Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

State complexity of prefix distance ☆

Timothy Ng *, David Rappaport, Kai Salomaa

School of Computing, Queen's University, Kingston, Ontario K7L 2N8, Canada

ARTICLE INFO

Article history:
Received 27 November 2015
Received in revised form 2 April 2016
Accepted 11 May 2016
Available online xxxx

Keywords:
Regular languages
State complexity
Prefix distance
Finite automata

ABSTRACT

The prefix distance between strings x and y is the number of symbol occurrences in the strings that do not belong to the longest common prefix of x and y. The suffix and the substring distances are defined analogously in terms of the longest common suffix and longest common substring, respectively, of two strings. We show that the set of strings within prefix distance k from an n state DFA (deterministic finite automaton) language can be recognized by a DFA with $(k+1)\cdot n - \frac{k(k+1)}{2}$ states and that this number of states is needed in the worst case. Also we give tight bounds for the nondeterministic state complexity of the set of strings within prefix, suffix or substring distance k from a regular language.

1. Introduction

Various similarity measures between strings and languages have been considered for information transmission applications. The edit distance counts the number of substitution, insertion and deletion operations that are needed to transform one string to another. The Hamming distance counts the number of positions in which two equal length strings differ. A distance measure between words can be extended in various ways as a distance between sets of strings (or languages) [3,4] and algorithms for computing the distance between languages are important for error-detection and error-correction applications [4,9,10]. The descriptional complexity of error/edit systems has been considered by Kari and Konstantinidis [8]. Other types of sequence similarity measures have been considered e.g. by Apostolico [1].

Instead of counting the number of edit operations, the similarity of strings can be defined by way of their longest common prefix, suffix, or substring, respectively [4]. For example, the prefix distance of strings x and y is the sum of the length of the suffix of x and the suffix of y that occurs after their longest common prefix. A parameterized prefix distance between regular languages has been considered by Kutrib et al. [11] for estimating the fault tolerance of information transmission applications.

The neighbourhood of radius k of a language L consists of all strings that are within distance k from some string in L. Calude et al. [3] have shown that the neighbourhood of a regular language with respect to an additive distance is regular. A distance is said to be additive if it, in a certain sense, respects string concatenation. This gives rise to the question how large is the (non)deterministic finite automaton (DFA, respectively, NFA) needed to recognize the neighbourhood of a regular language, that is, what is the state complexity of neighbourhoods of regular languages.

☆ An extended abstract of this paper appeared in the Proceedings of the 20th International Conference Implementation and Application of Automata, Umeå, Sweden, August 18–21, 2015.

* Corresponding author.
E-mail addresses: ng@cs.queensu.ca (T. Ng), daver@cs.queensu.ca (D. Rappaport), ksalomaa@cs.queensu.ca (K. Salomaa).

http://dx.doi.org/10.1016/j.tcs.2016.05.018
0304-3975/

Povarov [16] has given an improved upper bound and a closely matching lower bound for the state complexity of Hamming neighbourhoods of radius one. Upper bounds for the state complexity of neighbourhoods with respect to an additive distance or quasi-distance have been obtained by the authors [14,17] using a construction based on weighted finite automata and a matching lower bound was given recently in [15].

It follows from Choffrut and Pighizzini [4] that the prefix, suffix and substring distances preserve regularity, that is, the neighbourhood of a regular language of finite radius remains regular. Here we study the state complexity of these neighbourhoods. The neighbourhood of radius r of a language L with respect to the prefix distance, roughly speaking, consists of strings that share a "long" prefix with a string $u \in L$, more precisely, it is required that the combined length of the parts of w and u outside their longest common suffix is at most the constant r. In view of this it seems reasonable to expect that the state complexity of prefix distance neighbourhoods does not incur a similar exponential size blow-up as the edit distance [15].

We show that if L is recognized by a deterministic finite automaton (DFA) of size n, the prefix neighbourhood of L of radius $k < n$ has a DFA of size $(k+1) \cdot n - \frac{k(k+1)}{2}$ and that this bound cannot be improved in the worst case. Our lower bound construction uses an alphabet of size $n-1$ and we show that the general upper bound cannot be reached using languages defined over a fixed alphabet.

We consider also the nondeterministic state complexity of prefix, suffix and substring neighbourhoods. If L has a nondeterministic finite automaton (NFA) of size n, the neighbourhood of L of radius k can be recognized by an NFA of size $n+k$. The upper bound for the substring neighbourhood of L of radius k is $(k+1) \cdot n + 2k$. In all cases we give matching lower bounds for nondeterministic state complexity, and in the lower bound constructions L has, in fact, a DFA of size n.

2. Preliminaries

Here we briefly recall some definitions and notation used in the paper. For all unexplained notions on finite automata and regular languages the reader may consult the textbook by Shallit [18] or the survey by Yu [19]. A survey of distances is given by Deza and Deza [5]. Recent surveys on descriptional complexity of regular languages include [6,7,12].

In the following Σ is always a finite alphabet, the set of strings over Σ is Σ^* and ε is the empty string. The reversal of a string $x \in \Sigma^*$ is x^R. The set of nonnegative integers is $\mathbb{N}_0$. The cardinality of a finite set S is denoted $|S|$ and the powerset of S is 2^S. A string $w \in \Sigma^*$ is a *substring* or *factor* of x if there exist strings $u, v \in \Sigma^*$ such that $x = uwv$. If $u = \varepsilon$, then w is a *prefix* of x. If $v = \varepsilon$, then w is a *suffix* of x.

A *nondeterministic finite automaton* (NFA) is a 5-tuple $A = (Q, \Sigma, \delta, Q_0, F)$ where Q is a finite set of states, Σ is an alphabet, δ is a multi-valued transition function $\delta : Q \times \Sigma \to 2^Q$, $Q_0 \subseteq Q$ is a set of initial states, and $F \subseteq Q$ is a set of final states. We extend the transition function δ to $Q \times \Sigma^* \to 2^Q$ in the usual way. A string $w \in \Sigma^*$ is *accepted* by A if, for some $q_0 \in Q_0$, $\delta(q_0, w) \cap F \neq \emptyset$ and the language recognized by A consists of all strings accepted by A. An ε-NFA is an extension of an NFA where transitions can be labeled by the empty string ε [18,19], i.e., δ is a function $Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$. It is known that every ε-NFA has an equivalent NFA without ε-transitions and with the same number of states. An NFA $A = (Q, \Sigma, \delta, Q_0, F)$ is a *deterministic finite automaton* (DFA) if $|Q_0| = 1$ and, for all $q \in Q$ and $a \in \Sigma$, $\delta(q, a)$ either consists of one state or is undefined. Two states p and q of a DFA A are equivalent if $\delta(p, w) \in F$ if and only if $\delta(q, w) \in F$ for every string $w \in \Sigma^*$. A DFA A is *minimal* if each state $q \in Q$ is reachable from the initial state and no two states are equivalent.

Note that our definition of a DFA allows some transitions to be undefined, that is, by a DFA we mean an incomplete DFA. It is well known that, for a regular language L, the sizes of the minimal incomplete and complete DFAs differ by at most one. The constructions in Section 3 are more convenient to formulate using incomplete DFAs but our results would not change in any significant way if we were to require that all DFAs are complete.

The (incomplete deterministic) *state complexity* of a regular language L, $\mathrm{sc}(L)$, is the size of the minimal DFA recognizing L. The *nondeterministic state complexity* of L, $\mathrm{nsc}(L)$, is the size of a minimal NFA recognizing L. A minimal NFA recognizing a regular language need not be unique. A common way of establishing lower bounds for nondeterministic state complexity relies on fooling sets.

Definition 1. A set of pairs of strings $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, $x_i, y_i \in \Sigma^*$, $i = 1, \ldots, m$, is a *fooling set* for a language L if $x_i y_i \in L$, $i = 1, \ldots, m$ and, for all $1 \leq i < j \leq m$, $x_i y_j \notin L$ or $x_j y_i \notin L$.

Proposition 1 *([2,7]). If L has a fooling set S then* $\mathrm{nsc}(L) \geq |S|$.

To conclude this section, we recall definitions of the distance measures used in the following. Generally, a function $d : \Sigma^* \times \Sigma^* \to [0, \infty)$ is a *distance* if it satisfies for all $x, y, z \in \Sigma^*$, the conditions $d(x, y) = 0$ if and only if $x = y$, $d(x, y) = d(y, x)$, and $d(x, z) \leq d(x, y) + d(y, z)$. The *neighbourhood* of a language L of radius k with respect to a distance d is the set

$$E(L, d, k) = \{w \in \Sigma^* \mid (\exists x \in L) d(w, x) \leq k\}.$$

Let $x, y \in \Sigma^*$. The *prefix distance* of x and y counts the number of symbols which do not belong to the longest common prefix of x and y [4]. It is defined by

$$d_p(x, y) = |x| + |y| - 2 \cdot \max_{z \in \Sigma^*} \{|z| \mid x, y \in z\Sigma^*\}.$$