Square on Deterministic, Alternating, and Boolean Finite Automata

Ivana Krajňáková(✉) and Galina Jirásková

Mathematical Institute, Slovak Academy of Sciences,
Grešákova 6, 040 01 Košice, Slovakia
{krajnakova,jiraskov}@saske.sk

Abstract. We investigate the state complexity of the square operation on languages represented by deterministic, alternating, and Boolean automata. For each k such that $1 \leq k \leq n-2$, we describe a binary language accepted by an n-state DFA with k final states meeting the upper bound $n2^n - k2^{n-1}$ on the state complexity of its square. We show that in the case of $k = n-1$, the corresponding upper bound cannot be met. Using the DFA witness for square with 2^n states where half of them are final, we get the tight upper bounds on the complexity of the square operation on alternating and Boolean automata.

1 Introduction

Square is a basic unary operation on formal languages which is defined as $L^2 = \{uv \mid u \in L \text{ and } v \in L\}$. It is known that if a language L is accepted by a deterministic finite automaton (DFA) of n states, then the language L^2 is accepted by a DFA of at most $n2^n - 2^{n-1}$ states [7]. This upper bound was proven to be tight in the binary case by Rampersad [8]. If the minimal DFA for L has more than one final state, then this upper bound cannot be met. In such a case the upper bound is $n2^n - k2^{n-1}$, where k is the number of final states in the minimal DFA for L [10].

In this paper we study the state complexity of the square of languages accepted by DFAs with more final states. Our motivation comes from the paper by Fellah et al. [3] on alternating finite automata (AFAs). They provided an upper bound $2^n + n + 1$ on the complexity of the square of a language represented by an n-state AFA. A language is accepted by an n-state AFA if and only if its reverse is accepted by a DFA with 2^n states where 2^{n-1} of them are final [1,3,5]. It follows that to prove the tightness of the upper bound $2^n + n + 1$, we need to find a language represented by a DFA with half of the states final which is hard for the square operation on DFAs.

Research supported by grant VEGA 2/0084/15 and grant APVV-15-0091. This work was conducted as a part of PhD study of the first author at Comenius University in Bratislava.

G. Pighizzini and C. Câmpeanu (Eds.): DCFS 2017, LNCS 10316, pp. 214–225, 2017.
DOI: 10.1007/978-3-319-60252-3_17

The problem seems to be interesting per se. Previously in [2], we tried to use Rampersad's binary witness for square [8] with k final states instead of original one. We were able to show the reachability of $n2^n - k2^{n-1}$ states in the subset automaton of an NFA for its square. However, to prove distinguishability a third letter was needed, so the binary case was left open. Surprisingly, in [2], we were unable to prove the tightness of the upper bound in the case of $n-1$ final states.

Here we solve both these open problems. We describe a binary language accepted by an n-state DFA with k final states meeting the upper bound $n2^n - k2^{n-1}$ on the state complexity of its square providing that $1 \leq k \leq n-2$. In the case of $k = n-1$, we prove that the corresponding upper bound $(2n+2)2^{n-2}$ cannot be met. To show it, we consider two cases. If the initial state is final, then we get the upper bound $(n+2)2^{n-2}$, and we show that it is tight in the binary case. If the initial state is not final, then the upper bound is $(n+3)2^{n-2}$ and is tight in the ternary case. The tight bound for binary languages is $(n+3)2^{n-2}-1$ in this case. This solves the complexity of square on DFAs completely. The binary alphabet is optimal since it is known that in the unary case, the tight upper bound is $2n-1$ [8].

Using these results we are able to describe a binary language accepted by an n-state AFA such that every AFA for its square has at least $2^n + n + 1$ states. This proves the tightness of the upper bound $2^n + n + 1$ given in [3]. We also consider Boolean finite automata (BFA) [1], and get the tight upper bound $2^n + n$ for the square on BFAs. To prove these results, we take the reversal of a language accepted by a DFA with 2^n states with half of them final meeting the corresponding upper bound for square on DFAs. Then this language is accepted by an n-state BFA, and we are able to prove that every BFA for its square has at least $2^n + n$ states. By more careful analysis of the number of final states in DFA for its square, we get the lower bound $2^n + n + 1$ for the square operation on AFAs. Our result can be extended for the concatenation operation just by concatenating two of our automata with different number of states. This provides an alternative proof of the tightness of the upper bound $2^m + n + 1$ for the concatenation operation on alternating automata with m and n states [4].

2 Preliminaries

Let Σ be a finite alphabet of symbols. Then Σ^* denotes the set of words over Σ including the empty word ε. A language is any subset of Σ^*. The concatenation of languages K and L is the language $KL = \{uv \mid u \in K \text{ and } v \in L\}$. The square of a language L is the language $L^2 = LL$. The cardinality of a finite set A is denoted by $|A|$, and its power-set by 2^A. The reader may refer to [9] for details.

A *nondeterministic finite automaton* (NFA) is a quintuple $A = (Q, \Sigma, \circ, I, F)$, where Q is a finite set of states, Σ is a finite non-empty alphabet, $\circ : Q \times \Sigma \to 2^Q$ is the transition function which is naturally extended to the domain $2^Q \times \Sigma^*$, $I \subseteq Q$ is the set of initial states, and $F \subseteq Q$ is the set of final states. The *language accepted by* A is the set $L(A) = \{w \in \Sigma^* \mid I \circ w \cap F \neq \emptyset\}$. For a symbol a, we say that (p, a, q) is a transition in NFA A if $q \in p \circ a$, and for

a word w, we write $p \xrightarrow{w} q$ if $q \in p \circ w$. An NFA A is *deterministic* (DFA) (and complete) if $|I| = 1$ and $|q \circ a| = 1$ for each q in Q and each a in Σ. We write $p \cdot a = q$ instead of $p \circ a = \{q\}$ in such a case. The *state complexity* of a regular language L, $\mathrm{sc}(L)$, is the smallest number of states in any DFA for L.

Every NFA $A = (Q, \Sigma, \circ, I, F)$ can be converted to an equivalent DFA $A' = (2^Q, \Sigma, \cdot, I, F')$, where $R \cdot a = R \circ a$ for each R in 2^Q and a in Σ, and $F' = \{R \in 2^Q \mid R \cap F \neq \emptyset\}$. We call the DFA A' the *subset automaton* of the NFA A. The subset automaton may not be minimal since some of its states may be unreachable or equivalent to other states.

A *Boolean finite automaton* (BFA) is a quintuple $A = (Q, \Sigma, \delta, g_s, F)$, where Q is a finite non-empty set of states, $Q = \{q_1, \ldots, q_n\}$, Σ is an input alphabet, δ is the transition function that maps $Q \times \Sigma$ into the set $\mathcal{B}_n$ of Boolean functions with variables $\{q_1, \ldots, q_n\}$, $g_s \in \mathcal{B}$ is the initial Boolean function, and $F \subseteq Q$ is the set of final states. The transition function δ is extended to the domain $\mathcal{B}_n \times \Sigma^*$ as follows: For all g in $\mathcal{B}_n$, a in Σ, and w in Σ^*, we have $\delta(g, \varepsilon) = g$; if $g = g(q_1, \ldots, q_n)$, then $\delta(g, a) = g(\delta(q_1, a), \ldots, \delta(q_n, a))$; $\delta(g, wa) = \delta(\delta(g, w), a)$. Next, let $f = (f_1, \ldots, f_n)$ be the Boolean vector with $f_i = 1$ iff $q_i \in F$. The language accepted by the BFA A is the set $L(A) = \{w \in \Sigma^* \mid \delta(g_s, w)(f) = 1\}$.

A Boolean finite automaton is called *alternating* (AFA) if the initial function is a projection $g(q_1, \ldots, q_n) = q_i$. For details, the reader may refer to [1,3,5,6,9]. The Boolean (alternating) state complexity of L, $\mathrm{bsc}(L)(\mathrm{asc}(L))$, is the smallest number of states in any BFA (AFA) for L. It is known that a language L is accepted by an n-state BFA (AFA) if and only if the language L^R is accepted by an 2^n-state DFA (with 2^{n-1} final states). We state it in the next two facts.

Fact 1 (cf. [3] Theorem 4.1, Corollary 4.2 and [5], Lemma 1). *Let L be a language accepted by an n-state BFA (AFA). Then the reversal L^R is accepted by a DFA of 2^n states (of which 2^{n-1} are final).* □

Corollary 2. *If L is a regular language, than* $\mathrm{bsc}(L) \geq \lceil \log(\mathrm{sc}(L^R)) \rceil$ *and* $\mathrm{asc}(L) \geq \lceil \log(\mathrm{sc}(L^R)) \rceil$. □

Fact 3 (cf. [5], Lemma 2). *If L^R be accepted by a DFA A of 2^n states, then L is accepted by an n-state BFA. If L^R be accepted by a DFA A of 2^n states of which 2^{n-1} are final, then L is accepted by an n-state AFA.* □

3 Square on DFAs

Let us begin with the precise method of construction an NFA for the square of some languages accepted by a minimal DFA with n states.

Construction 4. (DFA $A \to$ NFA N for $L^2(A)$)
Let $A = (\{q_0, q_1, \ldots, q_{n-1}\}, \Sigma, \cdot, q_0, F_A)$ be a minimal DFA. We construct NFA $N = (\{q_0, q_1, \ldots, q_{n-1}\} \cup \{0, 1, \ldots, n-1\}, \Sigma, \circ, I, F_N)$ as follows:

(a) take A and add a copy of A with the state set $\{0, 1, \ldots, n-1\}$;
(b) for each symbol a and each state q_i with $q_i \cdot a \in F_A$, add transition $(q_i, a, 0)$;

(c) the set of initial states of N is $I = \{q_0\}$ if $q_0 \notin F$, and $I = \{q_0, 0\}$ otherwise;
(d) the set of final state of N is $F_N = \{j \in \{0, 1, \ldots, n-1\} \mid q_j \in F_A\}$.

Proposition 5 (Upper Bound). *Let L be a language with $\mathrm{sc}(L) = n$, and let the minimal DFA for L have k final states. Then $\mathrm{sc}(L^2) \leq n2^n - k2^{n-1}$.*

Proof. Let L be accepted by DFA $A = (\{q_0, q_1, \ldots, q_{n-1}\}, \Sigma, \cdot, q_0, F_A)$ and let $|F_A| = k$. Construct an NFA N for L^2 as described above. Since A is deterministic, every reachable subset in the subset automaton of N is in the form of $\{q_i\} \cup S$, where $S \subseteq \{0, 1, \ldots, n-1\}$. Furthermore, if q_i is a final state of A, then $0 \in S$ because of the used construction. It follows that subsets containing a final state of A and missing 0 are unreachable. Hence the subset automaton of N has at most $n2^n - k2^{n-1}$ reachable states. □

Notice that the upper bound given by above proposition is maximal if $k = 1$, and it is $n2^n - 2^{n-1}$ in this case. The binary witness language meeting this bound was presented by Rampersad in 2006 [8].

Theorem 6 [8, Theorem 1]**.** *For every integer $n \geq 3$, there exists a DFA M with n states such that the minimal DFA accepting the language $L^2(M)$ has $n2^n - 2^{n-1}$ states.* □

Unfortunately, the square of Rampersad's automaton with k final states does not meet the upper bound on the state complexity in the general case. Here we provide the binary witness automaton with k final states that meets the upper bound $n2^n - k2^{n-1}$.

Theorem 7. *Let $n \geq 3$ and $1 \leq k \leq n-2$. Then there exists a minimal n-state DFA A with k final states defined over a binary alphabet such that every DFA for $L(A)^2$ has at least $n2^n - k2^{n-1}$ states.*

Proof. Let us take n-state DFA $A = (\{q_0, q_1, \ldots, q_{n-1}\}, \Sigma, \cdot, q_0, F_A)$ with k final states shown in Fig. 1. Notice that q_0 and q_1 remain non-final with every k in this DFA and there are two cycles; one on a, $(q_0, q_1, \ldots, q_{n-1})$, of length n and the second on b, $(q_2, q_3, \ldots, q_{n-1})$, of length $n-2$.

Let us build an NFA N for $L(A)^2$ as in Construction 4. An example of NFA N if $n = 6$ and $k = 2$ is shown in Fig. 2.

We observe that there are only two types of states reachable in the subset automaton of N:

- $\{q_i\} \cup S$, where $S \subseteq \{0, 1, \ldots, n-1\}$ and $0 \leq i \leq n-k-1$;
- $\{q_i, 0\} \cup S$, where $S \subseteq \{1, \ldots, n-1\}$ and $n-k \leq i \leq n-1$.

We denote this family of sets as $\mathcal{R}$. We can see that in $\mathcal{R}$ there are exactly $(n-k)2^n + k2^{n-1} = n2^n - k2^{n-1}$ sets. Our goal is to show that the sets in $\mathcal{R}$ are reachable and also pairwise distinguishable in the subset automaton of N for $L(A)^2$.

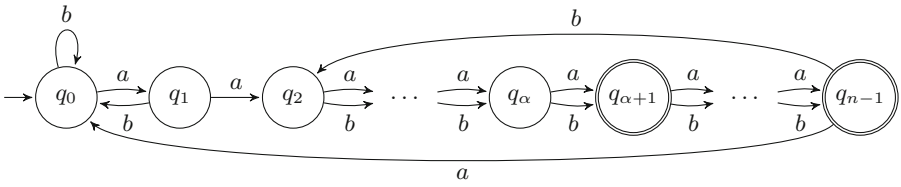

Fig. 1. A witness DFA A with k final states meeting the bound $n2^n - k2^{n-1}$, where $\alpha = n - k - 1$.

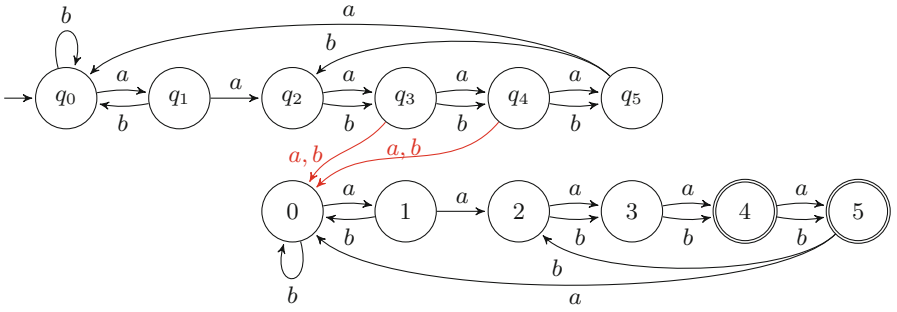

Fig. 2. NFA N for square of $L(A)$, if $n = 6$ and $k = 2$.

Let us start with reachability. We use mathematical induction by number of elements in set/state. The sets with one and two elements are reachable, because:

$$\to \{q_0\} \xrightarrow{a} \{q_1\} \xrightarrow{a} \cdots \xrightarrow{a} \{q_{n-k-1}\} \xrightarrow{a} \{q_{n-k}, 0\},$$

$$\{q_{n-k}, 0\} \xrightarrow{b} \{q_{n-k+1}, 0\} \xrightarrow{b} \cdots \xrightarrow{b} \{q_{n-2}, 0\} \xrightarrow{b} \{q_{n-1}, 0\},$$

$$\{q_{n-1}, 0\} \xrightarrow{a} \{q_0, 1\} \xrightarrow{b} \{q_0, 0\},$$

$$\{q_0, 1\} \xrightarrow{a} \{q_1, 2\} \xrightarrow{b} \{q_0, 3\} \xrightarrow{b} \{q_0, 4\} \xrightarrow{b} \cdots \xrightarrow{b} \{q_0, n-1\} \xrightarrow{b} \{q_0, 2\},$$

$$\{q_0, (j-i) \bmod n\} \xrightarrow{a^i} \{q_i, j\} \text{ for } i = 0, 1, \ldots, n-k-1 \text{ and } j = 0, 1, \ldots, n-1.$$

Assume now that every set in $\mathcal{R}$ with t elements is reachable. We show that then every set in $\mathcal{R}$ of size $t+1$ is reachable. Let $S = \{q_i, s_1, s_2, \ldots, s_t\}$ be our desired set in $\mathcal{R}$ of size $t+1$, where $q_i \in Q$ and $0 \le s_1 < s_2 < \cdots < s_t \le n-1$. We deal with three cases:

(1) We show the reachability of sets of the second type, so let $n-k \le i \le n-1$ and therefore $s_1 = 0$. We can write i as $i = \alpha + \beta$, where $\alpha = n-k-1$ and $0 \le \beta \le k$, so our desired set is $S = \{q_{\alpha+\beta}, 0, s_2, s_3, \ldots, s_t\}$.

Let $s_2 = 1$, and take the set $\{q_{\alpha+\beta-1}, 0, s_3 - 1, \ldots, s_t - 1\}$, which is in $\mathcal{R}$ and is reachable because it has t elements. Then we have

$$\{q_{\alpha+\beta-1}, 0, s_3 - 1, \ldots, s_t - 1\} \xrightarrow{a} \{q_{\alpha+\beta}, 0, 1, s_3, \ldots, s_t\} = S.$$

Let $s_2 \geq 2$ and take the set $\{q_\alpha, s_2 \cdot b^{n-1-\beta} - 1, \ldots, s_t \cdot b^{n-1-\beta} - 1\}$, which is in $\mathcal{R}$ and is reachable because it has t elements. Then we have

$$\{q_\alpha, s_2 \cdot b^{n-1-\beta} - 1, \ldots, s_t \cdot b^{n-1-\beta} - 1\} \xrightarrow{a} \{q_{\alpha+1}, s_2 \cdot b^{n-1-\beta}, \ldots, s_t \cdot b^{n-1-\beta}\}$$

$$\xrightarrow{b^{\beta-1}} \{q_{\alpha+\beta}, 0, s_2 \cdot b^{n-2}, \ldots, s_t \cdot b^{n-2}\} = \{q_{\alpha+\beta}, 0, s_2, \ldots, s_t\} = S.$$

(2) Next we show the reachability of sets of the first type in the next two steps. Let $i = 0$. We distinguish between three cases of s_1.

Firstly let $s_1 = 0$. We start from the set reached previously in (1) to achieve S in case of $s_2 = 1$ by $\{q_{n-1}, 0, s_3 - 1, \ldots, s_t - 1, n-1\} \xrightarrow{a} \{q_0, 0, 1, s_3, \ldots, s_t\} = S$. Otherwise, if desired $s_2 \geq 2$, we reach S using previously reached set

$$\{q_0, 0, 1, s_3 - s_2 + 1, \ldots, s_t - s_2 + 1\} \xrightarrow{a} \{q_1, 1, 2, s_3 - s_2 + 2, \ldots, s_t - s_2 + 2\}$$

$$\xrightarrow{b^{n-2}} \{q_0, 0, 2, s_3 - s_2 + 2, \ldots, s_t - s_2 + 2\} \xrightarrow{b^{s_2-2}} \{q_0, 0, s_2, \ldots, s_t\} = S.$$

Secondly let $s_1 \geq 1$. Then the set $S' = \{q_{n-1}, 0, s_2 - s_1, \ldots, s_t - s_1\}$ is reached in (1). If $s_1 = 1$, then $S' \xrightarrow{a} S$, otherwise $s_1 \geq 2$, and $S' \xrightarrow{aab^{n-2}b^{s_1-2}} S$.

(3) Let $1 \leq i \leq n-k-1$. Now we can reach the remaining sets of the first type using sets achieved in (2) like this $\{q_0, (s_1 - i) \bmod n, \ldots, (s_t - i) \bmod n\} \xrightarrow{a^i} \{q_i, s_1, \ldots, s_t\} = S$.

Let us continue with proving distinguishability of reached sets. Note that in N we have

$$\{n-1\} \xrightarrow{b} \{2\} \xrightarrow{a} \{3\} \xrightarrow{b^{n-2}} \{3\} \xrightarrow{ab^{n-2}} \{4\} \xrightarrow{ab^{n-2}} \cdots \xrightarrow{ab^{n-2}} \{n-1\}.$$

This means that the word $w = b(ab^{n-2})^{n-3}$ is accepted from the state $n-1$. Let us read w from a different state t, $2 \leq t \leq n-2$. First we have $t \circ b \in \{3, 4, \ldots, n-1\}$. Next $\{3, 4, \ldots, n-1\} \circ (ab^{n-2})^{n-3} = \{0\}$, so w is not accepted from t. Similarly, reading w from $\{0, 1\}$ results in the set $\{0\}$, thus w is not accepted from $\{0, 1\}$ either. Moreover, w is not accepted from $\{q_i\}$, because $\{q_i\} \circ w \subseteq \{q_j, 0\}$, where either $j = 0$ if $i < n-1$, or $j = n-1$ if $i = n-1$. Therefore w is accepted by N from and only from the state $n-1$. Notice that each state t in $\{1, 2, \ldots, n-1\}$ has exactly one in-transition on a going from the state $t-1$, so the word $a^{n-1-t}w$ is accepted by N only from state t, $0 \leq t \leq n-2$. It follows that two sets $\{q_i\} \cup S$ and $\{q_j\} \cup T$ in $\mathcal{R}$ are distinguishable if $S \neq T$.

Now consider two distinct subsets $\{q_i\} \cup S$ and $\{q_j\} \cup S$ in $\mathcal{R}$. Without loss of generality, we have $0 \leq i < j \leq n-1$. We will discuss three cases:

(1) Let $i = 0$ and $j = 1$. Then

$$\{q_0\} \cup S \xrightarrow{(ab^{n-2})^{n-2}} \{q_0, 0\} \xrightarrow{a} \{q_1, 1\} \xrightarrow{a^{n-k-1}} \{q_{n-k}, 0, n-k\},$$

$$\{q_1\} \cup S \xrightarrow{(ab^{n-2})^{n-2}} \{q_{n-1}, 0\} \xrightarrow{a} \{q_0, 1\} \xrightarrow{a^{n-k-1}} \{q_{n-k-1}, n-k\}.$$

Now we can distinguish these sets because they differ in the element from the second automaton copy.

(2) Let $i = 0$ and $j \geq 2$. Then

$$\{q_0\} \cup S \xrightarrow{b^{n-1-j}a} \{q_1\} \cup S_1,$$

$$\{q_j\} \cup S \xrightarrow{b^{n-1-j}a} \{q_0\} \cup S_1'.$$

If the subsets S_1 and S_1' are the same, then we continue as in (1), otherwise we continue as in case of $S \neq T$.

(3) Let $i \geq 1$. Then

$$\{q_i\} \cup S \xrightarrow{a^{n-j}} \{q_{i+(n-j)}\} \cup S_1,$$

$$\{q_j\} \cup S \xrightarrow{a^{n-j}} \{q_0\} \cup S_1'.$$

Similarly as in (2), if the subsets S_1 and S_1' are the same we continue as in (1) or (2), otherwise we continue as in case of $S \neq T$. □

3.1 Square if $|F| = n - 1$

Recall that the automaton in the proof of Theorem 7 must have at least two non-final states. We show that for every language L accepted by an n-state DFA $A = (Q, \Sigma, \cdot, q_0, F)$ with a single non-final state, the state complexity of L^2 never meets the upper bound set in Proposition 5. In particular, we show:

(a) if $q_0 \in F$, then $\mathrm{sc}(L^2) \leq (n+2)2^{n-2}$ and this bound is tight if $|\Sigma| \geq 2$;
(b) if $q_0 \notin F$, then $\mathrm{sc}(L^2) \leq (n+3)2^{n-2}$ and this bound is tight if $|\Sigma| \geq 3$.

First, we consider the case of $|F| = n - 1$ and $q_0 \in F$.

Lemma 8. *Let $n \geq 3$ and let L be a regular language accepted by an n-state DFA $A = (Q, \Sigma, \delta, q_0, F)$ with $n-1$ final states, where $q_0 \in F$. Then $\mathrm{sc}(L^2) \leq (n+2)2^{n-2}$, and this bound is tight if $|\Sigma| \geq 2$.*

Proof. The formula for the upper bound is based on the observation that q_0 is initial and also accepting in A, so the initial state in the subset automaton for $L(A)^2$ is $\{q_0, 0\}$. It follows that for every $i \in \{0, 1, \ldots, n-1\}$ if $\{q_i\} \cup X$ is reachable, then $i \in X$. So we consider the following family $\mathcal{R}$ of possible sets in the subset automaton for $L(A)^2$:

$$\begin{aligned}\mathcal{R} = &\big\{\{q_0, 0\} \cup X \mid X \subseteq \{1, 2, \ldots, n-1\}\big\} \cup \\ &\big\{\{q_1, 1\} \cup X \mid X \subseteq \{0, 2, 3, \ldots, n-1\}\big\} \cup \\ &\big\{\{q_i, 0, i\} \cup X \mid 2 \leq i \leq n-1, X \subseteq \{1, 2, \ldots, n-1\} \setminus \{i\}\big\}.\end{aligned}$$

This family consists of $(n+2)2^{n-2}$ sets. Hence $\mathrm{sc}(L^2) \leq (n+2)2^{n-2}$. To prove the tightness of this upper bound, we introduce the DFA A shown in Fig. 3 and we show that every DFA for $L(A)^2$ has at least $(n+2)2^{n-2}$ states. Notice that A has the same structure as the DFA in the Fig. 1, so the proof continues similarly as the proof of Theorem 7. □

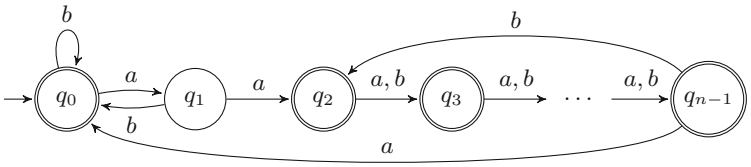

Fig. 3. A witness DFA A with $n-1$ final states meeting the bound $(n+2)2^{n-2}$, where $q_0 \in F$.

Now we consider the case where $|F| = n-1$ and $q_0 \notin F$.

Lemma 9. *Let $n \geq 3$. Let L be a regular language accepted by an n-state DFA $A = (Q, \Sigma, \cdot, q_0, F)$, where $|F| = n-1$ and $q_0 \notin F$. Then $\mathrm{sc}(L^2) \leq (n+3)2^{n-2}$, and the bound is tight if $|\Sigma| \geq 3$. The bound $(n+3)2^{n-2} - 1$ can be met by a binary language.*

Proof. We start with the upper bound. Suppose we have constructed an NFA N from the DFA A as described in Construction 4. Consider the corresponding subset automaton of N. We first show that two distinct subsets of this automaton, $\{q_i\} \cup S$ and $\{q_j\} \cup S$, where $\{i, j\} \subseteq S$ are equivalent. If a word w is rejected from state $\{q_i\} \cup S$ then $s \xrightarrow{w} 0$ for each element s in S. It follows that w is rejected from $\{q_j\} \cup S$ because $\{q_j\} \cup S \xrightarrow{w} \{q_0, 0\}$. Likewise, if w is rejected from $\{q_j\} \cup S$ then w is rejected from $\{q_i\} \cup S$. Excluding these equivalent subsets gives us the family $\mathcal{R}$ of $(n+3)2^{n-2}$ reachable and pairwise distinguishable subsets of the subset automaton of N, which is:

$$\begin{aligned}\mathcal{R} = &\{\{q_0\} \cup X \mid X \subseteq \{0, 1, \ldots, n-1\}\} \quad \cup \\ &\{\{q_i\} \cup X \mid X \subseteq \{0, 1, \ldots, n-1\}, 0 \in X, i \notin X\}.\end{aligned}$$

To prove the tightness of this upper bound, we introduce the DFA B shown in Fig. 4 and we show that every DFA for $L(B)^2$ has at least $(n+3)2^{n-2}$ states. Construct an NFA N for the square of $L(B)^2$ as described in Construction 4. Let us show that each set in $\mathcal{R}$ is reachable in the subset automaton of N and that all these sets are pairwise distinguishable.

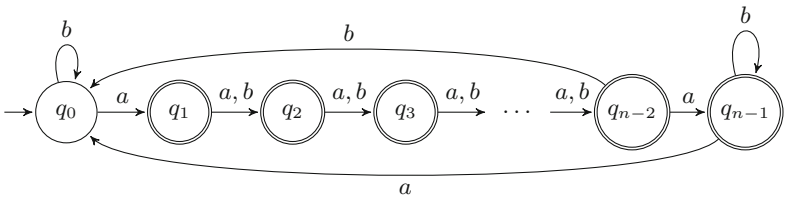

Fig. 4. A binary DFA B with $\mathrm{sc}(L^2(B)) = (n+3)2^{n-2} - 1$.

We prove the reachability by induction on the size of subsets. The basis, where $|S| \leq 2$, holds true up to one set, namely $\{q_0, n-1\}$, since we have

$$\to \{q_0\} \xrightarrow{a} \{q_1, 0\} \xrightarrow{b} \{q_2, 0\} \xrightarrow{b} \cdots \xrightarrow{b} \{q_{n-2}, 0\} \xrightarrow{b} \{q_0, 0\},$$

$$\{q_{n-2}, 0\} \xrightarrow{a} \{q_{n-1}, 0, 1\} \xrightarrow{b} \{q_{n-1}, 0, 2\} \xrightarrow{b} \cdots \xrightarrow{b} \{q_{n-1}, 0, n-2\} \xrightarrow{b} \{q_{n-1}, 0\},$$

$$\{q_{n-1}, 0\} \xrightarrow{a} \{q_0, 1\} \xrightarrow{b} \{q_0, 2\} \xrightarrow{b} \cdots \xrightarrow{b} \{q_0, n-2\}.$$

We deal with $\{q_0, n-1\}$ later. Now assume that each set in $\mathcal{R}$ of size t is reachable. Let $S = \{q_i, s_1, s_2, \ldots, s_t\}$ be a set of size $t+1$. Consider several cases.

(1) Let $i = 1$, so $s_1 = 0$. Then $\{q_0, s_2 - 1, \ldots, s_t - 1\} \xrightarrow{a} \{q_1, 0, s_2, \ldots, s_t\}$, where the former set of size t is reachable by induction hypothesis.

(2) Let $1 \leq i \leq n-2$, so $S = \{q_i, 0, s_2, s_3, \ldots, s_t\}$.

$$\begin{aligned}
&\text{If } s_2 = 1, \text{ then} && \{q_{i-1}, 0, s_3 - 1, \ldots, s_t - 1\} \xrightarrow{a} S.\\
&\text{If } s_2 \geq 2 \text{ and } s_t \leq n-2, \text{ then} && \{q_{i-1}, 0, s_2 - 1, \ldots, s_t - 1\} \xrightarrow{b} S.\\
&\text{If } s_2 \geq 2 \text{ and } s_t = n-1, \text{ then} && \{q_{i-1}, 0, s_2 - 1, \ldots, s_{t-1} - 1, n-1\} \xrightarrow{b} S.
\end{aligned}$$

This induction step with case (1) as the basis proves case (2) by induction on i.

(3) Let $i = n-1$, so $S = \{q_{n-1}, 0, s_2, s_3, \ldots, s_t\}$. Consider two cases of s_t.

$$\begin{aligned}
&\text{If } s_t \leq n-2, \text{ then} && \{q_{n-2}, 0, s_3 - s_2, \ldots, s_t - s_2\} \xrightarrow{ab^{s_2-1}} S.\\
&\text{If } s_t = n-1, \text{ then} && \{q_{n-2}, 0, s_3 - s_2, \ldots, s_{t-1} - s_2, n-2\} \xrightarrow{ab^{s_2-1}} S.
\end{aligned}$$

The starting set is reachable by induction on t in both cases.

(4) Let $i = 0$, so $S = \{q_0, s_1, s_2, \ldots, s_t\}$. We consider four cases of s_1 and s_t:

$$\begin{aligned}
&\text{If } s_1 = 0, s_t \leq n-2, \text{ then} && \{q_{n-1}, 0, n-1, s_3 - s_2, \ldots, s_t - s_2\} \xrightarrow{ab^{s_2-1}} S.\\
&\text{If } s_1 = 0, s_t = n-1, \text{ then} && \{q_{n-1}, 0, n-1, s_3 - s_2, \ldots, s_{t-1} - s_2, n-2\} \xrightarrow{ab^{s_2-1}} S.\\
&\text{If } s_1 \geq 1, s_t \leq n-2, \text{ then} && \{q_{n-1}, 0, s_2 - s_1, \ldots, s_t - s_1\} \xrightarrow{ab^{s_1-1}} S.\\
&\text{If } s_1 \geq 1, s_t = n-1, \text{ then} && \{q_{n-1}, 0, s_2 - s_1, \ldots, s_{t-1} - s_1, n-2\} \xrightarrow{ab^{s_1-1}} S.
\end{aligned}$$

The starting sets are considered in case (3).

This proves reachability. To prove distinguishability, notice that the word b^n is accepted by NFA N only from state $n-1$. It follows that $a^{n-1-t}b^n$ is accepted only from state t, $0 \leq t \leq n-1$. Hence two sets $\{q_i\} \cup S$ and $\{q_j\} \cup T$ are distinguishable if $S \neq T$. Consider two sets $\{q_i\} \cup S$, $\{q_j\} \cup S$ where $0 \leq i < j \leq n-1$ and assume that $\{i, j\} \not\subseteq S$. Let $i = 0$ and $S \subseteq \{0, 1, \ldots, n-1\}$. Then $j \notin S$ and we have

$$\{q_0\} \cup S \xrightarrow{a^{n-1-j}b^n} \{q_0, 0\} \xrightarrow{a} \{q_1, 0, 1\},$$

$$\{q_j\} \cup S \xrightarrow{a^{n-1-j}b^n} \{q_{n-1}, 0\} \xrightarrow{a} \{q_0, 1\},$$

where the resulting states differ in state 0. If $i \geq 1$, then we use a^{n-j} to get the case above.

Up to now, we reached all sets in $\mathcal{R}$ except for $\{q_0, n-1\}$. This set remains unreachable because of the inability to reach it by a nor b from other state. Hence $\mathrm{sc}(L^2(B)) = (n+3)2^{n-2} - 1$. To reach the set $\{q_0, n-1\}$, we add one more symbol to B. We define the transitions on the symbol c as follows:

$$\delta(q_0, c) = q_0; \quad \delta(q_i, c) = q_{i+i} \text{ if } 1 \leq i \leq n-2; \quad \delta(q_{n-1}, c) = q_0.$$

Denote the resulting DFA over $\{a, b, c\}$ by C. Then in the corresponding subset automaton for $L^2(C)$ the set $\{q_0, n-1\}$ is reachable from $\{q_0, n-2\}$ by c. Thus $\mathrm{sc}(L^2(C)) = (n+3)2^{n-2}$. □

As a corollary of the two lemmas above, we get the next result.

Corollary 10. *Let $n \geq 3$ and L be a language over Σ accepted by an n-state DFA in which $n-1$ states are final. Then $\mathrm{sc}(L^2) \leq (n+3)2^{n-2}$, and this bound is tight if $|\Sigma| \geq 3$. The bound $(n+3)2^{n-2} - 1$ is met by a binary language.* □

We tested the state complexity of square on all binary automata with 3, 4 and 5 states where the initial state is the only non-final state. But we did not find any binary automaton with the state complexity of its square greater than $(n+3)2^{n-2} - 1$. The following result shows that this lower bound is tight for every $n \geq 4$ on a binary alphabet.

Theorem 11. *Let $n \geq 4$ and L be a binary language accepted by an n-state DFA in which $n-1$ states are final. Then $\mathrm{sc}(L^2) \leq (n+3)2^{n-2} - 1$, and this bound is tight.*

Proof Idea. We already showed the witness language with $\mathrm{sc}(L^2) \geq (n+3)2^{n-2}-1$ in Lemma 9. It remains to show that the upper bound $(n+3)2^{n-2}$ cannot be met in binary case.

The reason of missing the upper bound by one was not reaching the $\{q_0, n-1\}$ in the subset automaton for the square in the first place. So to find a harder DFA for square than B in Fig. 4 we need to reach all possible distinguishable states. We found out that our desired automaton must have certain transitions to reach them. For example, transitions on a must form a permutation and transitions on b are exactly as in DFA B in Fig. 4. But these certain transitions plus our original restrictions for this case counteract our effort to distinguish these states. It follows that if some subset automaton for the square has $(n+3)2^{n-2}$ reachable states, many of them are equivalent. Thus the state complexity $(n+3)2^{n-2} - 1$ is the best that we can do. □

3.2 Square on Unary DFAs

To complete the overview about the square operation on deterministic automata we should not forget unary alphabets. We refer to the paper by Rampersad [8] once again. Notice that the complexity of square in this case is exponentially smaller than in the binary case.

Theorem 12 [8, Theorems 3 and 4 with $k = 2$]**.** *Let L be a unary language with $\mathrm{sc}(L) = n$. Then $\mathrm{sc}(L^2) \leq 2n - 1$ and the bound is tight.*

4 Square on AFAs and BFAs

Fellah et al. in [3, Theorem 9.3] showed that if a language K is accepted by an m-state AFA and a language L is accepted by an n-state AFA, then the language KL is accepted by an AFA of at most 2^m+n+1 states. It follows that 2^n+n+1 is an upper bound for the square on AFAs. Here we use our results from the previous section to prove tightness of this upper bound. For the square on BFAs, we get the tight upper bound 2^n+n. Recall that $\mathrm{asc}(L)$ is the smallest number of states in any AFA for L and $\mathrm{bsc}(L)$ is defined analogously.

Theorem 13 (Square on AFAs). *Let $n \geq 2$. Let L be a regular language over Σ with $\mathrm{asc}(L) = n$. Then $\mathrm{asc}(L^2) \leq 2^n+n+1$, and the bound is tight if $|\Sigma| \geq 2$.*

Proof. From given upper bound from [3, Theorem 9.3] we know that $\mathrm{asc}(L^2) \leq 2^n + n + 1$. For tightness, let L^R be the language accepted by the DFA A defined in the proof of Theorem 7 with 2^n states where half of the state are final, that is, $k = 2^{n-1}$. By Fact 3, L is accepted by AFA with n states. Using Theorem 7 we know that $\mathrm{sc}((L^R)^2) = 2^n 2^{2^n} - 2^{n-1}2^{2^n-1}$. By Corollary 2, $\mathrm{asc}(L^2) \geq \lceil \log(\mathrm{sc}((L^R)^2)) \rceil = 2^n + n$.

Suppose for a contradiction that L^2 is accepted by an AFA with $2^n + n$ states. By Fact 1, the language $(L^2)^R$ is accepted by a 2^{2^n+n}- state DFA with 2^{2^n+n-1} final states. It follows that the minimal DFA for $(L^2)^R$ has at most 2^{2^n+n-1} final states. However, the minimal DFA for the language $(L^2)^R$ has $2^n 2^{2^n} - 2^{n-1}2^{2^n-1} = 2^{n-1}2^{2^n} + 2^{n-1}2^{2^n-1}$ states, where $2^{n-1}2^{2^n-1} + 2^{n-1}2^{2^n-1}-1$ of them are non-final. Thus the number of final states in the minimal DFA for $(L^2)^R$ is $2^{n-1}(2^{2^n} + 2^{2^n-1}) - 2^{n-1}(2^{2^n-1} + 2^{2^n-1-1})$, and since $n \geq 2$, we get

$$
\begin{aligned}
&2^{n-1}(2^{2^n} + 2^{2^n-1}) - 2^{n-1}(2^{2^n-1} + 2^{2^n-1-1}) = \\
&2^{2^n}2^{n-1}(1 + \frac{1}{2} - \frac{1}{2^{2^n-1}} - \frac{1}{2^{2^n-1+1}}) > \\
&2^{2^n+n-1}(1 + \frac{1}{2} - \frac{1}{4} - \frac{1}{4}) = 2^{2^n+n-1}.
\end{aligned}
$$

Hence the minimal DFA for $(L^2)^R$ has more than 2^{2^n+n-1} final states, a contradiction. It follows that $\mathrm{asc}(L^2) \geq 2^n + n + 1$. □

Theorem 14 (Square on BFAs). *Let $n \geq 2$. Let L be a regular language over Σ with $\mathrm{bsc}(L) = n$. Then $\mathrm{bsc}(L^2) \leq 2^n + n$, and the bound is tight if $|\Sigma| \geq 2$.*

Proof. The upper bound follows from the upper bound 2^m+n on the complexity of the concatenation operation on BFAs [5, Theorem 4]. Let L^R be a language accepted by DFA A from Fig. 1 with 2^n states and one final state. By Fact 3, L is accepted by an n-state BFA. We are able to determine the state complexity of $(L^R)^2$ using Theorem 7: $\mathrm{sc}\big((L^R)^2\big) = 2^n \cdot 2^{2^n} - 2^{2^n-1}$. By Corollary 2,

$$\mathrm{bsc}(L^2) \geq \lceil \log\big(2^n \cdot 2^{2^n} - 2^{2^n-1}\big) \rceil = 2^n + n.$$

□

5 Conclusions

We studied the state complexity of the square of languages represented by deterministic, alternating, and Boolean finite automata. First, for each k such that $1 \leq k \leq n-2$, we showed that the upper bound $n2^n - k2^{n-1}$ on the square of languages represented by n-state DFAs with k final states is tight in the binary case. Then we analysed the case of $n-1$ final states, where we proved that the bound $(2n+2)2^{n-2}$ cannot be met. We provided the tight upper bound $(n+2)2^{n-2}$ for the case when the initial state is final and we found a binary witness. When the initial state is the only non-final state, we obtained the upper bound $(n+3)2^{n-2}$ with a ternary witness. In the binary case we proved that the tight upper bound is $(n+3)2^{n-2}-1$.

Finally, we used our results on the square on DFAs to describe binary witness languages meeting the upper bounds 2^n+n+1 and 2^n+n for square on alternating and Boolean finite automata, respectively. Our results can be extended for the concatenation operation just by concatenating two of our automata with different number of states. This provides an alternative solution for the open problem stated by Fellah et al. in [3].

References

1. Brzozowski, J.A., Leiss, E.L.: On equations for regular languages, finite automata, and sequential networks. Theor. Comput. Sci. **10**, 19–35 (1980). http://dx.doi.org/10.1016/0304-3975(80)90069-9
2. Čevorová, K., Jirásková, G., Krajňáková, I.: On the square of regular languages. In: Holzer, M., Kutrib, M. (eds.) CIAA 2014. LNCS, vol. 8587, pp. 136–147. Springer, Cham (2014). http://dx.doi.org/10.1007/978-3-319-08846-4_10
3. Fellah, A., Jürgensen, H., Yu, S.: Constructions for alternating finite automata. Int. J. Comput. Math. **35**(1–4), 117–132 (1990). http://dx.doi.org/10.1080/00207169008803893
4. Hospodár, M., Jirásková, G.: Concatenation on deterministic and alternating automata. In: Bordihn, H., Freund, R., Nagy, B., Vaszil, G. (eds.) NCMA 2016, vol. 321, pp. 179–194. Österreichische Computer Gesellschaft, books@ocg.at (2016)
5. Jirásková, G.: Descriptional complexity of operations on alternating and boolean automata. In: Hirsch, E.A., Karhumäki, J., Lepistö, A., Prilutskii, M. (eds.) CSR 2012. LNCS, vol. 7353, pp. 196–204. Springer, Heidelberg (2012). doi:10.1007/978-3-642-30642-6_19
6. Leiss, E.L.: Succint representation of regular languages by boolean automata. Theor. Comput. Sci. **13**, 323–330 (1981)
7. Maslov, A.N.: Estimates of the number of states of finite automata. Soviet Math. Doklady **11**, 1373–1375 (1970)
8. Rampersad, N.: The state complexity of L^2 and L^k. Inf. Process. Lett. **98**(6), 231–234 (2006). http://dx.doi.org/10.1016/j.ipl.2005.06.011
9. Sipser, M.: Introduction to the Theory of Computation. Cengage Learning, Boston (2012)
10. Yu, S., Zhuang, Q., Salomaa, K.: The state complexities of some basic operations on regular languages. Theor. Comput. Sci. **125**(2), 315–328 (1994). http://dx.doi.org/10.1016/0304-3975(92)00011-F