Theoretical Computer Science 609 (2016) 469–493

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Combinatorics on partial word borders ☆

Emily Allen [a], F. Blanchet-Sadri [b,*], Michelle Bodnar [c], Brian Bowers [c], Joe Hidakatsu [d], John Lensmire [e]

[a] Department of Mathematical Sciences, Carnegie Mellon University, 5032 Forbes Ave., Pittsburgh, PA 15289, USA
[b] Department of Computer Science, University of North Carolina, P.O. Box 26170, Greensboro, NC 27402-6170, USA
[c] Department of Mathematics, University of California, San Diego, 9500 Gilman Drive #0112, La Jolla, CA 92093-0112, USA
[d] Department of Mathematics, University of Michigan, 530 Church Street, Ann Arbor, MI 48109-1043, USA
[e] Department of Mathematics, UCLA, Box 951555, Los Angeles, CA 90095, USA

ARTICLE INFO

Article history:
Received 20 May 2014
Received in revised form 12 September 2015
Accepted 3 November 2015
Available online 11 November 2015
Communicated by G. Ausiello

Keywords:
Combinatorics on words
Partial words
Borders
Border arrays

ABSTRACT

We develop a powerful graph theoretical approach that can compute the number of partial words, sequences with wildcard or hole characters, having a set of strong and weak periods, the number of partial words having a set of border lengths, the number of partial words having a maximum border length, the population size of a border array, the number of partial words having any set of required compatibilities and incompatibility sets, any of the above restricting to a fixed number of holes, any of the above restricting to a set of hole positions, to name a few. In the process, we establish some elegant relationships between these numbers, the Bell numbers, and the Stirling numbers of the second kind.

1. Introduction

The fundamental concept of a *border* of a string plays a major role in several research areas including string searching algorithms, pattern matching, text compression, and computational biology (see, for instance, [10,19]). A string $w = w[0..n)$ is said to have a border of length $\ell < n$ if its prefix of length ℓ is equal to its suffix of same length. We can classify strings by their border lengths, but we can also go a step further by examining the border lengths of prefixes. To do so, the *border array* $\mathbf{b} = \mathbf{b}[0..n)$ of w (also called its "failure function") is defined such that each $\mathbf{b}[i]$ is the length of the longest border of w's prefix of length $i + 1$. Problems of efficiently constructing, counting, validating, enumerating, and verifying border arrays provide interesting algorithmic and combinatorial challenges [2,11]. For instance, Moore, Smyth, and Miller [16] have showed how to count and generate all strings of length n constructed using exactly k letters that give rise to distinct patterns and distinct border arrays, for all positive integers n and k, providing algorithms that compute all such strings in constant time per string. Their ideas and results, which lead to algorithms that are much more time and space efficient than computing or counting $\Theta(k^n)$ strings, find applications for generating data sets useful in testing various algorithms on strings. Several other concepts related to borders have been introduced such as *abelian borders* [9,17], *parameterized border arrays* [21,22], *border correlations* [6,14], to name a few.

☆ This material is based upon work supported by the National Science Foundation under Grant Nos. DMS-0754154 and DMS-1060775. The Department of Defense is also gratefully acknowledged. Part of this material was presented at LATA'09 [4].

* Corresponding author.
E-mail address: blanchet@uncg.edu (F. Blanchet-Sadri).

http://dx.doi.org/10.1016/j.tcs.2015.11.006
0304-3975/

The above-mentioned border related concepts have been studied not only for regular strings (or total words), but also for strings with wildcard or hole characters (or partial words) that allow positions of the string to match any letter of the alphabet (a subclass of the so-called indeterminate strings). Partial words provide a generalization of strings that has both theoretical and practical importance due to the fact that they model data that is not perfect but corrupted. It is useful to consider several different methods for classifying partial words. Two partial words are *equal* if they represent the same sequence of characters, and they are *compatible* if they are equal for all positions where both are non-hole characters. Aside from equality and compatibility, we can consider partial words according to their border lengths and border arrays. A non-empty partial word is *bordered* if one of its proper prefixes is compatible with one of its suffixes; it is *unbordered* otherwise. Two types of borders have been identified: *simple* and *non-simple* (see [1,5,7] for recent works on bordered partial words).

The problem of generalizing to partial words the enumeration of strings with distinct border arrays was suggested in [1]. Unfortunately, as discussed there, a translation of Moore et al.'s results to partial words is not trivial since some canonical strings associated with the border arrays cannot be obtained using their tree construction, and additionally, some border arrays cannot be generated at all. In this paper, we describe graph theoretical approaches that yield many interesting connections between various values of the *population size* of a border array, i.e., the number of partial words sharing the array, as well as many results that can be used to study properties of partial words. Our compatibility graph represents the character compatibilities a partial word must have in order to have a specific border array. We show how p-distinct partial words, i.e., they represent distinct patterns, and b-distinct partial words, i.e., they have distinct border arrays, having a fixed length n and constructed with exactly k letters, can be counted. We bound the number of holes in partial words with specific border lengths and bound the number of b-distinct partial words with various properties. In doing so, we establish some elegant relationships between these numbers, the Bell numbers, and the Stirling numbers of the second kind.

The contents of our paper are as follows: In Section 2, we review some basic concepts on partial words, borders, and border arrays, and discuss some preliminary results on them including a relationship between border lengths and strong and weak periods. In Section 3, we compute the maximum number of holes a non-simply bordered partial word of a fixed length over a k-letter alphabet can have, and we show that this number is constant for all $k \geq 2$. We also study this number when we replace "non-simply bordered" by "unbordered", obtaining an upper bound by using a result of Turán in extremal graph theory. An exact formula for $k = 2$ is also derived, and we prove that the number is constant for all large enough k. In Section 4, we extend the concept of the maximum number of holes a bordered partial word of a fixed length over k letters can have to specify the longest border length. We also examine when the partial word uses exactly k letters. In Section 5, we give a graphical approach to determining population sizes that uses the "connected component array" and that is motivated by the study of correlation population sizes of Guibas and Odlyzko, among others [8,13,18]. We also describe another approach that uses the "subgraph component polynomial" of Tittmann et al. [20] for the enumeration of vertex induced subgraphs with respect to the number of connected components. In Section 6, we study the hole set of a border array, i.e., the set of all possible sets of hole positions of a partial word with that border array. We give in particular a characterization of the hole set of all border arrays over the binary alphabet. In Section 7, we give a recursive formula, then a closed form formula, for counting weakly one-periodic partial words. The problem is equivalent to computing the subgraph component polynomial for a path graph. We also examine partial words with weak periods one through some given threshold. In Section 8, we count border arrays. Finally in Section 9, we conclude with some suggestions for future work.

2. Preliminaries on borders and border arrays

We first give an overview of basic concepts of combinatorics on partial words. We denote by $\mathbb{N}$ the set of non-negative integers $\{0, 1, \ldots\}$. For integers i and j such that $0 \leq i \leq j$, the set $\{i, \ldots, j\}$ is abbreviated by $[i..j]$ or by $[i..j+1)$.

Let A be a non-empty finite set called an *alphabet*. Each element $a \in A$ is a *letter*. A *total word* over A is a finite sequence of letters from A. A *partial word* over A is a finite sequence of characters from $A_\diamond = A \cup \{\diamond\}$, the alphabet A being extended with the "hole" character $\diamond$ (a *total word* is a partial word that does not contain the $\diamond$ character). We denote by $w[i]$ the character at position i of the partial word w.

The *length* of a partial word w is denoted by $|w|$ and represents the number of characters in w. The *empty word* is the sequence of length zero and is denoted by ε. For a partial word w, the powers of w are defined recursively by $w^0 = \varepsilon$ and for $i \geq 1$, $w^i = ww^{i-1}$. The set of all words over the alphabet A is denoted by A^*, while the set of all partial words over A is denoted by $A_\diamond^*$.

If w_1 and w_2 are two partial words over A of equal length, then w_1 is *contained in* w_2, denoted by $w_1 \subset w_2$, if $w_1[i] = w_2[i]$ whenever $w_1[i] \in A$. Partial words w_1 and w_2 are *compatible* if there exists a partial word w such that $w_1 \subset w$ and $w_2 \subset w$. This is denoted by $w_1 \uparrow w_2$. Given partial words w_1 and w_2 such that $w_1 \uparrow w_2$, the *least upper bound* of w_1 and w_2 is the partial word $w_1 \vee w_2$, where $w_1 \subset (w_1 \vee w_2)$ and $w_2 \subset (w_1 \vee w_2)$, and if $w_1 \subset w$ and $w_2 \subset w$ then $(w_1 \vee w_2) \subset w$.

A partial word u is a *factor* of a partial word w if there exist x, y such that $w = xuy$. The factor u is *proper* if $u \neq \varepsilon$ and $u \neq w$. We say that u is a *prefix* of w if $x = \varepsilon$ and a *suffix* of w if $y = \varepsilon$. For integers i and j such that $0 \leq i \leq j < |w|$, the notation $w[i..j]$, or $w[i..j+1)$, abbreviates the factor $w[i] \cdots w[j]$ of w.