HAL

archives-ouvertes.fr

On Maximal Unbordered Factors

Alexander Loptev, Gregory Kucherov, Tatiana Starikovskaya

▶ To cite this version:

HAL Id: hal-01250522

https://hal.archives-ouvertes.fr/hal-01250522

Submitted on 4 Jan 2016

On Maximal Unbordered Factors

Alexander Loptev[1], Gregory Kucherov[2], and Tatiana Starikovskaya[3]

[1] Higher School of Economics, `alexander.loptev@gmail.com`
[2] Laboratoire d'Informatique Gaspard Monge, Université Paris-Est & CNRS,
Marne-la-Vallée, Paris, France, `gregory.kucherov@univ-mlv.fr`
[3] University of Bristol, Bristol, United Kingdom, `tat.starikovskaya@gmail.com`

Abstract. Given a string S of length n, its maximal unbordered factor is the longest factor which does not have a border. In this work we investigate the relationship between n and the length of the maximal unbordered factor of S. We prove that for the alphabet of size $\sigma \geq 5$ the expected length of the maximal unbordered factor of a string of length n is at least $0.99n$ (for sufficiently large values of n). As an application of this result, we propose a new algorithm for computing the maximal unbordered factor of a string.

1 Introduction

If a proper prefix of a string is simultaneously its suffix, then it is called a border of the string. Given a string S of length n, its maximal unbordered factor is the longest factor which does not have a border. The relationship between n and the length of the maximal unbordered factor of S has been a subject of interest in the literature for a long time, starting from the 1979 paper of Ehrenfeucht and Silberger [7].

Let $b(S)$ be the length of the maximal unbordered factor of S and $\pi(S)$ be the minimal period of S. Ehrenfeucht and Silberger showed that if the minimal period of S is smaller than $\frac{1}{2}n$, then $b(S) = \pi(S)$. Following this, they raised a natural question: How small $b(S)$ must be to guarantee $b(S) = \pi(S)$? Their conjecture was that $b(S)$ must be smaller than $\frac{1}{2}n$. However, this conjecture was proven false two years later by Assous and Pouzet [1]. As a counterexample they gave a string

$$S = a^m b a^{m+1} b a^m b a^{m+2} b a^m b a^{m+1} b a^m$$

of length $n = 7m+10$. The length of the maximal unbordered factor of this string is $b(S) = 3m + 6 \leq \frac{3}{7}n + 2 < \frac{1}{2}n$ (with $b a^{m+1} b a^m b a^{m+2}$ and $a^{m+2} b a^m b a^{m+1} b$ being unbordered), and the minimal period $\pi(S) = 4m + 7 \neq b(S)$.

The next attempt to answer the question was undertaken by Duval [3]: He improved the bound to $\frac{1}{4}n + \frac{3}{2}$. But the final answer to the question of Ehrefeucht and Silberger was given just recently by Holub and Nowotka [10]. They showed that $b(S) \leq \frac{3}{7}n$ implies $b(S) = \pi(S)$, and, as follows from the example of Assous and Pouzet, this bound is tight.

Therefore, when either $b(S)$ or $\pi(S)$ is small, $b(S) = \pi(S)$. Exploiting this fact, one can even compute the maximal unbordered factor itself in linear time. The key idea is that in this case the maximal unbordered factor is an unbordered conjugate of the minimal period of S, and both the minimal period and its unbordered conjugate can be found in linear time [15, 6].

The interesting cases are those where $b(S)$ (and, consequently, $\pi(S)$) is big. Yet, it is generally believed that they are the most common ones. This is supported by experimental resuts shown in Fig. 1 that plots the average difference between the length n of a string and the length of its maximal unbordered factor. Guided by the experimental results, we state the following conjecture:

Conjecture 1. Expected length of the maximal unbordered factor of a string of length n is $n - \mathcal{O}(1)$.
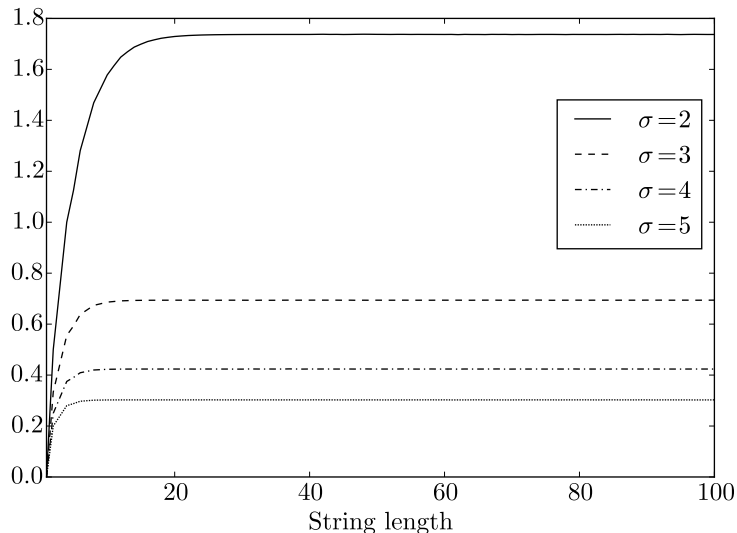

Fig. 1: Average difference between the length n of a string and the length of its maximal unbordered factor for $1 \leq n \leq 100$ and alphabets of size $2 \leq \sigma \leq 5$.

To the best of our knowledge, there have been no attempts to prove the conjecture or any lower bound at all in the literature. In Section 4 we address this gap and make the very first step towards proving the conjecture. We show that the expected length of the maximal unbordered factor of a string of length n over the alphabet A of size $\sigma \geq 2$ is at least $n(1 - \xi(\sigma) \cdot \sigma^{-4}) + \mathcal{O}(1)$, where $\xi(\sigma)$ is a function that converges to 2 quickly with the growth of σ. In particular, this theorem implies that for alphabets of size $\sigma \geq 5$ the expected length of the maximal unbordered factor of a string is at least $0.99n$ (for sufficiently large values of n). To prove the theorem we developed a method of generating strings

with large unbordered factors which we find to be interesting on its own (see Section 3).

It follows that the algorithm for computing maximal unbordered factors we sketched earlier cannot be used in a majority of cases. Instead, one can consider the following algorithm. A border array of a string is an array containing the maximal length of a border of each prefix of this string. Note that a prefix of a string is unbordered exactly when the corresponding entry in the border array is zero. Therefore, to compute the maximal unbordered factor of a string S it suffices to build border arrays of all suffixes of a string. It is well-known that a single border array can be constructed in linear time, which gives quadratic time bound for the algorithm. In Section 5 we show how to modify this algorithm to make use of the fact that the expected length of the maximal unbordered factor is big. We give $\mathcal{O}(\frac{n^2}{\sigma^4})$ time bound for the modified algorithm, as well as confirm its efficiency experimentally.

Related work. Apart from the aforementioned results, we consider our work to be related to three areas of research.

As we have already mentioned, the maximal unbordered factor can be found by locating the rightmost zeros in the border arrays of suffixes of a string and better understanding of structure of border arrays would give more efficient algorithms for the problem. Structure of border arrays has been studied in [9, 8, 5, 4, 14, 2].

In contrast to the problem we consider in this work, one can be interested in the problem of preprocessing a string to answer online factor queries related to its borders. This problem has been considered by Kociumaka et al. [13, 12]. They proposed a series of data structures which, in particular, can be used to determine if a factor is unbordered in logarithmic time.

Finally, repeating fragments in a string (borders of factors is one example of such fragments) were studied in connection with the *Longest Common Extension* problem which asks, given a pair of positions i, j in a string, to return the longest fragment that occurs both at i and j. This problem has many solutions, yet recently Ilie at al. [11] showed that the simplest solution, i.e. simply scanning the string and comparing pairs of letters starting at positions i and j, is the fastest on average. The authors also proved that the longest common extension has expected length smaller than $\frac{1}{\sigma-1}$, where σ is the size of the alphabet.

2 Preliminaries

We start by introducing some standard notation and definitions.

Power sums. We will need the following identities.

Fact 1. $S(x) = \sum_{i=1}^{k} i\, x^{i-1} = \frac{k\, x^{k+1} - (k+1)\, x^k + 1}{(x-1)^2}$ *for all $x \neq 1$.*

Proof.

$$S(x) = \Big(\sum_{i=1}^{k} x^i\Big)' = \Big(\frac{x^{k+1} - x}{x - 1}\Big)' = \frac{((k+1)x^k - 1)(x - 1) - (x^{k+1} - x)}{(x - 1)^2}$$

Simplifying, we obtain

$$S(x) = \sum_{i=1}^{k} i\, x^{i-1} = \frac{k\, x^{k+1} - (k+1)\, x^k + 1}{(x - 1)^2}$$

□

Corollary 1. $S(x) = \sum_{i=1}^{k} i\, x^{i-1} = \frac{k\, x^k}{x - 1} + \mathcal{O}(x^{k-2})$ *for $x \geq 1.5$.*

Strings. The alphabet A is a finite set of size σ. We refer to the elements of A as *letters*. A *string* over A is a finite ordered sequence of letters (possibly empty). Letters in a string are numbered starting from 1, that is, a string S of *length n* consists of letters $S[1], S[2], \ldots, S[n]$. The length n of S is denoted by $|S|$. A set of all strings of length n is denoted A^n.

For $1 \leq i \leq j \leq n$, $S[i..j]$ is a *factor* of S with endpoints i and j. The factor $S[1..j]$ is called a *prefix* of S, and the factor $S[i..n]$ is called a *suffix* of S. A prefix (or a suffix) different from S and the empty string is called *proper*.

If a proper prefix of a string is simultaneously its suffix, then it is called a *border*. For example, borders of a string *ababa* are *a* and *aba*. The *maximal border* of a string is its longest border. For S we define its *border array* B (also known as the *failure function*) to contain the lengths of the maximal borders of all its prefixes, i.e. $B[i]$ is the length of the maximal border of $S[1..i]$, $i = 1..n$. The last entry in the border array, $B[n]$, contains the length of the maximal border of S. It is well-known that the border array and therefore the maximal border of S can be found in $\mathcal{O}(n)$ time and space [15].

A period of S is an integer π such that for all i, $1 \leq i \leq n - \pi$, $S[i] = S[i+\pi]$. The minimal period of a string has length $n - B[n]$, and hence can be computed in linear time as well.

Unbordered strings. A string is called *unbordered* if it has no border. Let $b(i, \sigma)$ be the number of unbordered strings in A^i. Nielsen [16] showed that unbordered strings can be constructed in a recursive manner, starting from unbordered strings of length 2 and inserting new letters in the "middle". The following theorem is a corollary of the proposed construction method:

Theorem 1 ([16]). *The sequence $\Big\{\frac{b(i,\sigma)}{\sigma^i}\Big\}_{i=1}^{\infty}$ is monotonically nonincreasing and it converges to a constant α, which satisfies $\alpha \geq 1 - \sigma^{-1} - \sigma^{-2}$.*

Corollary 2 ([16]). $b(i, \sigma) \geq \sigma^i - \sigma^{i-1} - \sigma^{i-2}$ *for all i.*

This corollary immediately implies that the expected length of the maximal unbordered factor of a string of length n is at least $n(1-\sigma^{-1}-\sigma^{-2})$. We improve this lower bound in the subsequent sections. We will make use of a lower bound on the number $b_j(i,\sigma)$ of unbordered strings such that its first letter differs from the subsequent j letters. An example of such string for $j = 2$ is *abcacbb*.

Lemma 1. $b_j(i,\sigma) \geq (\sigma-1)^{j+1}\sigma^{i-j-1} - \sigma^{i-2}$ *for all* $i \geq j+1$.

Proof. The number of such strings is equal to $b(i,\sigma)$ minus the number $b_j^-(i,\sigma)$ of unbordered strings of length i that do not have the property. We estimate the latter from above by the number of such strings in the set of all strings with their first letter not equal to the last letter. Hence, $b_j^-(i,\sigma) \leq (\sigma-1)\sigma^{i-1} - (\sigma-1)^{j+1}\sigma^{i-j-1}$. Recall that $b(i,\sigma) \geq \sigma^i - \sigma^{i-1} - \sigma^{i-2}$ by Theorem 1. The claim follows. □

Remark. *The right-hand side of the inequality of Lemma 1 is often negative for* $\sigma = 2$. *We will not use it for this case.*

The *maximal unbordered factor* of a string (MUF) is naturally defined to be the longest factor of the string which is unbordered.

3 Generating strings with large MUF

In this section we explain how to generate strings of some fixed length n with large maximal unbordered factors. To show the lower bounds we announced, we will need many of such strings. The idea is to generate them from unbordered strings.

Let S be an unbordered string of length $i \geq \lceil \frac{n}{2} \rceil$. Consider a string $SP_1 \ldots P_k$ of length n, where $P_1, \ldots, P_k$ are prefixes of S. It is not difficult to see that the maximal unbordered factor of any string of this form has length at least i. (Because S is one of its unbordered factors.) The number of such strings that can be generated from S is 2^{n-i-1}, because each of them corresponds to a composition of $n-i$, i.e. representation of $n-i$ as a sum of a sequence of strictly positive integers. But, some of these strings can be equal. Consider, for example, an unbordered string $S = aaabab$. Then the two strings *aaababaaa* (S appended with its prefix *aaa*) and *aaababaaa* (S appended with its prefixes *a* and *aa*) will be equal. However, we can show the following lemma.

Lemma 2. *Let* $S_1 \neq S_2$ *be two unbordered strings. Any two strings of the form above generated from* S_1 *and* S_2 *are distinct.*

Proof. Suppose that the produced strings are equal. If $|S_1| = |S_2|$, we immediately obtain $S_1 = S_2$, a contradiction. Otherwise, w.l.o.g. assume $|S_1| < |S_2|$. Then S_2 is equal to a concatenation of S_1 and some of its prefixes. The last of these prefixes is simultaneously a suffix and a prefix of S_2, i.e. S_2 is not unbordered. A contradiction. □

Our idea is to produce as many strings of the form $SP_1 \ldots P_k$ as possible, taking extra care to ensure that all strings produced from a fixed string S are distinct. From unbordered strings of length $i = n$ and $i = n - 1$ we produce just one string of length n. (For $i = n$ it is the string itself and for $i = n - 1$ it is the string appended with its first letter.) For unbordered strings of length $i \leq n - 2$ we propose a different method based on the lemma below.

Lemma 3. *Each unbordered string S of length i such that its first letter differs from the subsequent j letters, where $\lceil n/2 \rceil \leq i < n - j$, gives at least 2^j distinct strings of the form $SP_1 \ldots P_k$.*

Proof. We choose the last prefix P_k to be the prefix of S of length at least $n - i - j$. We place no restrictions on the first $k - 1$ prefixes.

Let us start by showing that all generated strings are distinct. Suppose there are two equal strings $SP_1 \ldots P_\ell$ and $SP_1' \ldots P_{\ell'}'$. Let P_d, P_d' be the first pair of prefixes that have different lengths. W.l.o.g. assume that $|P_d| < |P_d'|$. Then $d \neq \ell$ and hence $|P_d| \leq j = n - i - (n - i - j)$. It follows that P_d' (which is a prefix of S) contains at least two occurrences of $S[1]$, one at the position 1 and one at the position $|P_d| + 1 \leq j + 1$. In other words, we have $S[1] = S[|P_d| + 1]$ and $|P_d| + 1 \leq j + 1$, which contradicts our choice of S.

If the length of the last prefix is fixed to some integer $m \geq n - i - j$, then each of the generated strings $SP_1 \ldots P_k$ is defined by the lengths of the first $k - 1$ of the appended prefixes. In other words, there is one-to-one correspondence between the generated strings and compositions of $n - i - m$. (Here we use $i \geq \lceil n/2 \rceil$ to ensure that every composition corresponds to a sequence of prefixes of S.) The number of compositions of $n - i - m$ is 1 when $m = n - i$ and $2^{n-i-m-1}$ otherwise. Summing up for all m from $n - i - j$ to $n - i$ we obtain that the number of the generated strings is 2^j. $\qquad\square$

Let us estimate the total amount of strings produced by this method. We produce one string from each unbordered string of length i. Then, from each unbordered string of length i such that its first letter differs from the second letter, we produce $1 = 2 - 1$ more string. If the first letter differs both from the second and the third letters, we produce $2 = 2^2 - 1 - 1$ more strings. And finally, if the first letter differs from the subsequent j letters, we produce $2^{j-1} = 2^j - \left(1 + 1 + 2 + \ldots + 2^{j-2}\right)$ strings. It follows that the number of strings we can produce from unbordered strings of length $i \leq n - 2$ is

$$b(i, \sigma) + \sum_{j=1}^{n-i-1} 2^{j-1} \cdot b_j(i, \sigma)$$

Recall that the maximal unbordered factor of each of the generated strings has length at least i and that none of them can be equal to a string generated from an unbordered string of different length.

4   Expected length of MUF

In this section we prove the main result of this paper.

Theorem 2. *Expected length of the maximal unbordered factor of a string of length n over an alphabet A of size $\sigma \geq 2$ is at least*

$$n \cdot (1 - \xi(\sigma) \cdot \sigma^{-4}) + \mathcal{O}(1) \tag{1}$$

where $\xi(2) = 8$ and $\xi(\sigma) = \frac{2\sigma^3 - 2\sigma^2}{(\sigma-2)(\sigma^2 - 2\sigma + 2)}$ for $\sigma > 2$.

Before we give a proof of the theorem, let us say a few words about $\xi(\sigma)$. This function is monotonically decreasing for $\sigma \geq 2$ and quickly converges to 2. We give the first four values for $\xi(\sigma)$ (rounded up to 3 s.f.) and $1 - \xi(\sigma) \cdot \sigma^{-4}$ (rounded down to 3 s.f.) in the table below.

	$\sigma = 2$	$\sigma = 3$	$\sigma = 4$	$\sigma = 5$
$\xi(\sigma)$	8.000	7.200	4.800	3.922
$1 - \xi(\sigma) \cdot \sigma^{-4}$	0.500	0.911	0.981	0.993

Corollary 3. *Expected length of the maximal unbordered factor of a string of length n over the alphabet A of size $\sigma \geq 5$ is at least $0.99n$ (for sufficiently large values of n).*

Proof of Theorem 2. Let $\beta_i^n(\sigma)$ be the number of strings in A^n such that the length of their maximal unbordered factor is i. Expected length of the maximal unbordered factor is then equal to

$$\frac{1}{\sigma^n} \sum_{i=1}^n i \cdot \beta_i^n(\sigma)$$

For the sake of simplicity, we temporarily omit $\frac{1}{\sigma^n}$, and only in the very end we will add it back. Recall that in the previous section we showed how to generate a set of distinct strings of length n with maximal unbordered factors of length at least i which contains

$$b(i, \sigma) + \sum_{j=1}^{n-i-1} 2^{j-1} \cdot b_j(i, \sigma)$$

strings for all $\lceil \frac{n}{2} \rceil \leq i \leq n - 2$ and $b(i, \sigma)$ strings for $i = \{n - 1, n\}$. Then

$$\sum_{i=1}^n i \cdot \beta_i^n(\sigma) \geq \underbrace{\sum_{i=\lceil n/2 \rceil}^n i \cdot b(i, \sigma)}_{(S_1)} + \underbrace{\sum_{i=\lceil n/2 \rceil}^{n-2} \sum_{j=1}^{n-i-1} 2^{j-1} \cdot i \cdot b_j(i, \sigma)}_{(S_2)} \tag{2}$$

We start by computing (S_1). Applying Corollary 2 and replacing $b(i,\sigma)$ with $\frac{b(n,\sigma)}{\sigma^{n-i}}$ in (S_1), we obtain:

$$(S_1) \geq \sum_{i=\lceil \frac{n}{2} \rceil}^{n} i \, \frac{b(n,\sigma)}{\sigma^{n-i}} = \frac{b(n,\sigma)}{\sigma^{n-1}} \Big(\sum_{i=\lceil \frac{n}{2} \rceil}^{n} i \, \sigma^{i-1} \Big)$$

Note that the lower limit in inner sum of (S_1) can be replaced by one because the correcting term is small:

$$\frac{b(n,\sigma)}{\sigma^{n-1}} \sum_{i=1}^{\lceil n/2 \rceil - 1} i\sigma^{i-1} \leq \frac{n^2 \cdot b(n,\sigma)}{4\sigma^{n/2}} = \mathcal{O}(\sigma^n)$$

We finally use Corollary 1 for $x = \sigma$ and $k = n$ to compute the right-hand side of the inequality:

$$(S_1) \geq \frac{n\sigma}{\sigma - 1} \cdot b(n,\sigma) + \mathcal{O}(\sigma^n) \tag{3}$$

We note that for $\sigma = 2$ the right-hand side is at least $2n \cdot (2^n - 2^{n-1} - 2^{n-2}) + \mathcal{O}(2^n) = n \cdot 2^{n-1} + \mathcal{O}(2^n)$ by Corollary 2 and $(S_2) \geq 0$. Hence, $\sum_{i=1}^{n} i \cdot \beta_i^n(2) \geq n \cdot 2^{n-1} + \mathcal{O}(2^n)$. Dividing both sides by 2^n, we obtain the theorem.

Below we assume $\sigma > 2$ and for these values of σ give a better lower bound on (S_2). Recall that $b_j(i,\sigma) \geq (\sigma-1)^{j+1}\sigma^{i-j-1} - \sigma^{i-2}$ (see Lemma 1). It follows that

$$(S_2) \geq \sum_{i=\lceil n/2 \rceil}^{n-2} \sum_{j=1}^{n-i-1} 2^{j-1} \cdot i \cdot \big((\sigma-1)^{j+1}\sigma^{i-j-1} - \sigma^{i-2} \big)$$

Let us change the order of summation:

$$(S_2) \geq \sum_{j=1}^{\lfloor n/2 \rfloor - 1} 2^{j-1} \cdot \big((\sigma-1)^{j+1}\sigma^{-j} - \sigma^{-1} \big) \sum_{i=\lceil n/2 \rceil}^{n-j-1} i \cdot \sigma^{i-1}$$

We can replace the lower limit in the inner sum of (S_2) by one as it will only change the sum by $\mathcal{O}(\sigma^n)$. After replacing the lower limit, we apply Corollary 1 to compute the inner sum:

$$(S_2) \geq \sum_{j=1}^{\lfloor n/2 \rfloor - 1} 2^{j-1} \cdot \big((\sigma-1)^{j+1}\sigma^{-j} - \sigma^{-1} \big) \cdot (n-j-1)\frac{\sigma^{n-j-1}}{\sigma - 1} + \mathcal{O}(\sigma^n)$$

We divide the sum above into positive and negative parts:

$$\underbrace{\sum_{j=1}^{\lfloor n/2 \rfloor - 1} (n-j-1)\, 2^{j-1}(\sigma-1)^j \sigma^{n-2j-1}}_{(P)} - \underbrace{\sum_{j=1}^{\lfloor n/2 \rfloor - 1} (n-j-1)2^{j-1}\frac{\sigma^{n-j-2}}{\sigma - 1}}_{(N)}$$

We start by computing (N). We again apply the trick with the lower limit and Fact 1, and replace $(n - j - 1)$ with k.

$$(N) = \frac{2^{n-3}}{\sigma - 1} \sum_{k=\lceil \frac{n}{2} \rceil}^{n-2} k \left(\frac{\sigma}{2}\right)^{k-1} = \frac{(n-2)\sigma^{n-2}}{(\sigma - 1)(\sigma - 2)} + \mathcal{O}(\sigma^n)$$

Computing (P) is a bit more involved. We divide it into two parts:

$$(P) = \underbrace{\frac{(n-1)\sigma^{n-1}}{2} \cdot \sum_{j=1}^{\lfloor n/2 \rfloor - 1} \left(\frac{2(\sigma - 1)}{\sigma^2}\right)^j}_{R_1} - \underbrace{\sigma^{n-1} \sum_{j=1}^{\lfloor n/2 \rfloor - 1} j \, 2^{j-1} (\sigma - 1)^j \sigma^{-2j}}_{R_2}$$

(R_1) is a sum of a geometric progression and it is equal to

$$\frac{(n-1)\sigma^{n-1}}{2} \cdot \frac{\left(\frac{2(\sigma - 1)}{\sigma^2}\right)^{\lfloor n/2 \rfloor} - \frac{2(\sigma - 1)}{\sigma^2}}{\frac{2(\sigma - 1)}{\sigma^2} - 1} = \frac{(n-1)\sigma^{n-1}}{2} \cdot \frac{2(\sigma - 1)}{\sigma^2 - 2\sigma + 2} + \mathcal{O}(\sigma^n)$$

Lemma 4. $(R_2) = \mathcal{O}(\sigma^n)$.

Proof. We start our proof by rewriting (R_2):

$$(R_2) = \sigma^{n-3}(\sigma - 1) \cdot \sum_{j=1}^{\lfloor n/2 \rfloor - 1} j \left(\frac{2(\sigma - 1)}{\sigma^2}\right)^{j-1}$$

We apply Fact 1 for $x = \frac{2(\sigma - 1)}{\sigma^2}$ and $k = \lfloor n/2 \rfloor - 1$ to compute the inner sum.

$$(R_2) = \sigma^{n-3}(\sigma - 1) \cdot \frac{(\lfloor n/2 \rfloor - 1) \cdot \left(\frac{2(\sigma - 1)}{\sigma^2}\right)^{\lfloor n/2 \rfloor} - \lfloor n/2 \rfloor \cdot \left(\frac{2(\sigma - 1)}{\sigma^2}\right)^{\lfloor n/2 \rfloor - 1} + 1}{\left(\frac{2(\sigma - 1)}{\sigma^2} - 1\right)^2}$$

The claim follows. □

We now summarize our findings. From equations for (P), (N), (R_1), and (R_2) we obtain (after simplification):

$$(S_2) \geq (P) - (N) = n \cdot \left(\frac{\sigma^n - \sigma^{n-1}}{\sigma^2 - 2\sigma + 2} - \frac{\sigma^{n-2}}{(\sigma - 1)(\sigma - 2)}\right) + \mathcal{O}(\sigma^n) \qquad (4)$$

We now return back to Equation (2) and use our lower bounds for (S_1) and (S_2) together with Corollary 2 for $b(n, \sigma)$:

$$\sum_{i=1}^{n} i \cdot \beta_i^n(\sigma) \geq n \cdot \left(\frac{\sigma^{n+1} - \sigma^n - \sigma^{n-1}}{\sigma - 1} + \frac{\sigma^n - \sigma^{n-1}}{\sigma^2 - 2\sigma + 2} - \frac{\sigma^{n-2}}{(\sigma - 1)(\sigma - 2)}\right) + \mathcal{O}(\sigma^n)$$

We now simplify the expression above and return back $\frac{1}{\sigma^n}$ as we promised in the very beginning of the proof to obtain:

$$\frac{1}{\sigma^n} \sum_{i=1}^{n} i \cdot \beta_i^n(\sigma) \geq n \cdot (1 - \xi(\sigma) \cdot \sigma^{-4}) + \mathcal{O}(1) \qquad (5)$$

where $\xi(\sigma) = \frac{2\sigma^3 - 2\sigma^2}{(\sigma - 2)(\sigma^2 - 2\sigma + 2)}$. This completes the proof of Theorem 2. □

Remark. Theorem 2 actually provides a lower bound on the expected length of the maximal unbordered prefix (rather than that of the maximal unbordered factor), which suggests that this bound could be improved.

5   Computing MUF

Based on our findings we propose an algorithm for computing the maximal unbordered factor of a string S of length n and give an upper bound on its expected running time. A basic algorithm would be to compute the border arrays (see Section 2 for the definition) of all suffixes of S. The border arrays contain the lengths of the maximal borders of all prefixes of all suffixes of S, i.e., of all factors of S. It remains to scan the border arrays and to select the longest factor such that the length of its maximal border is zero. Since a border array can be computed in linear time, the running time of this algorithm is $\mathcal{O}(n^2)$.

   The algorithm we propose is a minor modification of the basic algorithm. We build border arrays for suffixes of S starting from the longest one. After building an array B_i for $S[i..n]$ we scan it and locate the longest factor $S[i..j]$ such that the length of its maximal border stored in $B_i[j]$ is zero. We then compare $S[i..j]$ and the current maximal unbordered factor (initialized with an empty string). If $S[i..j]$ is longer, we update the maximal unbordered factor and proceed. At the moment we reach a suffix shorter than the current maximal unbordered factor, we stop.

Theorem 3. *The maximal unbordered factor of a string of length n over an alphabet A of size σ can be found in $\mathcal{O}(\frac{n^2}{\sigma^4})$ expected time.*

Proof. Let $b(S)$ be the length of the maximal unbordered factor of S. Then the running time of the algorithm is $\mathcal{O}((n - b(S)) \cdot n)$, because $b(S)$ will be a prefix of one of the first $n - b(S) + 1$ suffixes of S (starting from the longest one). Averaging this bound over all strings of length n, we obtain that the expected running time is

$$\mathcal{O}\left(\frac{1}{\sigma^n} \sum_{S \in A^n} (n - b(S)) \cdot n\right) = \mathcal{O}\left(n \cdot \left(\frac{1}{\sigma^n} \sum_{S \in A^n} (n - b(S))\right)\right)$$

and $\frac{1}{\sigma^n} \sum_{S \in A^n} (n - b(S)) = \mathcal{O}(\frac{n}{\sigma^4})$ as it follows from Theorem 2 and properties of $\xi(\sigma)$.   □

   We performed a series of experiments to confirm that the expected running time of the proposed algorithm is much smaller than that of the basic algorithm. We compared the time required by the algorithms for strings of length $1 \le n \le 100$ over alphabets of size $\sigma = \{2, 3, 4, 5, 10\}$. The time required by the algorithms was computed as the average time on a set of size $N = 10^6$ of randomly generated strings of given length. The experiments were performed on a PC equipped with one 2.6 GHz Intel Core i5 processor. As it can be seen in Fig. 2, the minor modification we proposed decreases the expected

running time dramatically. Obtained results were similar for all considered alphabet sizes. All source files, results, and plots can be found in a repository `http://github.com/avlonger/unbordered`.
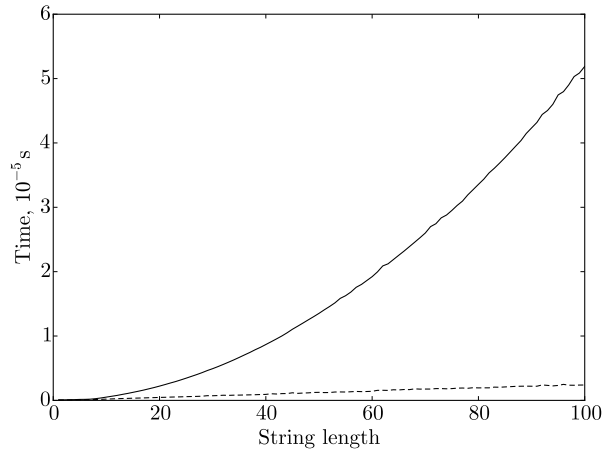

Fig. 2: Average running times of the proposed algorithm (dashed line) and the basic algorithm (solid line) for strings over the alphabet of size $\sigma = 2$.

We note that the data structures [13, 12] can be used to compute the maximal unbordered factor in a straightforward way by querying all factors in order of decreasing length. This idea seems to be very promising since these data structures need to be built just once, for the string S itself. However, the data structures are rather complex and both the theoretical bound for the expected running time, which is $\mathcal{O}(\frac{n^2}{\sigma^4} \log n)$, and our experiments show that this solution is slower than the one described above.

6 Conclusion

We consider the contributions of this work to be three-fold. We started with an explicit method of generating strings with large unbordered factors. We then used it to show that the expected length of the maximal unbordered factor and the minimal period of a string of length n is $\Omega(n)$, leaving the question raised in Conjecture 1 open. As an immediate application of our result, we gave a new algorithm for computing maximal unbordered factors and proved its efficiency both theoretically and experimentally.

Acknowledgements

References

1. R. Assous and M. Pouzet. Une caractérisation des mots périodiques. *Discrete Mathematics*, 25(1):1–5, 1979.
2. J. Clément and L. Giambruno. On the number of prefix and border tables. In *Proc. of the Eleventh Latin American Theoretical Informatics Symposium*, volume 8392 of *Lecture Notes in Computer Science*, pages 442–453. Springer Berlin Heidelberg, 2014.
3. J.-P. Duval. Relationship between the period of a finite word and the length of its unbordered segments. *Discrete Mathematics*, 40(1):31–44, 1982.
4. J.-P. Duval, T. Lecroq, and A. Lefebvre. Border array on bounded alphabet. *J. Autom. Lang. Comb.*, 10(1):51–60, January 2005.
5. J.-P. Duval, T. Lecroq, and A. Lefebvre. Efficient validation and construction of Knuth-Morris-Pratt arrays. In *Proc. of Conference in Honor of Donald E. Knuth*, 2007.
6. J.-P. Duval, T. Lecroq, and A. Lefebvre. Linear computation of unbordered conjugate on unordered alphabet. *Theor. Comp. Sci.*, 522(0):77–84, 2014.
7. A. Ehrenfeucht and D.M. Silberger. Periodicity and unbordered segments of words. *Discrete Mathematics*, 26(2):101–109, 1979.
8. F. Franěk, S. Gao, W. Lu, P. J. Ryan, W. F. Smyth, Y. Sun, and L. Yang. Verifying a border array in linear time. *J. Comb. Math. Comb. Comput*, 42:223–236, 2000.
9. P. Gawrychowski, A. Jeż, and Ł. Jeż. Validating the Knuth-Morris-Pratt Failure Function, Fast and Online. *Theor. Comp. Sys.*, 54(2):337–372, February 2014.
10. Š. Holub and D. Nowotka. The Ehrenfeucht–Silberger problem. *Journal of Comb. Theor., Series A*, 119(3):668–682, 2012.
11. L. Ilie, G. Navarro, and L. Tinta. The longest common extension problem revisited and applications to approximate string searching. *Journal of Discrete Algorithms*, 8(4):418–428, 2010.
12. T. Kociumaka, J. Radoszewski, W. Rytter, and T. Waleń. Efficient data structures for the factor periodicity problem. In *Proc. of the Nineteenth Symposium on String Processing and Information Retrieval*, volume 7608 of *Lecture Notes in Computer Science*, pages 284–294. Springer Berlin Heidelberg, 2012.
13. T. Kociumaka, J. Radoszewski, W. Rytter, and T. Waleń. Internal pattern matching queries in a text and applications. In *Proc. of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 532–551, 2014.
14. D. Moore, W. F. Smyth, and D. Miller. Counting distinct strings. *Algorithmica*, 23(1), 1999.
15. J.H. Morris Jr. and V.R. Pratt. A linear pattern-matching algorithm, report 40. Technical report, University of California, Berkeley, 1970.
16. P. Nielsen. A note on bifix-free sequences. *IEEE Trans. Inf. Theor.*, 19(5):704–706, September 1973.