Theoretical Computer Science 610 (2016) 91–107

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

On the state complexity of closures and interiors of regular languages with subwords and superwords

P. Karandikar [a,c,*,1], M. Niewerth [b,*,2], Ph. Schnoebelen [c,*,3]

[a] Chennai Mathematical Institute, France
[b] University of Bayreuth, Germany
[c] LSV, ENS Cachan, CNRS, France

A R T I C L E   I N F O

Article history:
Received 5 November 2014
Received in revised form 23 September 2015
Accepted 28 September 2015
Available online 3 October 2015

Keywords:
Finite automata and regular languages
Subwords and superwords
State complexity
Combined operations
Closures and interiors of regular languages

A B S T R A C T

The downward and upward closures of a regular language L are obtained by collecting all the subwords and superwords of its elements, respectively. The downward and upward interiors of L are obtained dually by collecting words having all their subwords and superwords in L, respectively. We provide lower and upper bounds on the size of the smallest automata recognizing these closures and interiors. We also consider the computational complexity of decision problems for closures of regular languages.

1. Introduction

State complexity is a standard measure of the descriptive complexity of regular languages. The most common state complexity problems ask, given a regularity-preserving operation f on languages, to bound the size of an automaton recognizing $f(L)$ when L is recognized by an n-state automaton. We refer to [26,46] for a survey of the main known results in the area.

In this article, we consider language operations based on subwords. Recall that a (scattered) subword of some word x is a word obtained from x by removing any number of letters at arbitrary positions in x, see formal definitions in Section 2. Symmetrically, a superword is obtained by inserting letters at arbitrary positions. Subwords and superwords occur in many areas of computer science, from searching in texts and databases [4] to the theory of codes [29], computational linguistics [40], and DNA computing [33].

For a language $L \subseteq \Sigma^*$, we write $\downarrow L$ for the set of all its subwords and $\uparrow L$ for the set of all its superwords (in Σ^*) and call them the *downward closure* and *upward closure* of L, respectively. Dual to closures are *interiors*. The *upward interior* and *downward interior* of L, denoted $\circlearrowleft L$ and $\circlearrowright L$, are the largest upward-closed and downward-closed sets included in L. It has

* Corresponding authors.
E-mail address: phs@lsv.ens-cachan.fr (Ph. Schnoebelen).

1 Partially funded by Tata Consultancy Services.
2 Supported by Grant MA 4938/21 of the DFG.
3 Supported by Grant ANR-11-BS02-001.

http://dx.doi.org/10.1016/j.tcs.2015.09.028

Table 1
A summary of the results on state complexity for closures and interiors, where $\psi(n)$ ($\leqslant 2^{2^n}$) is the nth Dedekind's number.[1]

Operation	Unbounded alphabet	Fixed alphabet
$\uparrow L$ (DFA to DFA)	$=2^{n-2}+1$ for $\lvert\Sigma\rvert \geqslant n-2$	$2^{\Omega(n^{1/2})}$ for $\lvert\Sigma\rvert=2$
$\downarrow L$ (DFA to DFA)	$=2^{n-1}$ for $\lvert\Sigma\rvert \geqslant n-1$	$2^{\Omega(n^{1/3})}$ for $\lvert\Sigma\rvert=2$
$\uparrow L$ (AFA to AFA)	$\geqslant 2^{\lfloor\frac{n-3}{2}\rfloor}$ and $<2^n$ for $\lvert\Sigma\rvert$ in $2^{\Omega(n)}$	$\vdots$
$\downarrow L$ (AFA to AFA)	$\geqslant 2^{\lfloor\frac{n-4}{3}\rfloor}$ and $\leqslant 2^n$ for $\lvert\Sigma\rvert$ in $2^{\Omega(n)}$	(unknown)
$\circlearrowleft L$ (NFA to NFA)	$>2^{2^{\lfloor\frac{n-4}{3}\rfloor}}$ and $\leqslant\psi(n)$ for $\lvert\Sigma\rvert$ in $2^{\Omega(n)}$	$\vdots$
$\circlearrowright L$ (NFA to NFA)	$\geqslant 2^{2^{\lfloor\frac{n-3}{2}\rfloor}}$ and $\leqslant\psi(n)$ for $\lvert\Sigma\rvert$ in $2^{\Omega(n)}$	$\vdots$

been known since [22] that $\downarrow L$ and $\uparrow L$ are regular for any L. Then $\circlearrowright L$ and $\circlearrowleft L$ are regular too by duality, as expressed in the following equalities:

$$\circlearrowright L = \Sigma^* \setminus \uparrow(\Sigma^* \setminus L)\,, \qquad \circlearrowleft L = \Sigma^* \setminus \downarrow(\Sigma^* \setminus L)\,. \tag{1}$$

Computing closures and interiors has several applications in computer-aided reasoning [32] and program verification. Computing closures is an essential ingredient in the verification of safety properties of channel systems – see [1,20] – while computing interiors is required for the verification of their game-theoretical properties [5]. More generally, the regularity of upward and downward closures make them good overapproximations of more complex languages – see [3,21,47] – and interiors can be used as regular underapproximations.

Recently Gruber et al. explicitly raised the issue of the state complexity of downward and upward closures of regular languages [18,19] (less explicit precursors exist, for example, [7]). Given an n-state automaton A that recognizes L, automata $A^{\downarrow}$ and $A^{\uparrow}$ that recognize $\downarrow L$ and, $\uparrow L$ respectively can be obtained by simply adding extra transitions to A. However, when A is a deterministic automaton (a DFA), the resulting $A^{\downarrow}$ and $A^{\uparrow}$ are in general not deterministic (are NFAs), and their determinization may entail an exponential blowup. With n denoting the number of states of A, Gruber et al. proved a $2^{\Omega(\sqrt{n}\log n)}$ lower bound on the number of states of any DFA recognizing $\downarrow L$ or $\uparrow L$ [19], to be compared with the $2^n - 1$ upper bound that comes from the simple closure+determinization method.

Okhotin improved on these results by showing an improved $2^{\frac{n}{2}-2}$ lower bound for $\downarrow L$. He also established the exact state complexity for $\uparrow L$ by proving a $2^{n-2}+1$ upper bound and showing that this is tight [39].

All the above lower bounds assume an unbounded alphabet, and Okhotin showed that his $2^{n-2}+1$ state complexity for $\uparrow L$ requires $n-2$ distinct letters. He then considered the case of languages over a *fixed alphabet* and, in the 3-letter case, he demonstrated exponential $2^{\sqrt{2n+30}-6}$ and $\frac{1}{5}4^{\sqrt{n/2}}n^{-\frac{3}{4}}$ lower bounds for $\downarrow L$ and $\uparrow L$ respectively [39]. In the 2-letter case, Héam had previously proved an $\Omega(r^{\sqrt{n}})$ lower bound for $\uparrow L$, here with $r = (\frac{1+\sqrt{5}}{2})^{\frac{1}{\sqrt{2}}}$ [23]. Regarding $\downarrow L$, the question whether its state complexity is exponential even when $|\Sigma| = 2$ was left open (note that the one-letter case is trivial).

The state complexity of interiors has not yet been considered in the literature. When working with DFAs, complementation is essentially free so that computing interiors reduces to computing closures, thanks to duality. However, when working with NFAs, the simple complement+closure+complement method comes with a quite large 2^{2^n} upper-bound on the number of states of an NFA that recognizes $\circlearrowleft L$ or $\circlearrowright L$ – it actually yields DFAs – and one would like to improve on this, or to prove a matching lower bound. As we explain in Section 5.3, this is related to the state complexity of closures when working with alternating automata (AFAs), a question recently raised in [25].

Our contribution Regarding closures with DFAs, we prove in Section 3 a tight 2^{n-1} state complexity for downward closure and show that its tightness requires unbounded alphabets. In Section 4 we prove an exponential lower bound on both $\downarrow L$ and $\uparrow L$ in the case of a two-letter alphabet, answering the open question raised above. Regarding interiors on NFAs, we show in Section 5 doubly-exponential lower bounds for downward and upward interiors, assuming an unbounded alphabet. We also provide improved upper bounds, lower than the naive 2^{2^n} but still doubly exponential. Table 1 shows a summary of the results. Finally, Section 6 proves lower bounds on unambiguous automata for the witness languages used in Section 3, and Section 7 considers the computational complexity of some basic decision problems for sets of subwords or superwords described by automata.

[1] Recall that the nth Dedekind number $\psi(n)$ is the number of antichains in the lattice of subsets of an n-element set, ordered by inclusion [34]. Kahn [30, Corollary 1.4] shows

$$\binom{n}{\lfloor n/2\rfloor} \leqslant \log_2 \psi(n) \leqslant \left(1 + \frac{2\log(n+1)}{n}\right)\binom{n}{\lfloor n/2\rfloor}\,.$$