Theoretical Computer Science 601 (2015) 58–66

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Standard Sturmian words and automata minimization algorithms[☆]

G. Castiglione, M. Sciortino[*]

University of Palermo, Dipartimento di Matematica e Informatica, Via Archirafi 34, 90123 Palermo, Italy

ARTICLE INFO

Article history:
Received 30 January 2014
Accepted 10 April 2015
Available online 14 July 2015

Keywords:
Standard Sturmian words
Automata minimization
Brzozowski's minimization algorithm

ABSTRACT

The study of some close connections between the combinatorial properties of words and the performance of the automata minimization process constitutes the main focus of this paper. These relationships have been, in fact, the basis of the study of the tightness and the extremal cases of Hopcroft's algorithm, that is, up to now, the most efficient minimization method for deterministic finite state automata. Recently, increasing attention has been paid to another minimization method that, unlike the approach proposed by Hopcroft, is not based on refinement of the set of states of the automaton, but on automata operations such as determinization and reverse, and is also applicable to non-deterministic finite automata. However, even for deterministic automata, it was proved that the method incurs, almost surely, in an explosion of the number of the states. Very recently, some polynomial variants of Brzozowski's method have been introduced. In this paper, by using some combinatorial properties of words, we analyze the performance of one of such algorithms when applied to a particular infinite family of automata, called *standard word automata*, constructed by using standard sturmian words. In particular, $\Theta(n \log n)$ is the worst case time complexity when the algorithm is applied to the infinite family of word automata associated to Fibonacci words representing also the worst case of Hopcroft's minimization algorithm.

1. Introduction

The connections between Combinatorics on Words and Automata Minimization have been highly explored in the past decade. In particular, some combinatorial properties of words have been fundamental to study the difficult cases for many algorithms that minimize the finite state automata. In this context, the standard Sturmian words have played an important role as they have allowed to define families of automata on which the behavior of the main minimization algorithms have been analyzed. It is important to remember that standard Sturmian words have proved to be a crucial tool in the study of extremal cases of many algorithms in several contexts, such as data compression algorithms (cf. [22]), periodicity of words [15,16,23], pattern matching algorithm, suffix automata (cf. [26]).

Also in the context of the automata minimization, such words are connected to the extremal behavior of the classical minimization algorithms. One of the most well known strategies for automata minimization is based on refinements of a partition of the set of states of the input automaton. Based on such a strategy, Hopcroft's algorithm (cf. [20]) is the best known algorithm that minimizes a complete deterministic finite automaton with an $O(|Q|\log|Q|)$ worst-case complexity,

☆ This work is supported by Italian MIUR Project PRIN 2010LYA9RH, "Automi e Linguaggi Formali: Aspetti Matematici e Applicativi".
* Corresponding author.
E-mail addresses: giuseppa.castiglione@unipa.it (G. Castiglione), marinella.sciortino@unipa.it (M. Sciortino).

http://dx.doi.org/10.1016/j.tcs.2015.07.023
0304-3975/© 2015 Elsevier B.V. All rights reserved.

where Q is the set of states. It is important to note that Hopcroft's algorithm is not deterministic and this fact has made its analysis complicated. The problem of the tightness of the upper bound of the complexity has been posed and studied in [4]. Such a problem has been solved in [3,9–12] where infinite families of unary and binary automata representing the worst case for each execution of Hopcroft's algorithm are shown. The construction of such binary automata, called *word automata*, is driven by binary words both to define the label of the transitions and to determine whether a state is final or not. When such automata are constructed by using standard Sturmian words the minimization process become complex and this fact is strictly related to some combinatorial properties of the words.

The minimization algorithm proposed by Brzozowski (cf. [7]) is quite different. Although its worst-case behavior is exponential, it is simple conceptually and in practice. It works also on non-deterministic automata and finds the minimal equivalent deterministic automaton by performing reverse and determinization operations repeated twice. If such a strategy is applied to word automata associated to Fibonacci words, the time complexity is at least quadratic (cf. [8]). However, recent studies on generic and average complexities of Brzozowski's minimization algorithm (cf. [14]) have proved that both these complexity are super-polynomial for the uniform distribution on deterministic automata. In this context, it is interesting to consider very recent polynomial variants of Brzozowski's algorithm (cf. [17–19]) for deterministic automata. The variant introduced in [17,19], applied to word automata associated to standard Sturmian word, has been considered in [13] where the authors prove that the special circular factors of the word represent the strings able to distinguish the states of the different equivalent classes of the final partition for the input word automaton. A quadratic time complexity of the algorithm is obtained when Fibonacci words are considered. The main goal of this paper is to analyze the behavior of the algorithm proposed in [18]. In particular, we establish a relationship between the combinatorial properties of the words and the performance of the algorithm on word automata. We prove that, when the algorithm is applied to word automata associated to Fibonacci words, the time complexity of the worst execution becomes $\Theta(n \log n)$, as in the case of Hopcroft's minimization algorithm. This fact suggests from one hand that some standard Sturmian words, and consequently the associated word automata, are able to capture some properties for which the minimization problem becomes inherently more complex; from the other one, the method introduced in [18] could be considered an interesting alternative to the usual minimization algorithms.

In Section 2 we give some preliminary notions of combinatorics on words and automata theory. In Section 3 we prove some combinatorial properties of standard Sturmian words. Section 4 describes the notion of word automaton that is constructed by starting from a binary word. In Section 5 we describe the polynomial variant of Brzozowski's algorithm proposed in [18], we study its behavior on standard word automata and we compute the exact running time on standard word automata associated to Fibonacci words.

2. Preliminaries

This section is devoted to give same basic definitions and notations that will be used throughout the paper both in the fields of the combinatoric on words and automata minimization.

Let Σ be a finite alphabet and v, u be two words in Σ^*. We say that v and u are conjugate if for some words $z, w \in \Sigma^*$ one has that $v = zw$ and $u = wz$.

The conjugation is an equivalence relation.

We say that a word $v \in \Sigma^*$ is a *circular factor* of a word w if v is a factor of some conjugate of w. Equivalently, a circular factor of w is a factor of ww of length not greater than the length of w (denoted by $|w|$). For example, *ca* is a circular factor of *abc* and note that it is not factor of *abc*. We denote by $\mathit{fact}(w, k)$ the set of all circular factors of w of length k.

Moreover, a circular factor u of w is said to be *special* if both ua and ub are circular factors of w. We denote by $\mathit{SP}(w)$ the set of all circular special factors of w.

Let w be a word of length n and $ww = w_1 w_2 \dots w_{2n}$. An occurrence of a circular factor u in the word w is a position $i \le n$ such that $w_i \dots w_{|u|+i-1} = u$. For example, *ca* is a circular factor of *abcac* that occurs in position 3 and 5.

In this paper we focus on a particular class of words, called *standard Sturmian words*. Let $d_1, d_2, \dots, d_n, \dots$ be a sequence of natural integers, with $d_1 \ge 0$ and $d_i > 0$, for $i = 2, \dots, n, \dots$. Consider the following sequence of words $\{s_n\}_{n \ge 0}$ over the alphabet $A = \{a, b\}$: $s_0 = b$, $s_1 = a$, $s_{n+1} = s_n^{d_n} s_{n-1}$ for $n \ge 1$. Each finite word s_n in the sequence is called *standard Sturmian word*. It is uniquely determined by the (finite) directive sequence $(d_0, d_1, \dots, d_{n-1})$. In the special case where the directive sequence is of the form $(1, 1, \dots, 1, \dots)$ we obtain the sequence of Fibonacci words. We denote by f_n the n-th finite Fibonacci word and $F_n = |f_n|$.

The notion of circular factor allows the standard Sturmian words to inherit many combinatorial properties of the infinite Sturmian words (see [5] and [21]). The following proposition is an instance of this fact.

Proposition 1. *Let w be a word of length n. The word w is a conjugate of a standard Sturmian word if and only if for each $k = 0, \dots, n-2$ there exists a unique circular special factor of w of length k.*

Let w be a standard Sturmian word with $|w|_a > |w|_b$. We define the *signature* of w as the integer value $\lfloor \frac{|w|_a}{|w|_b} \rfloor$. Without loss of generality, in what follows, we take into consideration only standard Sturmian words w with $|w|_a > |w|_b$.

The following lemma gives a description of the structure of a generic standard Sturmian word up to conjugacy (cf. [10]).