On state complexity for subword-closed languages

Jérôme Guyot[1]

DER Informatique, Univ. Paris-Saclay, ENS Paris-Saclay, Gif-sur-Yvette, France
jerome.guyot@ens-paris-saclay.fr

Abstract. This paper investigates the state complexities of subword-closed and superword-closed languages, comparing them to regular languages. We focus on the square root operator and the substitution operator. We establish an exponential lower bound for superword-closed languages for the n-th root. For subword-closed languages we analyze in detail a specific instance of the square root problem for which a quadratic complexity is proven. For the substitution operator, we show an exponential lower bound for the general substitution. We then find some conditions for which we prove a quadratic upper bound.

Introduction

State complexity. The number of states of the canonical automaton recognizing a regular language L is known as its state complexity, denoted $\kappa(L)$. It is a common measure of the complexity of regular languages[6]. Finite state automaton are often used as data structure: the size of the automata thus becomes an important parameter in the complexity analysis of some algorithms.

For an operation or a function f on regular languages, the natural question would be what is the state complexity of $f(L)$ when L has state complexity n ? This leads to the definition of *the state complexity of* f as the function $\phi_f : \mathbb{N} \to \mathbb{N}$ such that $\phi_f(n)$ is the maximum state complexity of $f(L)$ with L having state complexity at most n. This notion can be extended to functions having multiple arguments, for example the state complexity of intersection would be given by $\phi_\cap(n_1, n_2)$. State complexity of regular languages already has a rich literature and the recent survey [12] describes the known results for a wide range of operations and classes of languages. As it can be difficult to find the exact complexity of some $f(L)$ or to give a formula for the function ϕ_f, the goal is often to obtain bounds on the complexity of $f(L)$ and on ϕ_f. This induces a classification of the operations on regular languages and finite automata based on the growth of ϕ_f.

State complexity of subregular classes. It is often interesting to measure the state complexity of a function f when we restrict its argument to a subregular class. As some applications only focus on a subregular class of automata it becomes natural to study the state complexity on this restricted domain. For example, computational linguistic uses automata to encode lexicons that are always finite languages: they are considered in [10] while their complement, cofinite languages, are considered in [3]. Linguistics

is also interested in locally-testable languages [16], and other areas like genomics or databases or text processing have their own subclasses of interest.

The study of state complexity restricted to such subclasses has recently become quite active after Brzozowski et al. initiated a systematic study of state complexity on various fundamental classes of subregular languages [9,5,7,8].

Subword-closed and superword-closed languages. In the context of computer-aided verification, several algorithms for program verification uses well-quasi-ordered data domains [11,1,4] and in particular, using Higman's lemma, words ordered by the subword order. These algorithms have to compute with subword-closed and superword-closed languages.

Superword and subword closed languages have other terminologies that depend on the characterization. We follow [20]. Some authors use "subword" for a factor, and use "scattered subword" for what we call a subword. In [8] superword-closed languages are called all-sided ideals when seeing them as shuffle ideals. The state complexity of superword and subword closed languages has not been analyzed extensively, a more studied problem is to obtain the subword and superword closure of a language and study its state complexity [15,13,14,19,17].

Brzozowski et al. considered subword-closed languages in [7] and superword-closed languages in [8]: they only consider the most usual operations: boolean combinations, concatenation, iteration and mirror. However, there exist other interesting operations to consider as they also preserve the subword/superword closedness such as the shuffle.

Our contribution. We are interested in completing the picture and consider the state complexity on other operations on subword-closed or superword-closed languages.

In the following sections, we focus on two operators the n^{th}-root operators and the substitution operator. For the root operators, we show that they have exponential complexity even when restricted to superword-closed languages. For the subword closed languages, when L is the language of subwords of a word w it seems that there is a quadratic upper bound, we do not know if it extends in the general case of subword closed languages. For the substitution operator, we show an exponential lower bound for the general substitution. In the case where only one letter is substituted, we show a quadratic upper bound when L and K are subword closed languages and based on disjoint alphabets and conjecture a quadratic upper bound when L and K are subword closed languages. Finally we proved a quadratic upper bound when L is directed and K is downward closed (without any hypothesis on the alphabet).

This work contributes to understanding the state complexities of subregular languages. It was done in the context of an initiation to research project at ENS Paris-Saclay. I warmly thank Philippe Schnoebelen for his valuable help and dedication to the project. I also thank Maelle Gautrin and Simon Corbard for initiating the research on the substitution operator.

1 Main results

We say that a word x is a *subword* of y, written $x \preccurlyeq y$, when it is a subsequence. For example $\texttt{JMIT} \preccurlyeq \texttt{JUMPIEST}$. For a language $L \subseteq \Sigma^*$, we let $\downarrow(L) = \{x \in \Sigma^* \mid \exists\ y \in L,\ x \preccurlyeq y\}$ denote the downward closure of L. L is *downward-closed* if and only if $\downarrow(L) = L$. We further let $\uparrow(L) = \{x \in \Sigma^* \mid \exists\ y \in L,\ y \preccurlyeq x\}$ denote the upward closure of L. L is *upward-closed* if and only if $\uparrow(L) = L$.

***k*-th root.** For $k \in \mathbb{N}$ and a language L, the k^{th} root of L is the set $\sqrt[k]{L} = \{x \mid x^k \in L\}$. This operation can be seen as the inverse of concatenation k times. It is well known that $\sqrt[k]{L}$ is regular when L is [18]. Further note that $\sqrt[k]{L}$ is upward-closed or downward-closed when L is.

Theorem 1. *For any $k \geq 2$, the state complexity of the k^{th} root operator is exponential even when restricted to upward-closed languages.*

For downward closed languages, we still do not know if k^{th} root has exponential state complexity. We describe an example showing why the situation is more complex than for upward closed.

Substitution. We write $L^{a\leftarrow K, b\leftarrow K', \cdots}$ for the result of substituting the languages $K, K', \ldots$ for every occurrence of $a, b, \ldots$ in any word of L. It is well known that substitutions preserve regularity. Observe that $L^{a\leftarrow K, b\leftarrow K', \cdots}$ is downward-closed when $L, K, K', \ldots$ are. For example take $L = \{\texttt{aa}, \texttt{ba}\}$ and $K = \{\texttt{c}, \texttt{bc}\}$ and then

$$L^{a\leftarrow K, b\leftarrow L} = \{\texttt{cc}, \texttt{bcc}, \texttt{cbc}, \texttt{bcbc}, \texttt{aac}, \texttt{aabc}, \texttt{bac}, \texttt{babc}\}$$

Theorem 2. *The state complexity of substitution is exponential even when restricted to downward-closed languages.*

In the case $L^{a\leftarrow K}$ where only one letter is substituted, we do not know if state complexity remains exponential. However, under some additional conditions we can prove a quadratic upper bound.

Theorem 3. *Let L, K be downward closed languages based on disjoints alphabets. Then $\kappa(L^{a\leftarrow K}) \leq \kappa(L)\kappa(K)$.*

Definition 1 (SREs,[2]).

An atom is a (particular case of) regular expression α that is either a letter-atom $a + \epsilon$ with $a \in \Sigma$, or a star-atom B^* with $B \subseteq \Sigma$.

A product of atoms (or product) I is a finite concatenation of atoms : $I = \prod_{1\leq i\leq n} \alpha_i$ where α_i is an atom. It is a regular expression and the empty product denotes ϵ.

An SRE E is a finite sum of products : $E = \sum_{1\leq j\leq m} I_j$. The empty sum denotes $\emptyset$. We denote by $[\![E]\!]$ the language described by E. The SREs form a subclass of regular expressions.

Theorem 4 ([2]). *A language L on a finite alphabet Σ is downward closed if and only if it can be defined by an SRE.*

Theorem 5. *Let I be a product of atoms and K a downward closed language, then $\kappa(I^{a\leftarrow K}) \leq \kappa(K)\kappa(I)$.*

References

1. Abdulla, P.A., Cerāns, K., Jonsson, B., Tsay, Y.K.: Algorithmic analysis of programs with well quasi-ordered domains. Information & Computation **160**, 109–127 (2000)
2. Abdulla, P.A., Collomb-Annichini, A., Bouajjani, A., Jonsson, B.: Using forward reachability analysis for verification of lossy channel systems. Formal Methods in System Design **25**(1), 39–65 (2004)
3. Bassino, F., Giambruno, L., Nicaud, C.: Complexity of operations on cofinite languages. In: Proc. 9th Latin American Symp. Theoretical Informatics (LATIN 2010). Lecture Notes in Computer Science, vol. 6034, pp. 222–233. Springer (2010)
4. Bertrand, N., Schnoebelen, Ph.: Computable fixpoints in well-structured symbolic model checking. Formal Methods in System Design **43**(2), 233–267 (2013)
5. Brzozowski, J., Li, B.: Syntactic complexity of R- and J-trivial regular languages. Int. J. Foundations of Computer Science **25**(07), 807–821 (2014)
6. Brzozowski, J.: Quotient complexity of regular languages. Journal of Automata, Languages and Combinatorics **15**(1–2), 71–89 (2010)
7. Brzozowski, J., Jirásková, G., Zou, C.: Quotient complexity of closed languages. Theory of Computing Systems **54**(2), 277–292 (2014)
8. Brzozowski, J., Jirásková, G., Li, B.: Quotient complexity of ideal languages. Theoretical Computer Science **470**, 36–52 (2013)
9. Brzozowski, J.A., Li, B., Liu, D.: Syntactic complexities of six classes of star-free languages. Journal of Automata, Languages and Combinatorics **17**(2–4), 83–105 (2012)
10. Câmpeanu, C., Culik, K., Salomaa, K., Yu, S.: State complexity of basic operations on finite languages. In: Proc. 4th Int. Workshop Implementing Automata (WIA '99). Lecture Notes in Computer Science, vol. 2214, pp. 60–70. Springer (2001)
11. Finkel, A., Schnoebelen, Ph.: Well-structured transition systems everywhere! Theoretical Computer Science **256**(1–2), 63–92 (2001)
12. Gao, Y., Moreira, N., Reis, R., Yu, S.: A survey on operational state complexity. Journal of Automata, Languages and Combinatorics **21**(4), 251–310 (2017)
13. Gruber, H., Holzer, M., Kutrib, M.: The size of Higman-Haines sets. Theoretical Computer Science **387**(2), 167–176 (2007)
14. Gruber, H., Holzer, M., Kutrib, M.: More on the size of Higman-Haines sets: Effective constructions. Fundamenta Informaticae **91**(1), 105–121 (2009)
15. Héam, P.C.: On shuffle ideals. RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications **36**(4), 359–384 (2002)
16. Heinz, J.: On the role of locality in learning stress patterns. Phonology **26**(2), 303–351 (2009)
17. Karandikar, P., Niewerth, M., Schnoebelen, Ph.: On the state complexity of closures and interiors of regular languages with subwords and superwords. Theoretical Computer Science **610**, 91–107 (2016)
18. Krawetz, B., Lawrence, J., Shallit, J.: State complexity and the monoid of transformations of a finite set. Int. J. Foundations of Computer Science **16**(3), 547–563 (2005)
19. Okhotin, A.: On the state complexity of scattered substrings and superstrings. Fundamenta Informaticae **99**(3), 325–338 (2010)
20. Sakarovitch, J., Simon, I.: Subwords. In: Lothaire, M. (ed.) Combinatorics on Words, Encyclopedia of Mathematics and Its Applications, vol. 17, chap. 6, pp. 105–142. Cambridge Univ. Press (1983)