while Holzer, Kutrib, and Meckel [7] considered basic operations on star-free languages represented by NFAs. Except for reversal on DFAs, all regular upper bounds have been shown to be met by star-free languages.

In this paper, we continue this research and study the state complexity and nondeterministic state complexity of square, power, positive closure, and complementation on star-free languages. As the main result of this paper, we get nontrivial upper and lower bounds for square on DFAs in Sect. 3. In Sect. 4, we obtain tight upper bounds for positive closure on DFAs and NFAs, and for power on unary DFAs and binary NFAs. Moreover, we present lower and upper bounds for complementation on unary NFAs that are better than in [7].

2 Preliminaries

Let Σ be a finite non-empty alphabet of symbols. Then Σ^* denotes the set of strings over Σ including the empty string ε. A language is any subset of Σ^*.

A *nondeterministic finite automaton* (NFA) is a 5-tuple $A = (Q, \Sigma, \cdot, s, F)$ where Q is a finite set of states, Σ is a finite alphabet, $\cdot\colon Q \times \Sigma \to 2^Q$ is the transition function which is naturally extended to the domain $2^Q \times \Sigma^*$, $s \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states. We say that (p, a, q) is a *transition* in NFA A if $q \in p \cdot a$. The *language accepted* by the NFA A is the set $L(A) = \{w \in \Sigma^* \mid s \cdot w \cap F \neq \emptyset\}$.

An NFA A is a *deterministic finite automaton* (DFA) if $|q \cdot a| = 1$ for each q in Q and each a in Σ. In a DFA, each symbol a in Σ induces a *transformation* on the set Q, given by $q \mapsto q \cdot a$. We frequently describe the transitions of DFAs by just describing their induced transformations.

The state complexity of L, $\mathrm{sc}(L)$, is the smallest number of states in any DFA for L. The nondeterministic state complexity of L, $\mathrm{nsc}(L)$, is defined analogously. It is well known that if a language L is finite with the longest string of length ℓ, then $\mathrm{sc}(L) \leq \ell + 2$ and $\mathrm{nsc}(L) \leq \ell + 1$.

The reader may refer to [2, 8, 16, 17] for details and all unexplained notions.

3 Square

In this section, we investigate the square operation on the class of star-free languages represented by DFAs. Let $A = (Q, \Sigma, \cdot, s, F)$ be an arbitrary DFA. As is well known, we may construct a DFA B recognizing $L(A)^2$ as follows:

- The state set of B is $Q \times 2^Q$, where 2^Q is the set of all subsets of Q.
- The initial state of B is $(s, \emptyset)$ if $s \notin F$, or $(s, \{s\})$ if $s \in F$.
- The final state set is $F_B = \{(q, S) \mid S \cap F \neq \emptyset\}$.
- The transition function $\odot$ is defined as follows for each $a \in \Sigma$:

$$(q, S) \odot a = \begin{cases} (q \cdot a, S \cdot a), & \text{if } q \cdot a \notin F; \\ (q \cdot a, S \cdot a \cup \{s\}), & \text{if } q \cdot a \in F. \end{cases}$$

To simplify the notation, we write $(q, S)a$ instead of $(q, S) \odot a$, and we write qa and Sa instead of $q \cdot a$ and $S \cdot a$ for each q in Q and each subset S of Q.

We will use the following lemma.

Lemma 1. *Let $q \in F$ and $T \subseteq Q$. Suppose (q, T) is reachable via w in B. For each $p \in T$, there is a suffix x of w such that $sx = p$ and $fx = q$ for some $f \in F$.*

Proof. Write $w = w_1 \cdots w_k y$ where $w_1 \cdots w_i$ is the i-th shortest prefix of w contained in $L(A)$. Let $z_i = w_{i+1} \cdots w_k$ and $z_k = \varepsilon$. Then $(s, \emptyset)w = (q, T) = (q, \{sz_1y, sz_2y, sz_3y, \ldots, sz_ky\})$. Let x be z_iy with $sz_iy = p$ and let $f = sw_1 \cdots w_i$. □

Upper Bound. For each q in Q, define the following set of states:

$$\mathrm{SCC}(q) = \{p \in Q \mid \exists x, y \in \Sigma^* \text{ such that } qx = p \text{ and } py = q\}.$$

The set $\mathrm{SCC}(q)$ is called the *strongly connected component* (SCC) of A containing q; it is the maximal set of states containing q such that for each state p in the set, there are directed paths leading from q to p and from p to q. Of particular importance is the component $\mathrm{SCC}(s)$ containing the initial state s.

Proposition 2. *Let A be a* DFA *of a star-free language in which all states are reachable. Consider the number of reachable states in the* DFA *B for $L(A)^2$.*

1. *If s is the unique final state of A, then $L(A) = L(A)^2$ and at most n states are reachable.*
2. *Otherwise, if $\mathrm{SCC}(s) = \{s\}$, then at most $(n-2)2^n + 2^{n-1} + 1$ states are reachable.*
3. *Otherwise, let $i = |\mathrm{SCC}(s)|$ and $j = |\mathrm{SCC}(s) \setminus F|$. Then the number of reachable states is at most $(n-1)2^n - 2(n-2-j) - j2^{n-i+1}$.*

The maximum value of these bounds is $(n-1)2^n - 2(n-2)$, corresponding to case (3) with either $j = 0$ or $i = n$.

Proof. Statement (1) holds true since only states of the form $(q, \{q\})$ are reachable. For (2), suppose $\mathrm{SCC}(s) = \{s\}$. The initial state of B in this case is either $(s, \emptyset)$ or $(s, \{s\})$. Let a be a letter and let $sa = q$. If $q = s$ then $(s, \emptyset)a = (s, \emptyset)$ and $(s, \{s\})a = (s, \{s\})$, so no new states are reached. If $q \neq s$, we reach $(q, \emptyset)$ or $(q, \{q\})$ if q is non-final, and we reach $(q, \{s\})$ or $(q, \{q, s\})$ if q is final. If $q \neq s$, then $q \notin \mathrm{SCC}(s)$, so we can never return to a state with s in the first component. Thus all non-initial states with s in the first component are unreachable.

Now consider states (q, S) with $q \neq s$. If q is non-final, there are at most 2^n reachable states of the form (q, S). If q is final, there are at most 2^{n-1} reachable states of the form (q, S), since S must contain s. Let $|F| = k$, and note:

- If $s \in F$, there are at most $(n-k)2^n + (k-1)2^{n-1} + 1$ reachable states.
- If $s \notin F$, there are at most $(n-k-1)2^n + k2^{n-1} + 1$ reachable states.

Both values are maximized by taking k as small as possible:

- If $s \in F$, we cannot take $k = 1$ or else we are in case (1) (since then s is the unique final state of automaton A), so taking $k = 2$ we get the upper bound $(n-2)2^n + 2^{n-1} + 1$.
- If $s \notin F$, taking $k = 1$ gives $(n-2)2^n + 2^{n-1} + 1$ again.

Now, we consider (3). Let q be a *non-final* state in $\mathrm{SCC}(s)$. Let T be a set that contains $\mathrm{SCC}(s) \setminus \{q\}$. We claim that (q, T) is unreachable.

Suppose for a contradiction that (q, T) is reachable. Then there is a state (p, U) and letter a such that $(p, U)a = (q, T)$. Since q is non-final, we have $pa = q$ and $Ua = T$. Notice that Ua contains $\mathrm{SCC}(s)\setminus\{q\}$, since T contains $\mathrm{SCC}(s)\setminus\{q\}$. Construct a subset V of U as follows: for each element $u \in \mathrm{SCC}(s) \setminus \{q\}$, choose one element $v \in U$ such that $va = u$. Then $|V| = |\mathrm{SCC}(s) \setminus \{q\}|$.

If $v \in V$, we have $va \in \mathrm{SCC}(s) \setminus \{q\}$, so there is a path from v to the initial state s. Since all states of A are reachable, there is also a path from s to v. Therefore v belongs to $\mathrm{SCC}(s)$. It follows that $V \subseteq \mathrm{SCC}(s)$. Since $|V| = |\mathrm{SCC}(s) \setminus \{q\}|$, we must have $V = \mathrm{SCC}(s) \setminus \{r\}$ for some state $r \in \mathrm{SCC}(s)$. Then $(\mathrm{SCC}(s) \setminus \{r\})a = (\mathrm{SCC}(s) \setminus \{q\})$.

We claim that $r = p$. To see this, first note that p must be in $\mathrm{SCC}(s)$. Indeed, there is a path from s to p since all states of A are reachable, and there is a path from p to s since $pa = q$ and $q \in \mathrm{SCC}(s)$. Now, if $r \neq p$, then $p \in V = \mathrm{SCC}(s) \setminus \{r\}$. But then $q \in Va$, which is impossible since $Va = \mathrm{SCC}(s) \setminus \{q\}$. It follows that $r = p$ and thus $V = \mathrm{SCC}(s) \setminus \{p\}$.

Thus we have $pa = q$ and $(\mathrm{SCC}(s) \setminus \{p\})a = \mathrm{SCC}(s) \setminus \{q\}$. But this means that a acts as a permutation on $\mathrm{SCC}(s)$. Since $L(A)$ is star-free, a must act as the identity on $\mathrm{SCC}(s)$. Therefore $p = q$ and $V = \mathrm{SCC}(s) \setminus \{q\}$.

This means that if T contains $\mathrm{SCC}(s) \setminus \{q\}$, then the state (q, T) only has immediate predecessors (that is, states from which (q, T) can be reached in one transition) of the form (q, U), where U contains a subset $V = \mathrm{SCC}(s) \setminus \{q\}$. We claim that either $\mathrm{SCC}(s) = \{s\}$, or all states (q, T) with $\mathrm{SCC}(s) \subseteq T$ are unreachable. Indeed, consider a path leading from the initial state to (q, T). If $q \neq s$, then (q, T) is unreachable since every state in the path must have q as the first component. If $q = s$, then the second component of the initial state which is either $(s, \emptyset)$ or $(s, \{s\})$ must contain $\mathrm{SCC}(s) \setminus \{s\}$, and this occurs only if $\mathrm{SCC}(s) = \{s\}$. If $\mathrm{SCC}(s) = \{s\}$, we are in case (2), which has been dealt with. Thus if $\mathrm{SCC}(s) \subseteq T$, then (q, T) is unreachable, as required.

We can also show that if $q \notin \mathrm{SCC}(s)$ is non-final, then the states (q, Q) and $(q, Q \setminus \{q\})$ are not reachable. Indeed, suppose we have $(p, S)a = (q, Q \setminus \{q\})$. Since q is non-final, we have $pa = q$ and $Sa = Q \setminus \{q\}$. Note that S cannot contain p, since otherwise Sa would contain q. Also, we must have $|S| \geq |Q \setminus \{q\}|$. It follows that $S = Q \setminus \{p\}$. But then $pa = q$ and $(Q \setminus \{p\})a = Q \setminus \{q\}$, so a is a permutation of Q and thus the identity. Then $p = q$ and $S = Q \setminus \{q\}$, meaning

$(p, S) = (q, Q \setminus \{q\})$. Thus our target state $(q, Q \setminus \{q\})$ can only be reached from itself, so it is unreachable. A similar argument shows that (q, Q) is unreachable.

Now we count the number of potentially reachable states in this case. Consider a state (q, S). If q is final, then S must contain s, and so for each final state q we get 2^{n-1} potentially reachable states. For each non-final state q, we have an upper bound of 2^n reachable states (q, S). If $|F| = k$, this gives a total upper bound of $(n-k)2^n + k2^{n-1}$. But we can get a better bound, since we know that if $q \in \mathrm{SCC}(s)$ is non-final and S contains $\mathrm{SCC}(s) \setminus \{q\}$, then (q, S) is not reachable. For each of the j non-final states $q \in \mathrm{SCC}(s)$, there are 2^{n-i+1} unreachable states (q, S) such that S contains $\mathrm{SCC}(s) \setminus \{q\}$. For each of the $n-k-j$ non-final states $q \notin \mathrm{SCC}(s)$, there are 2 unreachable states (q, Q) and $(q, Q \setminus \{q\})$. Subtracting these states gives an upper bound of $(n-k)2^n + k2^{n-1} - 2(n-k-j) - j2^{n-i+1}$.

If we plug in $k = 2$ here, we get the stated upper bound. If we plug in $k \geq 3$ we get something smaller than the stated upper bound. However, for $k = 1$ we get something larger than the stated bound, so we need to do some extra work.

Suppose $|F| = k = 1$, and set $F = \{q\}$. We can assume $q \neq s$ since otherwise we are in case (1). Since $s \notin F$, the initial state of the square DFA is $(s, \emptyset)$. We claim states (s, S) with $q \in S$ are not reachable.

To see this, suppose for a contradiction that a state (s, S) with $q \in S$ is reachable. To reach (s, S) we must first pass through some reachable state of the form (q, T). So there exists a string y such that $qy = s$ and $Ty = S$. Choose $p \in T$ such that $py = q$. Let w be a string leading from the initial state to (q, T). Then by Lemma 1, there exists a suffix x of w such that $sx = p$ and $qx = q$. Now $sxy = py = q$ and $qxy = qy = s$, so the string xy swaps s and q. This is a contradiction, since $L(A)$ is star-free.

So let us take our previous bound of $(n-k)2^n + k2^{n-1} - 2(n-k-j) - j2^{n-i+1}$, set $k = 1$, and subtract these new states we have proved to be unreachable. There are 2^{n-1} states (s, S) with $q \in S$. However, we have already counted the states (s, Q) and $(s, Q \setminus \{s\})$ as not reachable, so instead of subtracting 2^{n-1} we subtract $2^{n-1} - 2$. Thus we get $(n-1)2^n + 2^{n-1} - 2(n-1-j) - j2^{n-i+1} - 2^{n-1} + 2 = (n-1)2^n - 2(n-2-j) - j2^{n-i+1}$, as required. □

Lower Bound. For $n \geq 4$, we define a function $f(n)$ as follows. Let $k = \lfloor n/2 \rfloor$. For $0 \leq i \leq n-1$, define $d(i) = min\{|k-1-i|, |k+1-i|\}$. Define a function $g(n, i)$ as follows:

$$g(n,i) = \begin{cases} 2^n - 2, & \text{if } i = k; \\ 2^{n-1}, & \text{if } i \in \{k-1, k+1\}; \\ 2^n - 2^{k-d(i)} \sum_{j=1}^{d(i)} \binom{k+1+d(i)}{k+1+j}, & \text{if } n \text{ is odd, } i \notin \{k-1, k, k+1\}; \\ 2^n - 2^{k-d(i)} \sum_{j=1}^{d(i)} \binom{k+d(i)}{k+j}, & \text{if } n \text{ is even, } i < k-1; \\ 2^n - 2^{k-1-d(i)} \sum_{j=1}^{d(i)} \binom{k+1+d(i)}{k+1+j}, & \text{if } n \text{ is even, } i > k+1. \end{cases} \tag{1}$$

Now set $f(n) = \sum_{i=0}^{n-1} g(n, i)$. We claim that $f(n)$ is a lower bound on the state complexity of the square of a star-free language with state complexity n. Based

on the results of computational random searches, we conjecture that this lower bound is tight, but we were unable to prove a matching upper bound.

Our witness for this lower bound is the DFA A defined as follows. Fix $n \geq 4$ and let $Q = \{0, 1, \ldots, n-1\}$. A transformation t of Q is *monotonic* if $p \leq q$ implies $pt \leq qt$ for all $p, q \in Q$. Let Σ be the set of all monotonic transformations of Q. By [9, Theorem 2.1], we have $|\Sigma| = \binom{2n-1}{n-1}$. Let $k = \lfloor n/2 \rfloor$ as before. Finally, let $A = (Q, \Sigma, \cdot, k, \{k-1, k+1\})$ be the DFA where $q \cdot t = qt$ for all $t \in \Sigma$.

Theorem 3. *Let $n \geq 4$, define A as above, and let B be the* DFA *for the square $L(A)^2$. The number of reachable and pairwise distinguishable states of B is precisely $f(n)$.*

Proof. We claim that the function $g(n, i)$ counts the number of reachable states of B with the form (i, S), for $i \in Q$ and $S \subseteq Q$. Thus the total number of reachable states is given by $f(n)$, which takes the sum of the $g(n, i)$ terms for each $i \in Q$. We must prove that the counting function $g(n, i)$ is correct, and also that all of the reachable states we count are pairwise distinguishable.

The initial state of B is $(k, \emptyset)$. First we show that the states $(k, \{k-j\})$ and $(k, \{k+j\})$ are reachable for each $j \geq 2$ such that the state in question is in Q. From the initial state, apply the transformation $(k \to k-1)$ to reach $(k-1, \{k\})$. Now apply the transformation which sends $k-1$ to k, fixes everything below $k-1$, and sends everything above $k-1$ to $k+j$. We reach $(k, \{k+j\})$. Symmetrically, from the initial state, apply $(k \to k+1)$ to reach $(k+1, \{k\})$. Then apply the transformation that sends $k+1$ to k, fixes everything above $k+1$, and sends everything below $k+1$ to $k-j$. We reach $(k, \{k-j\})$.

Next, we show that $(k, \{k-2, k-3, \ldots, 0\})$ and $(k, \{k+2, k+3, \ldots, n-1\})$ are reachable. Assume we have reached $(k, \{k-2, k-3, \ldots, k-j\})$ for some $j \geq 2$. Apply $(k \to k+1)$ to reach $(k+1, \{k, k-2, k-3, \ldots, k-j\})$. Then apply the transformation that fixes 0 and every $q \geq k+2$, sends $k+1$ to k, sends k to $k-2$, and sends q to $q-1$ if $1 \leq q \leq k-1$. We reach $(k, \{k-2, k-3, \ldots, k-j-1\})$ and by applying these two transformations repeatedly we reach $(k, \{k-2, k-3, \ldots, 0\})$. A symmetric argument works for $(k, \{k+2, k+3, \ldots, n-1\})$.

Next, we show the following states are reachable: $(k-1, Q)$, $(k-1, Q \setminus \{k-1\})$, $(k+1, Q)$, $(k+1, Q \setminus \{k+1\})$. From $(k, \{k-2, k-3, \ldots, 0\})$, apply $(k \to k-1)$ to reach $(k-1, \{k, k-2, k-3, \ldots, 0\})$. Repeatedly apply $(k \to k+1 \to \cdots \to n-2 \to n-1)$ to reach $(k-1, Q \setminus \{k-1\})$. Then apply $(k \to k-1)$ to reach $(k-1, Q)$. Symmetrically, we can reach $(k+1, Q \setminus \{k+1\})$ and then $(k+1, Q)$.

Next, we describe how to reach states of the form (q, S) where S satisfies certain properties. For each q, we define:

$$S_{<q} = \{p \in S \mid p < q\}, \quad S_{>q} = \{p \in S \mid p > q\}, \quad S_q = S \cap \{q\}.$$

Then S is the disjoint union $S_{<q} \cup S_q \cup S_{>q}$. We claim that (q, S) is reachable if one of the following conditions holds:

1. n is odd, $|S_{<q}| \leq \min\{q, k+1\}$, and $|S_{>q}| \leq \min\{n-q-1, k-1\}$.
2. n is odd, $|S_{<q}| \leq \min\{q, k-1\}$, and $|S_{>q}| \leq \min\{n-q-1, k+1\}$.

3. n is even, $|S_{<q}| \leq \min\{q, k+1\}$, and $|S_{>q}| \leq \min\{n-q-1, k-2\}$.
4. n is even, $|S_{<q}| \leq \min\{q, k-1\}$, and $|S_{>q}| \leq \min\{n-q-1, k\}$.

Additionally, if q is a final state ($k-1$ or $k+1$) then S must contain the initial state k.

To see this, first observe that $S_{<q}$ always has size at most q and $S_{>q}$ always has size at most $n-q-1$; no sets exist which do not satisfy these bounds. Now suppose condition (1) holds; then n is odd and $n = 2k+1$. Observe that $Q_{<k+1}$ has size $k+1$ and $Q_{>k+1}$ has size $n-(k+1)-1 = 2k+1-k-1-1 = k-1$. Since $|S_{<q}| \leq k+1$ and $|S_{>q}| \leq k-1$, there is a monotonic transformation that sends $Q_{<k+1}$ to $S_{<q}$ and $Q_{>k+1}$ to $S_{>q}$; we can define the transformation as follows: For i in $Q_{<k+1}$, map i to the $(i+1)$-th smallest element of $S_{<q}$, or to the largest element of $S_{<q}$ if $i+1 > |S_{<q}|$. For $i \in Q_{>k+1}$, map i to the $(i-(k+1))$-th smallest element of $S_{>q}$, or to the largest element of $S_{>q}$ if $i-(k+1) > |S_{>q}|$. Finally, $k+1$ is mapped to q. If S contains q, reach $(k+1, Q)$ and apply this transformation to reach (q, S). Otherwise, reach $(k+1, Q \setminus \{k+1\})$ and apply this transformation to reach (q, S).

If condition (2), (3) or (4) holds, we use symmetric arguments. For condition (2), observe that $Q_{<k-1}$ has size $k-1$ and $Q_{>k-1}$ has size $k+1$. Since the condition implies $|S_{<q}| \leq k-1$ and $|S_{>q}| \leq k+1$, there is a monotonic transformation that sends $Q_{<k-1}$ to $S_{<q}$ and $Q_{>k-1}$ to $S_{>q}$, defined analogously to the transformation before. For condition (3), $Q_{<k+1}$ has size $k+1$, but since $n = 2k$ is even, the set $Q_{>k+1}$ has size $n-(k+1)-1 = 2k-k-1-1 = k-2$. For condition (4), $Q_{<k-1}$ has size $k-1$ and $Q_{>k-1}$ has size k.

This argument shows that all states (q, S) which satisfy the four conditions described above (and satisfy $k \in S$ if q is final) are reachable. Now we count the number of states satisfying these conditions.

Case 1: States of the form $(k-1, S)$. Note that $S_{<k-1}$ has size at most $k-1$. If n is odd, then $S_{>k-1}$ has size at most $k+1$. If n is even, then $S_{>k-1}$ has size at most k. So for all S, either condition (2) or condition (4) is satisfied. Also, since $k-1$ is final, S must contain the initial state k. Thus every state $(k-1, S)$ with $k \in S$ is reachable; there are 2^{n-1} such states.

Case 2: States of the form $(k+1, S)$. One may verify that for all S, either condition (1) or condition (3) is satisfied. Since $k+1$ is final, S must contain k. Thus every state $(k+1, S)$ with $k \in S$ is reachable; there are 2^{n-1} such states.

Case 3: States of the form (k, S). Note that $S_{<k}$ has size at most k. If n is odd then $S_{>k}$ has size at most k. If n is even then $S_{>k}$ has size at most $k-1$. If $|S_{<k}| \leq k-1$, then condition (2) or condition (4) is satisfied. If $|S_{<k}| = k$, then condition (1) or condition (3) is satisfied unless $S_{>k}$ is too large. The only way it can be too large is if n is odd and $|S_{>k}| = k$, or n is even and $|S_{>k}| = k-1$. But in both these cases, we have $|S_{<k}| + |S_{>k}| = n-1$ and thus $S_{<k} \cup S_{>k} = Q \setminus \{k\}$. So there are only two choices for S that do not meet a reachability condition: Q and $Q \setminus \{k\}$. Thus there are $2^n - 2$ reachable states of the form (k, S).

Case 4: States of the form (q, S), $q < k-1$. Write $q = k-1-d$ with $d \geq 1$. Then $S_{<q}$ has size at most $k-1-d$. If n is odd, then $S_{>q}$ has size at most $k+1+d$.

If n is even, then $S_{>q}$ has size at most $k+d$. We always have $|S_{<q}| \leq k+1$. If n is odd, then condition (2) is met as long as $|S_{>q}| \leq k+1$. If n is even, then condition (4) is met as long as $|S_{>q}| \leq k$. Let us count the number of sets S which fail these conditions.

Write S as the disjoint union $S_{<q} \cup S_q \cup S_{>q}$. There are 2^{k-1-d} choices for $S_{<q}$ and 2 choices for S_q. If n is odd, to fail condition (2), we need $|S_{>q}| = k+1+j$ for some j with $1 \leq j \leq d$. For each j, there are $\binom{k+1+d}{k+1+j}$ choices for $S_{>q}$; to get the total we sum over j. So when n is odd, there are $2^{k-d}(\sum_{j=1}^{d} \binom{k+1+d}{k+1+j})$ choices for S.

If n is even, to fail condition (4), we need $|S_{>q}| = k+j$ for some j such that $1 \leq j \leq d$. For each j, there are $\binom{k+d}{k+j}$ choices for $S_{>q}$. Summing over j, when n is even, there are $2^{k-d}(\sum_{j=1}^{d} \binom{k+d}{k+j})$ choices for S.

Case 5: States of the form (q, S), $q > k+1$. Write $q = k+1+d$ with $d \geq 1$. Then $S_{<q}$ has size at most $k+1+d$. If n is odd, then $S_{>q}$ has size at most $k-1-d$. If n is even, then $S_{>q}$ has size at most $k-2-d$. We always have $|S_{>q}| \leq k-2$. Thus condition (1) (if n is odd) or (3) (if n is even) is met as long as $|S_{<q}| \leq k+1$. We count the number of sets that fail these conditions.

Write S as the disjoint union $S_{<q} \cup S_q \cup S_{>q}$. If n is odd, there are 2^{k-1-d} choices for $S_{>q}$. If n is even, there are 2^{k-2-d} choices for $S_{>q}$. There are two choices for S_q. To fail condition (1) or (3), whichever is relevant, we need $|S_{<q}| = k+j+1$ for some j with $1 \leq j \leq d$. For each j, there are $\binom{k+1+d}{k+1+j}$ choices. So if n is odd, there are $2^{k-d}(\sum_{j=1}^{d} \binom{k+1+d}{k+1+j})$ choices for S. If n is even, there are $2^{k-1-d}(\sum_{j=1}^{d} \binom{k+1+d}{k+1+j})$ choices for S.

This covers all cases, and taking the sum of all the above counts for each $q \in Q$ gives the lower bound $f(n)$ stated earlier.

Finally, we prove distinguishability of all the reached states. We assume that $n \geq 5$ here; the case $n = 4$ can be verified computationally. Distinguishability can be proved using just four monotonic transformations $\{a, b, c, d\}$, defined as follows for $q \in Q$:

- $qa = q+1$ if $0 \leq q \leq n-2$ and $(n-1)a = (n-1)$,
- $qb = q-1$ if $1 \leq q \leq n-1$ and $0b = 0$,
- $qc = 0$ if $0 \leq q \leq k$ and $qc = q$ otherwise,
- $qd = n-1$ if $k \leq q \leq n-1$ and $qd = q$ otherwise.

We claim that it suffices to prove the following statements:

- For each $q \in Q$, there is a string x_q that is accepted by q in A, not accepted by any other state of A, and not accepted by any state of B of the form $(p, \emptyset)$ for $p \in Q$.
- For each $q \in Q$, there is a string y_q that is accepted by $(q, \emptyset)$ in B, not accepted by $(p, \emptyset)$ for $p \neq q$, and not accepted by any state of A.

Indeed, let (p, S) and (q, T) be two distinct states of B. If $S \neq T$, then there exists a state r which belongs to the symmetric difference of S and T; then x_r distinguishes the states. If $S = T$, then $p \neq q$, and y_p distinguishes the states.

We define x_q as follows: If $0 \leq q \leq k-1$, then $x_q = a^{k-1-q}d$. If $q = k$, then $x_q = ac$. If $k+1 \leq q \leq n-1$, then $x_q = b^{q-(k+1)}c$. We define y_q as follows: If $0 \leq q \leq k$, then $y_q = a^{k+1-q}cac$. If $k+1 \leq q \leq n-1$, then $y_q = b^{q-(k+1)}cac$.

Now we verify these strings have the desired properties. Since $n \geq 5$, we have $n-1 \neq k+1$, and thus d sends every state of A to a non-final state except for $k-1$, which it fixes. Similarly, c sends every state of A to a non-final state except for $k+1$, which it fixes. It follows easily that x_q is accepted by q in A, but rejected in each other state of A.

Now consider $(p, \emptyset)$ in B; observe that for all $i \geq 0$, the state $(p, \emptyset)a^i$ does not contain $k-1$ in its second component. Therefore if $q \leq k-1$, then $(p, \emptyset)$ does not accept x_q. A similar argument works if $q \geq k+1$. For $q = k$, the second component of $(p, \emptyset)a$ cannot contain $k+1$, and it follows x_q is not accepted.

Next consider whether $(p, \emptyset)$ in B accepts y_q. Observe that $(p, \emptyset)a^{k+1-q}c$ contains k in the second component if and only if $p = q$. Thus $(p, \emptyset)a^{k+1-q}ca$ contains $k+1$ in the second component if and only if $p = q$. It follows then when $q \leq k+1$, the state $(p, \emptyset)$ accepts y_q if and only if $p = q$. If $q \geq k+1$, a similar argument applies. Finally, no state of A accepts y_q because k is not in the image of $a^{k+1-q}c$ or $b^{q-(k+1)}c$, but on the other hand, k is the only state of A which accepts ac. This completes the proof. □

4 Power, Positive Closure, and Complementation

Here we consider the k-th power, positive closure, and complementation on star-free and unary star-free languages.

By definition, every finite and every co-finite language is star-free. In the unary case, the minimal DFA for a star-free language must have the cycle of length one, because otherwise the string $w = a$ performs a non-trivial permutation on the states of this cycle. It follows that every unary star-free language is either finite or co-finite. Notice that the binary language $\{a, b\}^*a$ is star-free since its minimal DFA is permutation-free, but it is neither finite nor co-finite.

In the following four theorems, we consider the k-th power and the positive closure on star-free languages represented by DFAs and NFAs.

Theorem 4 (Power on Unary DFAs). *Let L be a unary star-free language with $\mathrm{sc}(L) \leq n$. Then $\mathrm{sc}(L^k) \leq k(n-1)+1$ and this bound is tight.*

Proof. The upper bound is the same as in the case of unary regular languages [14, Theorem 3]. For tightness, consider the co-finite language $L = a^{n-1}a^*$. We have $L^k = a^{k(n-1)}a^*$, which is a co-finite language with desired complexity. □

Theorem 5 (Power on NFAs). *Let $n, k \geq 2$. Let L be a star-free language over an alphabet Σ with $\mathrm{nsc}(L) \leq n$. Then $\mathrm{nsc}(L^k) \leq kn$, and this bound is tight if $|\Sigma| \geq 2$. In the unary case, a lower bound is $k(n-1)+1$.* □

Theorem 6 (Positive Closure on DFAs). *Let $n \geq 6$. Let L be a star-free language over an alphabet Σ with $\mathrm{sc}(L) \leq n$. Then*

- *$\mathrm{sc}(L^+) \leq 2^{n-1} + 2^{n-2} - 1$, and this bound is tight if $|\Sigma| \geq 4$;*
- *if $|\Sigma| = 1$, then $\mathrm{sc}(L^+) \leq n^2 - 7n + 13$, and this bound is tight.* □

Theorem 7 (Positive closure on NFAs). *Let L be a star-free language with* $\mathrm{nsc}(L) \leq n$. *Then* $\mathrm{nsc}(L^+) \leq n$, *and the bound is tight already in the unary case.*

Proof. The upper bound is the same as for regular languages. For tightness, consider the co-finite language $L = a^{n-1}a^*$. Then $L^+ = L$, so $\mathrm{nsc}(L^+) = n$. □

Next we consider complementation. In [7, Theorem 11], an upper bound $O(n^2)$ and a lower bound $(n-1)(n-2)$ were obtained. We provide a more precise upper bound and a better lower bound in the following theorem.

Theorem 8 (Complementation on Unary NFAs). *Let $n \geq 3$. Let L be a unary star-free language with* $\mathrm{nsc}(L) \leq n$. *Then* $\mathrm{nsc}(L^c) \leq n^2 - 2$. *There exists a unary star-free language L with* $\mathrm{nsc}(L^c) \geq (n-1)^2 + 1$.

Proof. First, let L be finite. Then the longest string in L is of length at most $n-1$. Thus L, as well as L^c, is accepted by a DFA with $n+1$ states, so $\mathrm{nsc}(L^c) \leq n+1$. Next, let L be co-finite. Recall that if we transform a unary n-state NFA for L to the Chrobak normal form, we get a tail with at most n^2-2 states and disjoint cycles of length $x_1, x_2, \ldots, x_k$ [5, Theorem 3.5]. The DFA equivalent to this NFA has a tail with at most n^2-2 states and a single cycle of length $\mathrm{lcm}(x_1, x_2, \ldots, x_k)$. Since L is co-finite, all states in this cycle must be final, so they can be merged into one state. Thus, in the minimal DFA for L, so also for L^c, the total number of states is at most $n^2 - 1$. Since the minimal DFA for L^c includes a state from which no string is accepted, we can omit it, and we get the desired upper bound. For the lower bound, consider the language L accepted by the NFA shown in Fig. 1. The language L consists of strings of length $cn + d(n-1)$ with $c \geq 1$ and $d \geq 0$ and the empty string. By [18, Lemma 5.1(ii)], the longest string in L^c is of length $(n-1)^2$. It follows that $\mathrm{nsc}(L^c) = (n-1)^2 + 1$. □

5 Conclusion

We examined the deterministic and nondeterministic state complexity of square, power, positive closure, and complementation on star-free languages. Our results are summarized in Tables 1 and 2 where also the size of alphabet used to describe witnesses is displayed. The tables also show all known results concerning descriptional complexity of basic regular operations on star-free and regular languages. Notice that the deterministic state complexity of square on star-free languages and the nondeterministic state complexity of union, intersection, concatenation, power, and complementation on unary star-free languages remain open.

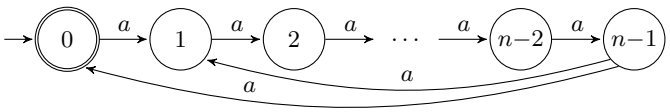

Fig. 1. A lower bound example for complementation on unary star-free languages.

	Star-free	$\lvert\Sigma\rvert$	Source	Regular	$\lvert\Sigma\rvert$	Source
$\mathrm{sc}(K \cup L)$	mn	2	[1, Theorem 1]	mn	2	[12, (1)]
$\mathrm{sc}(K \cap L)$	mn	2	[1, T1]	mn	2	[18, T4.3]
$\mathrm{sc}(KL)$	$m2^n - 2^{n-1}$	4	[1, T2]	$m2^n - 2^{n-1}$	2	[12, (2)]
$\mathrm{sc}(L^*)$	$(3/4)2^n$	4	[1, T4]	$(3/4)2^n$	2	[12, (3)]
$\mathrm{sc}(L^R)$	$2^n - 1$	$n-1$	[1, T5]	2^n	2	[11, Prop.1]
$\mathrm{sc}(L^2)$	$\circ$	$\binom{2n-1}{n-1}$	**here, Sect. 3**	$n2^n - 2^{n-1}$	2	[14, T1]
$\mathrm{sc}(L^+)$	$\mathbf{(3/4)2^n - 1}$	**4**	**here, T6**	$(3/4)2^n - 1$	2	[12, (3)]
$\mathrm{nsc}(K \cup L)$	$m+n+1$	2	[7, T2]	$m+n+1$	2	[6, T1]
$\mathrm{nsc}(K \cap L)$	mn	2	[7, T3]	mn	2	[6, T3]
$\mathrm{nsc}(KL)$	$m+n$	2	[7, T6]	$m+n$	2	[6, T7]
$\mathrm{nsc}(L^*)$	$n+1$	1	[7, T13]	$n+1$	1	[6, T9]
$\mathrm{nsc}(L^R)$	$n+1$	2	[7, T8]	$n+1$	2	[10, T2]
$\mathrm{nsc}(L^c)$	2^n	2	[7, T5]	2^n	2	[10, T5]
$\mathrm{nsc}(L^k)$	$\boldsymbol{kn}$	**2**	**here, T5**	kn	2	[4, T3]
$\mathrm{nsc}(L^+)$	$\boldsymbol{n}$	**1**	**here, T7**	n	1	[6, T9]

Table 1. Descriptional complexity of operations on star-free and regular languages. For the state complexity of square on star-free languages, we have $\sum_{i=0}^{n-1} g(n,i) \le \circ \le (n-1)2^n - 2(n-2)$ where $g(n,i)$ is defined by (1) on p. 5.

	Unary star-free	Source	Unary regular	Source
$\mathrm{sc}(K \cup L)$	$\max\{m,n\}$	[1, T6(1)]	mn; $\gcd(m,n)=1$	[13, T4]
$\mathrm{sc}(K \cap L)$	$\max\{m,n\}$	[1, T6(1)]	mn; $\gcd(m,n)=1$	[13, T4]
$\mathrm{sc}(KL)$	$m+n-1$	[1, T6(2)]	mn; $\gcd(m,n)=1$	[18, T5.4]
$\mathrm{sc}(L^*)$	$n^2 - 7n + 13$	[1, T6(3)]	$(n-1)^2 + 1$	[18, T5.3]
$\mathrm{sc}(L^k)$	$\boldsymbol{k(n-1)+1}$	**here, T4**	$k(n-1)+1$	[14, T4]
$\mathrm{sc}(L^+)$	$\boldsymbol{n^2 - 7n + 13}$	**here, T6**	$(n-1)^2$	[18, T5.3]
$\mathrm{nsc}(K \cup L)$	$m+n \le \cdot \le m+n+1$	[7, T9]	$m+n+1$; $m \ne kn$	[6, T2]
$\mathrm{nsc}(K \cap L)$	$\Theta(m^2)$; $n = m+1$	[7, T10]	mn; $\gcd(m,n)=1$	[6, T4]
$\mathrm{nsc}(KL)$	$m+n-1 \le \cdot \le m+n$	[7, T12]	$m+n-1 \le \cdot \le m+n$	[6, T8]
$\mathrm{nsc}(L^*)$	$n+1$	[7, T13]	$n+1$	[6, T9]
$\mathrm{nsc}(L^k)$	$\boldsymbol{k(n-1)+1 \le \cdot \le kn}$	**here, T5**	$k(n-1)+1 \le \cdot \le kn$	[6, T8]
$\mathrm{nsc}(L^+)$	$\boldsymbol{n}$	**here, T7**	n	[6, T9]
$\mathrm{nsc}(L^c)$	$\boldsymbol{(n-1)^2+1 \le \cdot \le n^2-2}$	**here, T8**	$2^{\Theta(\sqrt{n \log n})}$	[6, T6]

Table 2. Descriptional complexity of operations on unary star-free and unary regular languages.

References

1. Brzozowski, J.A., Liu, B.: Quotient complexity of star-free languages. Internat. J. Found. Comput. Sci. **23**(6), 1261–1276 (2012). https://doi.org/10.1142/S0129054112400515
2. Brzozowski, J.A., Szykula, M.: Large aperiodic semigroups. Internat. J. Found. Comput. Sci. **26**(7), 913–932 (2015). https://doi.org/10.1142/S0129054115400067
3. Câmpeanu, C., Culik II, K., Salomaa, K., Yu, S.: State complexity of basic operations on finite languages. In: Boldt, O., Jürgensen, H. (eds.) WIA 1999. LNCS, vol. 2214, pp. 60–70. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-45526-4_6
4. Domaratzki, M., Okhotin, A.: State complexity of power. Theoret. Comput. Sci. **410**(24-25), 2377–2392 (2009). https://doi.org/10.1016/j.tcs.2009.02.025
5. Geffert, V.: Magic numbers in the state hierarchy of finite automata. Inform. and Comput. **205**(11), 1652–1670 (2007). https://doi.org/10.1016/j.ic.2007.07.001
6. Holzer, M., Kutrib, M.: Nondeterministic descriptional complexity of regular languages. Internat. J. Found. Comput. Sci. **14**(6), 1087–1102 (2003). https://doi.org/10.1142/S0129054103002199
7. Holzer, M., Kutrib, M., Meckel, K.: Nondeterministic state complexity of star-free languages. Theoret. Comput. Sci. **450**, 68–80 (2012). https://doi.org/10.1016/j.tcs.2012.04.028
8. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley, Boston (1979)
9. Howie, J.M.: Products of idempotents in certain semigroups of transformations. Proc. Edinburgh Math. Soc. **17**(3), 223–236 (1971). https://doi.org/10.1017/S0013091500026936
10. Jirásková, G.: State complexity of some operations on binary regular languages. Theoret. Comput. Sci. **330**(2), 287–298 (2005). https://doi.org/10.1016/j.tcs.2004.04.011
11. Leiss, E.L.: Succint representation of regular languages by boolean automata. Theoret. Comput. Sci. **13**, 323–330 (1981). https://doi.org/10.1016/S0304-3975(81)80005-9
12. Maslov, A.N.: Estimates of the number of states of finite automata. Soviet Math. Doklady **11**, 1373–1375 (1970)
13. Pighizzini, G., Shallit, J.: Unary language operations, state complexity and Jacobsthal's function. Internat. J. Found. Comput. Sci. **13**(1), 145–159 (2002). https://doi.org/10.1142/S012905410200100X
14. Rampersad, N.: The state complexity of L^2 and L^k. Inform. Process. Lett. **98**(6), 231–234 (2006). https://doi.org/10.1016/j.ipl.2005.06.011
15. Schützenberger, M.P.: On finite monoids having only trivial subgroups. Information and Control **8**(2), 190–194 (1965). https://doi.org/10.1016/S0019-9958(65)90108-7
16. Sipser, M.: Introduction to the theory of computation. Cengage Learning (2012)
17. Yu, S.: Regular languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, Vol. 1, pp. 41–110. Springer, Heidelberg (1997). https://doi.org/10.1007/978-3-642-59136-5_2
18. Yu, S., Zhuang, Q., Salomaa, K.: The state complexities of some basic operations on regular languages. Theoret. Comput. Sci. **125**(2), 315–328 (1994). https://doi.org/10.1016/0304-3975(92)00011-F