Theoretical Computer Science 578 (2015) 42–57

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

On the boundary of regular languages ☆

Jozef Jirásek [a,1], Galina Jirásková [b,*,2]

[a] Institute of Computer Science, Faculty of Science, P.J. Šafárik University, Jesenná 5, 040 01 Košice, Slovakia
[b] Mathematical Institute, Slovak Academy of Sciences, Grešákova 6, 040 01, Košice, Slovakia

ARTICLE INFO

Article history:
Received 28 October 2013
Received in revised form 25 June 2014
Accepted 14 January 2015
Available online 20 January 2015

Keywords:
Regular languages
Boundary
Finite automata
State complexity

ABSTRACT

We prove that the tight bound on the state complexity of the boundary of regular languages, defined as $\mathrm{bd}(L) = L^* \cap (\overline{L})^*$, is $3/8 \cdot 4^n + 2^{n-2} - 2 \cdot 3^{n-2} - n + 2$. Our witness languages are described over a five-letter alphabet. Next, we show that this bound cannot be met by any quaternary language if $n \geq 5$. However, the state complexity of boundary in the quaternary case is smaller by just one. Finally, we prove that the state complexity of boundary in the binary and ternary cases is $\Theta(4^n)$.

1. Introduction

The famous Kuratowski's "14-theorem" states that, in a topological space, at most 14 sets can be produced by applying the operations of closure and complement to a given set [2,5]. In analogy with this theorem, Brzozowski et al. [1] proved that there is only a finite number of distinct languages that arise from the operations of Kleene (or positive) closure and complement performed in any order and any number of times. Every such language can be expressed, up to inclusion of the empty string, as one of the following five languages and their complements: L, L^*, $(\overline{L})^*$, $(\overline{L^*})^*$, $((\overline{L})^*)^*$, where $\overline{L}$ and L^* denote the complement and Kleene closure of L, respectively. If the *state complexity* of a regular language L, that is, the number of states of the minimal deterministic finite automaton for L, is n, then the state complexity of $\overline{L}$ is also n, and the state complexity of L^* and $(\overline{L})^*$ is at most $3/4 \cdot 2^n$ [6,13]. The state complexity of $(\overline{L^*})^*$ could potentially be double-exponential [9], however, as shown in [3], it is only $2^{\Theta(n \log n)}$.

Brzozowski, Grant, and Shallit in [1] also studied the concepts of "open" and "closed" sets. A language L is said to be Kleene-closed if $L = L^*$, where L^* is the Kleene closure of L. A language is Kleene-open if its complement is Kleene-closed. The same notions can be defined for positive closure. These are natural analogues of the concepts with the same names from point-set topology, and in [1], the authors found many natural analogues of the classical theorems.

The boundary of a language is defined as $\mathrm{bd}(L) = L^* \cap (\overline{L})^*$, respectively, as $L^+ \cap (\overline{L})^+$ for positive closure [1,9,10]. In this paper, we study the state complexity of the boundary of regular languages in the case of Kleene closure. To simplify the

☆ This work was presented at the CIAA 2013 conference held in Halifax, Canada on July 16–19, 2013, and its extended abstract appeared in the conference proceedings [4].

* Corresponding author.
E-mail addresses: jozef.jirasek@upjs.sk (J. Jirásek), jiraskov@saske.sk (G. Jirásková).

[1] Supported by VEGA grant 1/0142/15.
[2] Supported by VEGA grant 2/0084/15.

http://dx.doi.org/10.1016/j.tcs.2015.01.022
0304-3975/

exposition, we will write everything in an exponent notation, using c to represent complement, thus L^{c*} stands for $(\overline{L})^*$, and so $\mathrm{bd}(L) = L^* \cap L^{c*}$.

We show that if a language L over an alphabet Σ is accepted by an n-state deterministic finite automaton (DFA), then the boundary $\mathrm{bd}(L)$ is accepted by a DFA of at most $3/8 \cdot 4^n + 2^{n-2} - 2 \cdot 3^{n-2} - n + 2$ states. We also show that this bound is tight in the case when the alphabet Σ has at least five symbols. Next, we show that if $n \geq 5$, then this bound cannot be met by any language defined over a four-letter alphabet, and that the tight bound in the quaternary case is $3/8 \cdot 4^n + 2^{n-2} - 2 \cdot 3^{n-2} - n + 1$. Finally, we prove that the state complexity of boundary in the binary and ternary cases is $\Theta(4^n)$. We also study the case when in a DFA for a language L, only the initial state is final. The upper bound for the boundary of L in such a case is $(n+2) \cdot 2^{n-2} + 1$, and we prove that this bound can be met by a binary language.

2. Preliminaries

In this section, we recall some basic definitions. For details and all unexplained notions, the reader may refer to [8,11,12].

For integers i and j with $i \leq j$, we denote by $[i, j]$ the set of integers $\{k \mid i \leq k \leq j\}$. The cardinality of a finite set A is denoted by $|A|$, and its power-set by 2^A.

Let Σ be a finite non-empty alphabet. Then Σ^* denotes the set of all strings over the alphabet Σ, including the empty string ε. A language over the alphabet Σ is any subset of Σ^*. Let K and L be languages over an alphabet Σ. Then $L^c = \Sigma^* \setminus L$, $K \cap L = \{w \in \Sigma^* \mid w \in K \text{ and } w \in L\}$, $KL = \{uv \mid u \in K \text{ and } v \in L\}$, and $L^* = \bigcup_{i \geq 0} L^i$, where $L^0 = \{\varepsilon\}$ and $L^{i+1} = L^i L$. The *boundary* of a language L is the set $\mathrm{bd}(L) = L^* \cap L^{c*}$, where we use L^{c*} to denote $(L^c)^*$.

A *nondeterministic finite automaton* (NFA) is a quintuple $A = (Q, \Sigma, \cdot, s, F)$, where Q is a finite non-empty set of states, Σ is a finite alphabet, $\cdot : Q \times \Sigma \to 2^Q$ is the transition function which is extended to the domain $2^Q \times \Sigma^*$ in the natural way, $s \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states. The language accepted by A is the set $L(A) = \{w \in \Sigma^* \mid s \cdot w \cap F \neq \emptyset\}$.

An NFA $A = (Q, \Sigma, \cdot, s, F)$ is *deterministic* (and complete) (DFA) if $|q \cdot a| = 1$ for each q in Q and each a in Σ. In such a case, we write $q \cdot a = q'$ instead of $q \cdot a = \{q'\}$. A state q of the DFA A is reachable if there exists a string w in Σ^* such that $s \cdot w = q$. Two states p and q are distinguishable if there exists a string w such that exactly one of the states $p \cdot w$ and $q \cdot w$ is final. Two states are equivalent if they are not distinguishable.

The *state complexity* of a regular language L, $\mathrm{sc}(L)$, is the smallest number of states in any DFA recognizing L. It is well known that a DFA is minimal (with respect to the number of states) if all its states are reachable, and no two distinct states are equivalent.

Every symbol a of the DFA A may be viewed as a transformation on the set Q, that is, as a mapping from Q to Q. A symbol a is called a *permutation* symbol if a performs a permutation on Q.

The *symmetric* group is the group of all permutations on the set $\{0, 1, \ldots, n-1\}$. The symmetric group is generated by a circular shift that maps i to $(i+1) \bmod n$, and by a swap permutation that swaps 0 and 1 and maps any other i to itself.

Every NFA $A = (Q, \Sigma, \cdot, s, F)$ can be converted to an equivalent DFA $A' = (2^Q, \Sigma, \cdot', \{s\}, F')$, where $R \cdot' a = R \cdot a$ and $F' = \{R \in 2^Q \mid R \cap F \neq \emptyset\}$ by the subset construction [7]. The DFA A' is called the *subset automaton* of the NFA A. The subset automaton need not be minimal since some of its states may be unreachable or equivalent.

Let $A = (Q_A, \Sigma, s_A, \cdot_A, F_A)$ and $B = (Q_B, \Sigma, s_B, \cdot_B, F_B)$ be two DFAs. Then $L(A) \cap L(B)$ is recognized by the *product automaton* $A \times B = (Q_A \times Q_B, \Sigma, \cdot, (s_A, s_B), F_A \times F_B)$, where $(p, q) \cdot a = (p \cdot_A a, q \cdot_B a)$.

3. Upper bound: construction of DFAs for boundary

The boundary of a regular language L is defined by $\mathrm{bd}(L) = L^* \cap L^{c*}$, where $L^{c*} = (L^c)^*$. Since the state complexity of star is $3/4 \cdot 2^n$ [6,13], the trivial upper bound on the state complexity of boundary is $9/16 \cdot 4^n$. The aim of this section is to get a slightly better upper bound $3/8 \cdot 4^n + 2^{n-2} - 2 \cdot 3^{n-2} - n + 2$.

We start with the construction of a DFA for $L^* \cap L^{c*}$. Without loss of generality, we may assume that the empty string is in L. Let a language L be accepted by a DFA $A = (Q, \Sigma, \cdot, s, F)$, where $|Q| = n$, $s \in F$, $|F| = k$, and $\cdot$ is the transition function extended to the domain $2^Q \times \Sigma^*$ in a natural way. Let $F^c = Q \setminus F$.

Construct an NFA N for the language L^* from the DFA A as follows. For each state q and each symbol a, if $q \cdot a \in F$, then add the transition from q to s on a. Next, construct an NFA N' for the language L^{c*} from the DFA A as follows. First, interchange the sets of final and non-final states to get a DFA for L^c. Then, add a transition from a state q to the state s on a symbol a whenever $q \cdot a \in F^c$. Finally, add a new initial and final state q_0 that goes on each symbol a to $\{s \cdot a\}$ if $s \cdot a \notin F^c$, and to $\{s, s \cdot a\}$ if $s \cdot a \in F^c$. Fig. 1 illustrates the construction of NFAs N and N'. Denote the transition functions of the NFAs N and N', extended to the domain $2^Q \times \Sigma^*$ in a usual way, by $\circ$ and $\bullet$, respectively.

Let D and D' be the subset automata of the NFAs N and N', respectively. Then the language $L^* \cap L^{c*}$ is accepted by the product automaton $D \times D'$, the states of which are pairs of subsets of Q. The initial state of the product automaton is the pair $(\{s\}, \{q_0\})$, and a pair (S, T) is final if S is a final state in D and T is a final state in D', that is, if $S \cap F \neq \emptyset$ and $T \cap F^c \neq \emptyset$. Each pair (S, T) goes to the pair $(S \circ a, T \bullet a)$ in the product automaton. Notice that

$$S \circ a = \begin{cases} S \cdot a, & \text{if } S \cdot a \cap F = \emptyset, \\ S \cdot a \cup \{s\}, & \text{otherwise,} \end{cases} \qquad T \bullet a = \begin{cases} T \cdot a, & \text{if } T \cdot a \cap F^c = \emptyset, \\ T \cdot a \cup \{s\}, & \text{otherwise.} \end{cases}$$