Theoretical Computer Science 610 (2016) 2–12

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

State complexity of inversion operations [☆]

Da-Jung Cho [a], Yo-Sub Han [a,*], Sang-Ki Ko [a], Kai Salomaa [b]

[a] Department of Computer Science, Yonsei University, Seoul 120-749, Republic of Korea
[b] School of Computing, Queen's University, Kingston, Ontario K7L 3N6, Canada

A R T I C L E   I N F O

Article history:
Received 31 October 2014
Received in revised form 25 March 2015
Accepted 14 April 2015
Available online 17 April 2015

Keywords:
State complexity
Inversion operations
Finite automata
Regular languages

A B S T R A C T

The reversal operation is well-studied in the literature and the deterministic (respectively, nondeterministic) state complexity of reversal is known to be 2^n (respectively, n). We consider the inversion operation where some substring of the given string is reversed. Formally, the inversion (respectively, prefix-inversion) of a language L consists of all strings $ux^R v$ such that $uxv \in L$ (respectively, all strings $u^R x$ where $ux \in L$). We show that the nondeterministic state complexity of prefix-inversion is $\Theta(n^2)$ and that of inversion is $\Theta(n^3)$. We show that the deterministic state complexity of prefix-inversion is at most $2^{n \cdot \log n + n}$ and has lower bound $2^{\Omega(n \log n)}$. The same lower bound holds for the state complexity of inversion, but for inversion we do not have a matching upper bound. We also study the state complexity of other variants of the inversion operation.

1. Introduction

Questions of descriptional complexity belong to the very foundations of automata and formal language theory [10,14,26,30]. The state complexity of finite automata has been studied since the 60s [16,19,20]. Maslov [18] originated the study of operational state complexity and Yu et al. [30] presented complete proofs for the state complexity of basic operations. Later, Yu and his co-authors [7,8,23,24] initiated the study on the state complexity of combined operations such as star-of-union, star-of-intersection and so on.

In biology, a *chromosomal inversion* occurs when a segment of a single chromosome breaks and rearranges within itself in reverse order [21]. It is known that the chromosomal inversion often causes genetic diseases [17]. Informally, the inversion operation reverses an infix of a given string. This can be viewed as a generalization of the reversal operation which reverses the whole string. The inversion of a language L is defined as the union of all inversions of strings in L. Therefore, the inversion of L always contains the reversal of L since a string is always an infix of itself.

Many researchers [2,4–6,13,27] have considered the inversion of DNA sequences in terms of formal language theory. Searls [25] considered closure properties of languages under various bio-inspired operations including inversion. Later, Yokomori and Kobayashi [29] showed that inversion can be simulated by the set of primitive operations and languages. Dassow et al. [5] investigated a generative mechanism based on some operations inspired by mutations in genomes such as deletion, transposition, duplication and inversion. Daley et al. [4] considered a hairpin inversion operation, which replaces the

☆ A preliminary version appeared in *Proceedings of the 16th International Workshop on Descriptional Complexity of Formal Systems*, DCFS 2014, LNCS **8614**, 102–113, Springer-Verlag, 2014.

* Corresponding author.

E-mail addresses: dajung@cs.yonsei.ac.kr (D.-J. Cho), emmous@cs.yonsei.ac.kr (Y.-S. Han), narame7@cs.yonsei.ac.kr (S.-K. Ko), ksalomaa@cs.queensu.ca (K. Salomaa).

http://dx.doi.org/10.1016/j.tcs.2015.04.017
0304-3975/© 2015 Elsevier B.V. All rights reserved.

hairpin part of a string with the inversion of the hairpin part. Note that the hairpin inversion operation is a variation of the inversion operation that reverses substrings of a string. Recently, Cho et al. [3] defined the pseudo-inversion operation and examined closure properties and decidability problems regarding the operation. Strong decidability and undecidability results of related operations were given by Ibarra [11]. Moreover, several string matching problems allowing inversions have been studied [2,27].

From a descriptional complexity point of view, reversal is an "easy" operation for NFAs. The reversal of a regular language L can be, roughly speaking, recognized by an NFA that is obtained by reversing the transitions of an NFA for L and, consequently, the nondeterministic state complexity of the reversal operation is n for NFAs that allow multiple initial states [9].[1] However, a corresponding simple NFA construction does not work for inversion and here we show that the nondeterministic state complexity of inversion is $\Theta(n^3)$. The prefix- (respectively, suffix-) inversion operations reverses a prefix (respectively, a suffix) of a given string. We show that the nondeterministic state complexity of prefix- and suffix-inversion is $\Theta(n^2)$. Moreover, we establish the nondeterministic state complexity of the pseudo-inversion, which is defined as the reversal of inversion, and the prefix-pseudo- and suffix-pseudo-inversion operations.

It is known that the deterministic state complexity of the reversal operation is 2^n [22]. The inversion operation is, in some sense, an extension of the reversal operation and using this correspondence it is easy to see that the state complexity of inversion is at least exponential. We give an upper bound $2^{n\cdot\log n+n}$ for the deterministic state complexity of prefix-inversion and an almost matching lower bound $2^{(n-3)\cdot\log(n-3)}$. The lower bound uses an alphabet depending on n.

The nondeterministic state complexity of inversion implies an upper bound 2^{n^3+2n} for the corresponding deterministic state complexity. Using a direct construction we improve this bound to $n\cdot 2^{n^3+n}$. Also the construction used for prefix-inversion yields the same lower bound $2^{(n-3)\cdot\log(n-3)}$ for the deterministic state complexity of inversion, however, a large gap remains with the upper bound.

With the exception of prefix-inversion, the precise deterministic state complexity of inversion and different variants of the operation remains open.

We give the basic notations and definitions in Section 2. We introduce the inversion and related operations in Section 3 and present the results for nondeterministic state complexity in Section 4 and the deterministic state complexity results in Section 5. In Section 6, we conclude the paper.

2. Preliminaries

We briefly present definitions and notations used throughout the paper. The reader may refer to the textbooks [26,28] or the handbook [22] for more details on formal language theory.

The cardinality of a finite set S is denoted $|S|$. When there is no danger of confusion, a singleton set $\{x\}$ is denoted simply by x. The set of functions from S to itself is S^S and the composition of functions $f, g \in S^S$ is $f \circ g$, where $(f \circ g)(x) = f(g(x))$, for $x \in S$.

Let Σ be a finite alphabet and Σ^* be a set of all strings over Σ. A language over Σ is any subset of Σ^*. The symbol λ denotes the null string and Σ^+ denotes $\Sigma^* \setminus \{\lambda\}$. Given a string $w = w_1 w_2 \cdots w_m$, $w_i \in \Sigma$, $1 \le i \le m$, we denote the *reversal* of w by $w^R = w_m w_{m-1} \cdots w_1$. Note that $\lambda^R = \lambda$.

A *nondeterministic finite automaton with λ-transitions* (λ-NFA) is a five-tuple $A = (\Sigma, Q, Q_0, F, \delta)$ where Σ is a finite alphabet, Q is a finite set of states, $Q_0 \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of final states and δ is a multi-valued transition function from $Q \times (\Sigma \cup \{\lambda\})$ into 2^Q. By an NFA we mean a nondeterministic automaton without λ-transitions, that is, A is an NFA if δ is a function from $Q \times \Sigma$ into 2^Q. The automaton A is *deterministic* (a DFA) if Q_0 is a singleton set and δ is a (total single-valued) function $Q \times \Sigma \to Q$. It is well known that the λ-NFAs, NFAs and DFAs all recognize the regular languages [22,26,28].

Proposition 2.1. *(See [28].) A λ-NFA has an equivalent NFA without λ-transitions and the same number of states.*

The (right) *Kleene congruence* of a language $L \subseteq \Sigma^*$ is the relation $\equiv_L \subseteq \Sigma^* \times \Sigma^*$ defined by setting, for $x, y \in \Sigma^*$,

$$x \equiv_L y \text{ iff } [(\forall z \in \Sigma^*)\, xz \in L \Leftrightarrow yz \in L].$$

It is well known that L is regular if and only if the index of $\equiv_L$ is finite and, in this case, the number of classes of $\equiv_L$ is equal to the size of the minimal DFA for L [22,26,28].

The deterministic (respectively, nondeterministic) state complexity of a regular language L, $\mathrm{sc}(L)$ (respectively, $\mathrm{nsc}(L)$) is the size of the minimal DFA (respectively, the size of a minimal NFA) recognizing L. Thus, $\mathrm{sc}(L)$ is equal to the number of classes of $\equiv_L$.

The nondeterministic state complexity of a language can be estimated using the *fooling set technique* that gives a lower bound for the size of NFAs [1].

[1] The result stated in [9] is $n+1$ because the NFA model used there allows only one initial state.