Asymptotic behavior of the numbers of runs and microruns
Mathieu Giraud

To cite this version:
Mathieu Giraud. Asymptotic behavior of the numbers of runs and microruns. Information and Computation, Elsevier, 2009, 207 (11), pp.1221-1228. <10.1016/j.ic.2009.02.007>. <hal-00438214>
Asymptotic behavior of the numbers of runs and microruns

Mathieu Giraud², b

a CNRS, LIFL, Université Lille 1, 59 655 Villeneuve d’Ascq cedex, France
b INRIA Lille Nord-Europe, 59 650 Villeneuve d’Ascq, France

Abstract

The notion of run (also called maximal repetition) allows a compact representation of the set of all tandem periodicities, even fractional, in a string. Since the work of Kolpakov and Kucherov in [8, 9], it is known that $\rho(n)$, the maximum number of runs in a string, is linear in the length $n$ of the string. Lower bounds have been provided by Franek et al. and Matsubara et al. (0.9445...) [5, 6, 10], and upper bounds have been provided by Rytter, Puglisi et al., and Crochemore and Ilie (1.048n) [12, 11, 1, 2]. However, very few properties are known for the $\rho(n)/n$ function. We show here by a simple argument that $\lim_{n \to \infty} \rho(n)/n$ exists and that this limit is never reached. We further study the asymptotic behavior of $\rho_p(n)$, the maximal number of runs with period at most $p$. Finally, we provide the first exact limits for some microruns. For example, we have $\lim_{n \to \infty} \rho_{14}(n)/n = 15/17$.

Key words: word combinatorics, maximal repetitions, runs, asymptotic behavior, maximum number of runs

1. Introduction

The study of repetitions is an important field of research, both for word combinatorics theory and for practice, with applications in domains like computational biology or cryptanalysis. The notion of run (also called maximal repetition or m-repetition [8]) allows a compact representation of the set of all tandem periodicities, even fractional, in a string. The proper counting of those runs is important for all algorithms dealing with repetitions.

Since the work of Kolpakov and Kucherov in [8, 9], it is known that $\rho(n)$, the maximum number of runs in a string, is linear in the length $n$ of the string. They gave the first algorithm computing all runs in a linear time, but without an actual constant.

A preliminary version of this paper appeared in [3]. All the results in Section 4.2, with the exact limits given in Table 3, are new for this extended article.

Email address: mathieu.giraud@lifl.fr (Mathieu Giraud)
Upper bounds have been recently provided by Rytter (5n) [12] and Puglisi, Simpson, and Smyth (3.48n) [11]. A 1.6n bound was obtained by Crochemore and Ilie [1]. They count separately the microruns, that is the runs with short periods, and the runs with larger ones. They show that the number of microruns with period at most 9 satisfies $\rho_9(n) \leq n$. For larger runs, they prove that

$$\rho_{\geq p}(n) \leq \frac{2}{p} \left( \sum_{i=0}^{\infty} \left( \frac{2}{3} \right)^i \right) n = \frac{6}{p} \cdot n.$$

Crochemore, Ilie, and Tinta extended those results with massive computations, bringing down the upper bound from 1.6n to 1.048n [2].

A lower bound of $\alpha n$, with $\alpha = 3/(1 + \sqrt{5}) = 0.927...$, has been given by [5] then [6]. In [5], Franek, Simpson, and Smyth propose a sequence of strings $(x_n)$ with increasing lengths such that $\lim_{n \to \infty} \rho(x_n)/|x_n| = \alpha$, where $r(x)$ is the number of runs in the string $x$. In [6], Franek and Yang show that $\alpha$ is an asymptotic lower bound by showing that there exists a whole family of asymptotic lower bounds arbitrarily close to $\alpha$. Recently, Matsubara et al. provided an $174719/184973 = 0.9445...$ lower bound by repeating a large run-rich string [10].

In fact, very few properties are known for the $\rho(n)/n$ function [6, 13]. In this paper, after giving some definitions (Section 2), we show by a simple rewriting argument that $\ell = \lim_{n \to \infty} \rho(n)/n$ exists and that this limit is never reached (Section 3.1), proving that

$$\frac{\rho(n)}{n} \leq \ell - \frac{1}{4n}.$$

Section 3.2 proves the convergence of $\rho(n)/n$ even in the case of a fixed alphabet, for example for binary strings. In Section 4.1, we further study the asymptotic behavior of $\rho_p(n)$, the number of runs with short periods, showing that $\ell_p = \lim_{n \to \infty} \rho_p(n)/n$ exists and that, for some constant $z_p,$

$$\ell_p - \frac{z_p}{n} < \frac{\rho_p(n)}{n} < \ell_p \leq \ell.$$

Moreover, we provide a simple way to exactly count some microruns (Section 4.2). We give in Table 3 the first exact limits $\ell_p$ for microruns on binary strings with $p \leq 14$. Section 5 gives some concluding remarks.
2. Definitions

Let $x = x_1x_2 \ldots x_n$ be a string over an alphabet. Let $p \geq 1$ be an integer. We say that $x$ has a period $p$ if for any $i$ with $1 \leq i \leq n - p$, $x_{i+p} = x_i$. We denote by $x[i..j]$ the substring $x_ix_{i+1} \ldots x_j$. A run is an interval $[i..j]$:

- such that $x[i..j]$ has period $p \leq (j - i + 1)/2$,
- that is maximal: if they exist, neither $x_{i-1} = x_{i-1+p}$, nor $x_{j+1} = x_{j+1-p}$,
- and such that $x[i..i+p-1]$ is primitive: it is not an integer power of another string.

We define by $r_p(x)$ the number of runs of period $\leq p$ in $x$, called microruns in [1], and by $r(x) = r_{\lfloor |x|/2 \rfloor}(x)$ the total number of runs in $x$. For example, the four runs of $x = \text{atatatt}$ are $[4..5]$ (tt), $[7..8]$ (tt), $[1..4]$ (atat) and $[2..8]$ (tatatt), and thus $r_1(x) = 2$, $r_2(x) = 3$, and $r_3(x) = r(x) = 4$. Given an integer $n \geq 2$, we now consider all strings of length $n$. We define as

$$\rho_p(n) = \max \{ r_p(x) \mid |x| = n \}$$

the maximum number of runs of period $\leq p$ in a string of length $n$. Then we define as

$$\rho(n) = \max \{ r(x) \mid |x| = n \} = \rho_{\lfloor n/2 \rfloor}(n)$$

the maximum number of runs in a string of length $n$. Kolpakov and Kucherov gave in [9] some values for $\rho(n)$ (Table 1). Table 4, at the end of this paper, shows some values for $\rho_p(n)$. Note that $r(x) = \rho(|x|)$ does not imply that $r_p(x) = \rho_p(|x|)$ for all $p$: for example, $r(\text{aatat}) = 2 = \rho(5)$ but $r_1(\text{aatat}) = 1 < \rho_1(5) = 2$.

Finally, we can define values $r_{\geq p}(x)$ and $\rho_{\geq p}(n)$ for macroruns, that is runs with a period at least $p$. Again, $r(x) = \rho(|x|)$ does not imply that $r_{\geq p}(x) = \rho_{\geq p}(|x|)$. For example, $r_{\geq 2}(\text{aatt}) = 0 < \rho_{\geq 2}(4) = 1 = r_{\geq 2}(\text{atat})$.

3. On the number of runs

3.1. Rewritings and asymptotic behavior of the number of runs

Franek et al. [5, 6] list some known properties for $\rho(n)$:

- For any $n$, $\rho(n + 2) \geq \rho(n) + 1$
- For any $n$, $\rho(n + 1) \leq \rho(n) + \lfloor n/2 \rfloor$
- For some $n$, $\rho(n + 1) = \rho(n)$
- For some $n$, $\rho(n + 1) = \rho(n) + 2$

We add the following two simple properties.
Proposition 1. The function $\rho$ is superadditive: for any $m$ and $n$, we have $\rho(m+n) \geq \rho(m) + \rho(n)$.

Proof. Take two strings $x$ and $y$ of respective lengths $m$ and $n$ such that $r(x) = \rho(m)$ and $r(y) = \rho(n)$. Let $\bar{y}$ be a rewriting of $y$ with characters not present in $x$. (See below for a discussion on the size of the alphabet.) Then $x\bar{y}$ is a string of length $m+n$ containing exactly the runs of $x$ and the rewritten runs of $y$. Thus $\rho(m+n) \geq r(x\bar{y}) = r(x) + r(y) = \rho(m) + \rho(n)$.

For any $t \geq 1$, we have in particular $\rho(tn) \geq t\rho(n)$.

Proposition 2. For any $n$, $\rho(4n) \geq 4\rho(n) + 1$

Proof. Take a string $x$ of length $n$ with $r(x) = \rho(n)$. Let $\bar{x}$ be a rewriting of $x$ with characters not present in $x$. Then $r(\bar{x}x\bar{x}x) \geq 4r(x) + 1$.

We can now state our main result:

Theorem 1. $\rho(n)/n$ converges to its upper limit $\ell$. Moreover, the limit is never reached, as for any $n$ we have

$$\frac{\rho(n)}{n} \leq \ell - \frac{1}{4n}.$$ 

Proof. Let $\ell$ be the upper limit of $\rho(n)/n$. This limit is finite because of [9]. Given $\varepsilon$, there is a $n_0$ such that $\rho(n_0)/n_0 \geq \ell - \varepsilon/2$. For any $n \geq n_0$, let be $t = [n/n_0]$. Then we have $\rho(n)/n \geq \rho(tn)/n \geq t\rho(n_0)/n$ by Proposition 1, thus $\rho(n)/n \geq t/(t+1) \cdot \rho(n_0)/n_0$. Let be $t_0$ such that $t_0/(t_0 + 1) \cdot \rho(n_0)/n_0 \geq \rho(n_0)/n_0 - \varepsilon/2$. Then, for any $n \geq t_0n_0$, we have $\rho(n)/n \geq \ell - \varepsilon$, thus $\ell = \lim_{n \to \infty} \rho(n)/n$. Finally, Proposition 2 gives $\ell \geq \rho(4n)/4n \geq \rho(n)/n + \frac{1}{4n}$.

The proof of convergence of $f(n)/n$ when $f$ is superadditive is known as Fekete’s Lemma [4, 14]. This convergence result was an open question of [6]. In fact, the motivation of [6] was the remark that “the sequence $|x_i|$ (of [5]) is only “probing” the domain of the function $\rho(n)$ and $r(x_i)$ is “pushing” the value of $\rho(n)$ above any in these “probing” points”. Then Frank and Yang [6] prove that every $\alpha - \varepsilon$ is an actual asymptotic lower bound by building specific sequences. With Propositions 1 and 2 and Theorem 1, it is now sufficient to study bounds on any $(\rho(n_i)/n_i)$ sequence (for a growing sequence $(n_i)$) to give bounds on $\rho(n)/n$.

Note that this convergence does not imply monotonicity. In fact, if $\ell < 1$, then $\rho(n)/n$ is asymptotically non monotonic, as there will be in this case an infinity of $n$’s such that $\rho(n+1) = \rho(n)$. Note also that, although Proposition 1 and 2 require to double the alphabet size, the alphabet remains finite: the proof of Theorem 1 only requires to double once this alphabet size. Moreover, it is possible to prove Proposition 1 without rewriting in a larger alphabet, thus proving the convergence of $\rho(n)/n$ when considering only binary strings. This second proof, more elaborated, is given in the next section.
The bound \( \ell - \frac{1}{\alpha} \) can be improved. For example, with a rewriting similar to the one used in Proposition 2, it can be shown that \( \rho(2n^2) \geq (2n + 1)\rho(n) \), giving by successive iterations \( \rho(n)/n \leq \ell - \frac{1}{2n} \). This has not been reported here to keep the proof simple.

Concerning microruns with period at most \( p \), Proposition 1 still holds:

**Proposition 3.** For any \( p, m, \) and \( n \), we have \( \rho_p(m + n) \geq \rho_p(m) + \rho_p(n) \). Thus for any \( p \), \( \rho_p(n)/n \) converges to its upper limit \( \ell_p \).

The proof is the same as above. On the contrary, Proposition 2 may be not true for microruns. For example, for any \( n \), \( \rho_1(n) = [n/2] \), and thus for any even \( n \), we have \( \rho_1(n)/n = \ell_1 = 1/2 \).

Finally, Theorem 1 is fully valid for macroruns. Moreover, taking the following inequality to the limit, we have \( \ell_{\geq p} \geq \ell/p \).

**Proposition 4.** For any \( p \) and \( n \), \( \rho_{\geq p}(pn) \geq \rho(n) \).

**Proof.** Take a string \( x = x_1x_2\ldots x_n \) of length \( n \) with \( r(x) = \rho(n) \). Let \( \bar{x} \) be the string \( x_1^2x_2^2\ldots x_n^2 \) of length \( pn \). Then \( r_{\geq p}(\bar{x}) \geq r(x) \).

### 3.2. A proof of Proposition 1 for fixed alphabets

Here we prove Proposition 1 without rewriting in a larger alphabet, thus proving the convergence of \( \rho(n)/n \) when considering only binary strings. This proof is borrowed and simplified from one part of a proof of Franek et al. (Theorem 2 of [5]). A key observation is that some runs of \( x \) and \( y \) are merged in \( xy \) only when a string \( z^2 \) is both a suffix of \( x \) and a prefix of \( y \) (case \( a_2 \) on Figure 1).

We first have this property:

**Proposition 5.** Let \( \Sigma \) be an alphabet with \( |\Sigma| \geq 2 \), and let \( x \) and \( y \) be strings on \( \Sigma \) such that \( |y| \geq 1 \). Then there exists strings \( x' \) and \( y' \) on \( \Sigma \) such that \( |x'| + |y'| = |x| + |y|, |y'| < |y| \) and \( r(x') + r(y') \geq r(x) + r(y) \).

**Proof.** Let \( w \) be the longest string, eventually empty, such that \( w \) is a suffix of \( x \) and a prefix of \( y \). Thus \( x = uw \) and \( y = uv \) for some strings \( u \) and \( v \). Let \( x' = uvw \) and \( y' = w \). Clearly \( |x'| + |y'| = |x| + |y| \) and \( |y'| \leq |y| \). Without loss of generality, we assume that \( y \) is not a suffix of \( x \). (If it is not the case, we rewrite \( y \) into \( \tilde{y} \) using an isomorphism of \( \Sigma \) onto itself.) Thus \( |y'| < |y| \). Now we consider the runs of period \( p \) that were counted in \( r(x) + r(y) \). The runs with \( 2p \) characters (“a square”) completely included in \( w \) were counted once in \( r(x) \) and once in \( r(y) \). Such runs are counted again once in \( r(x') \) and once in \( r(y') \). By definition of \( w \), all the others runs counted in \( r(x) \) and \( r(y) \) are counted exactly once in \( r(x') \), without being merged.

To prove Proposition 1, we take two strings \( x_0 \) and \( y_0 \) of respective lengths \( m \) and \( n \) such that \( r(x_0) = \rho(m) \) and \( r(y_0) = \rho(n) \). Applying recursively Proposition 5 gives a finite sequence of pairs of strings \((x_0, y_0), (x_1, y_1), \ldots (x_t, y_t)\) with \( r(x_t) + r(y_t) \geq r(x_{t-1}) + r(y_{t-1}) \) and \( |y_0| > |y_1| > \ldots > |y_t| = 0 \) for some \( t \).
Finally, $|x_t| = |x_0| + |y_0| = m + n$, and thus $\rho(m + n) \geq r(x_t) \geq r(x_0) + r(y_0) = \rho(m) + \rho(n)$, proving Proposition 1.

Note that the proof of Franek et al. in [5] was in a different context, and that no result leading to our Proposition 1 was stated as such in their paper.

4. On the number of microruns

In the following sections, $p$ is fixed and we study the asymptotic behavior of the number of microruns $\rho_p(n)/n$ beyond the result of Proposition 3. In section 4.1, the idea is to bound the new runs created by the concatenation of two strings. In section 4.2, the idea is to count exactly the new runs created by the concatenation of a string and a character. Both sections provide new bounds or exact limits on the number of some microruns.

4.1. New runs obtained by string concatenation

Let $x$ and $y$ be two strings, and $s$ be a run of $xy$ with period $q \leq p$. Then $s$ is exactly in one of the following two cases (Figure 1):

- a) $s$ has at least two periods included in $x$, or at least two periods included in $y$;
- b) $s$ has strictly less than two periods included in $x$, and strictly less than two periods included in $y$.

We call the runs in the case b) the new runs between $x$ and $y$, and we denote by $\text{NR}_p(x,y)$ the number of such runs. Then $r_p(xy) \leq r_p(x) + r_p(y) + \text{NR}_p(x,y)$, the inequality coming from the fact that a run from $x$ can be merged with a run from $y$ (case a2 on Figure 1). We can bound the number of new runs, and thus have an upper bound on $r_p(xy)$:

**Proposition 6.** Let $z_p = \max\{\text{NR}_p(x,y) \mid |x| = |y| = 2p - 1\}$ the maximum number of new runs with period $q \leq p$ between strings of length $2p - 1$. Then, for every strings $x$ and $y$ of any length, we have $\text{NR}_p(x,y) \leq z_p$. 

Figure 1: a1) Run with at least two periods included in $x$. a2) Run with at least two periods included in $x$, and at least two periods included in $y$. b) “New runs” between $x$ and $y$. To bound the new runs with period $q \leq p$, it is sufficient to consider strings of length $4p - 2$. Note that $4p$ characters would be required to exactly count the new runs.
Table 2: Values for $z_p$ for binary strings with worst-case examples of length $\leq 4p - 2$.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$z_p$</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2</td>
<td>1</td>
<td>tt</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>ttat ta</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>ataaata ataat</td>
</tr>
<tr>
<td>5, 6, 7</td>
<td>5</td>
<td>ttatatta taatat</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>tttatatattta taattat</td>
</tr>
<tr>
<td>9, 10</td>
<td>7</td>
<td>ttatatatattata taatatataa</td>
</tr>
</tbody>
</table>

Proof. Any new run with period $q \leq p$ has at most $2q - 1 \leq 2p - 1$ characters in $x$, and in $y$ (Figure 1).

Proposition 7. For any $m$ and $n$, $\rho_p(m + n) \leq \rho_p(m) + \rho_p(n) + z_p$.

Proof. Let $x$ and $y$ be two strings such that $|x| = m$, $|y| = n$, and $r_p(xy) = \rho_p(m + n)$. Then $\rho_p(m + n) = r_p(xy) \leq r_p(x) + r_p(y) + \text{NR}_p(x, y) \leq \rho_p(m) + \rho_p(n) + z_p$.

Table 2 provides some values of $z_p$ for binary strings. An immediate bound on $z_p$ is $z_p \leq z_p - 1 + 2$. Knowing bounds on $z_p$ helps to characterize the asymptotic behavior of the number of microruns:

Theorem 2. For any $n$, we have $\ell_p \leq \rho_p(n)/n + z_p/n$, and thus

$$\ell_p - \frac{z_p}{n} \leq \frac{\rho_p(n)}{n} \leq \ell_p \leq \ell.$$

Proof. By Proposition 7, for any $t \geq 1$, we have $\rho_p(tn) \leq t\rho_p(n) + (t - 1)z_p$. Thus $\rho_p(tn)/tn \leq \rho_p(n)/n + \frac{t-1}{t}z_p/n$. Taking this inequality to the limit, as $t$ goes to infinity, gives the result.

Thus we know that the convergence of $\rho_p(n)/n$ to $\ell_p$ is faster than $z_p/n$. Note that we do not have a similar result for $\rho(n)$, as we do not have a convenient way to bound $\rho(m + n)$ like in Proposition 6.

As a side result of Theorem 2, we have new bounds of the number of some microruns. For example, brute-force computations give for binary strings $z_9 = 7$ and $\rho_9(34) = 26$, thus $\ell_9 \leq 33/34 = 0.970$. For binary strings, this result is better than Lemma 2 of [1] which proved the $n$ bound by the count of amortizing positions for centers of runs. The next section further improves this bound and finds exact values for some $\ell_p$’s.

4.2. The exact number of microruns

In this section, we propose to count exactly the number of microruns, by considering the concatenation of a string and a single character. Let $x$ be a string, $\alpha \in \Sigma$ a character, and $s$ be a run of $x\alpha$ with period $q \leq p$. Then $s$ is exactly in one of the following two cases (Figure 2):
Figure 2: a1) Runs of $x$, extended in $\alpha$. b) “New runs” between $x$ and $\alpha$. Unlike in Fig. 1, there is no case a2) where some runs are merged. To count the new runs with period $q \leq p$, it is sufficient to consider the suffix of $x$ of length $2p$.

- a) $s$ has at least two periods included in $x$;
- b) $s$ has strictly less than two periods included in $x$.

As in the previous section, we call the runs in the case b) the new runs between $x$ and $\alpha$, and we denote by $\text{NR}_p(x, \alpha)$ the number of such runs. In fact, the last $2p$ characters of $x$ are sufficient to know $\text{NR}_p(x, \alpha)$:

**Proposition 8.** Let $x$ be a string with $|x| = n \geq 2p$. Then we have $\text{NR}_p(x, \alpha) = \text{NR}_p(x[n-2p+1 \ldots n], \alpha)$.

**Proof.** Any new run with period $q \leq p$ has at most $2q-1 \leq 2p-1$ characters in $x$. Any run of $x$ with period $q \leq p$ extending in $\alpha$ has at least 2 periods in the last $2p$ characters of $x$ (Figure 2). Knowing the last $2p$ characters of $x$ is thus sufficient to tell apart the two kinds of runs.

If $v$ is a string of length $2p$ and $n \geq 2p$, we define $f_{n+1}^p(v) = \max_{x \in \Sigma} |x| + |v| = n \ r_p(xv)$ as the maximum number of runs of all the strings ending with the suffix $v$. The function $f_{n+1}^p$ can be entirely determined from the functions $f_{n}^p$ and $\text{NR}_p$:

**Proposition 9.** If $|w| = 2p-1$, $\alpha \in \Sigma$, and $n \geq 2p$, then

$$f_{n+1}^p(w\alpha) = \max_{\beta \in \Sigma} \left( f_{n}^p(\beta w) + \text{NR}_p(\beta w, \alpha) \right).$$

**Proof.** To compute $f_{n+1}^p(w\alpha) = \max_{x \in \Sigma} |x| + |w\alpha| = n+1 \ r_p(xw\alpha)$, we suppose, without loss of generality, that the string $x$ is of length at least one and we write $x = y\beta$, where $y$ is a string and $\beta \in \Sigma$.

$$f_{n+1}^p(w\alpha) = \max_{y\beta \in \Sigma} ( |y\beta| + |w\alpha| = n+1 \ r_p(y\beta w\alpha) )$$

$$= \max_{y\beta \in \Sigma} ( r_p(y\beta w) + \text{NR}_p(y\beta w, \alpha) )$$

$$= \max_{y\beta \in \Sigma} ( r_p(y\beta w) + \text{NR}_p(\beta w, \alpha) )$$

$$= \max_{\beta \in \Sigma} ( f_{n}^p(\beta w) + \text{NR}_p(\beta w, \alpha) )$$

(Proposition 8)
Once all the $|\Sigma|^{2p+1}$ values of NR\(p\) are computed, the above equation can be used to recursively determine any $f_{p}^{n}$ function in $O(n \cdot |\Sigma|^{2p+1})$ time. Then any number of microruns $\rho_p(n) = \max_{|v|=2p} f_{p}^{n}(v)$ follows. When $n$ grows, an additive periodic behavior can emerge:

**Theorem 3.** If for some $n_0$, $k$ and $s$ with $2p \leq n_0 < n_0 + k$, we have $f_{p}^{n_0+k} = f_{p}^{n_0} + s$ then, for any $n \geq n_0$, we have $f_{p}^{n+k} = f_{p}^{n} + s$, and $\ell_p = s/k$.

**Proof.** If the property $f_{p}^{n+k} = f_{p}^{n} + s$ is true for some $n$, then

$$f_{p}^{n+1+k}(w \alpha) = \max_{\beta \in \Sigma} \left( f_{p}^{n+k}(\beta w) + NR_{p}(\beta w, \alpha) \right)$$

$$= \max_{\beta \in \Sigma} \left( \left(f_{p}^{n}(\beta w) + s\right) + NR_{p}(\beta w, \alpha) \right)$$

$$= f_{p}^{n+1}(w \alpha) + s$$

and the property is true for $n + 1$. By induction, it is true for every $n \geq n_0$. In particular, for every $t \geq 0$, we have $f_{p}^{n_0+tk} = f_{p}^{n_0} + ts$, that is $\rho_p(n_0 + tk) = \rho_p(n_0) + ts$ and finally $\lim_{t \to \infty} \rho_p(n_0 + tk)/(n_0 + tk) = s/k$. As $\rho_p(n)/n$ converges (Proposition 3), the Theorem is proved.

By computing $f_{p}^{n}$ functions for successive $n$ and by checking the additive periodicity condition of Theorem 3, one can have exact values of $\ell_p$ for small $p$’s. Table 3 lists results for $p \leq 14$ on binary strings. Note that the periodicity on $\rho_p(n)$ can appear before the periodicity on $f_{p}^{n}$. For example, as soon as $n \geq 35$, $\rho_9(n + 13) = \rho_9(n) + 11$, but the periodicity on $f_{9}^{n}$ only starts at $n_0 = 51$.

Using the result of Crochemore and Ilie’s Proposition 1 [1] for large runs, we get an upper bound on $\rho(n)/n$. For binary strings, the exact value $\ell_{14} = 15/17$ gives:

$$\ell \leq \ell_{14} + \ell_{\geq 15} \leq \frac{15}{17} + \frac{6}{15} = 1.282...$$

Thus the number of runs in a binary string of length $n$ is not more than $1.29n$. This result was better than the 1.6$n$ bound published in [1], but the better bound of 1.048$n$ has now been published [2]. Nevertheless, the values we give in the Table 3 are the first known exact limits for such microruns.

**5. Perspectives**

The results on the asymptotic behavior of the functions $\rho$ and $\rho_p$ of Theorems 1 and 2 simplify the research on lower and upper bounds. Moreover, the application of the Theorem 3 provides the first exact limits for the number of some microruns. We hope that these results will bring a better understanding of the number of runs and be a step towards proving the conjecture of [8] ($\ell \leq 1$).

As Theorem 2 and 3 provide upper bounds or exact limits for some microruns, they can be used to bound the total number of runs. In both cases, this would require large evaluations of $z_p$ or $f_{p}^{n}(w)$ values that could be improved.
\[
\ell_1 = \ell_2 = \frac{1}{2} = 0.5 \quad f_1^4 = f_1^2 + 1
\]
\[
\ell_3 = \ell_4 = \frac{3}{4} = 0.75 \quad f_3^{16} = f_3^{12} + 3
\]
\[
\ell_5 = \ell_6 = \frac{7}{9} = 0.777... \quad f_5^{36} = f_5^{27} + 7
\]
\[
\ell_7 = \frac{4}{5} = 0.8 \quad f_7^{62} = f_7^{27} + 4
\]
\[
\ell_8 = \ell_9 = \ell_{10} = \ell_{11} = \ell_{12} = \frac{11}{13} = 0.846... \\
\begin{align*}
\ell_8 &= f_8^{49} + 11 \\
\ell_9 &= f_9^{51} + 11 \\
\ell_{10} &= f_{10}^{36} + 11 \\
\ell_{11} &= f_{11}^{107} + 11 \\
\ell_{12} &= f_{12}^{132} + 11
\end{align*}
\]
\[
\ell_{13} = \ell_{14} = \frac{15}{17} = 0.882... \\
\begin{align*}
\ell_{13} &= f_{13}^{113} = f_{13}^{96} + 15 \\
\ell_{14} &= f_{14}^{104} = f_{14}^{87} + 15
\end{align*}
\]

Table 3: Exact limits of \(\rho_p(n)/n\) for binary strings, obtained by successive applications of the equation in Proposition 9 until the additive periodicity condition of Theorem 3 is true. Each assertion on the right gives the smallest \(n \geq 2^p\) such that \(f_{p+k}^n = f_p^n + s\) for some \(s\) and \(k > 0\). The value \(\ell_{13}\) required three hours of computation on a standard 2 GHz workstation. This time is almost entirely spent in the initial computation of the \(2^{2^{p+1}}\) values of the function \(NR_p\), obtained by aggregate calls to \texttt{mreps} [7]. The successive computations of \(f_p^n\) are done in a few seconds.
<table>
<thead>
<tr>
<th>( n )</th>
<th>( \rho(n) )</th>
<th>( 1 )</th>
<th>( 2 )</th>
<th>( 3 )</th>
<th>( 4 )</th>
<th>( 5 )</th>
<th>( 6 )</th>
<th>( 7 )</th>
<th>( 8 )</th>
<th>( 9 )</th>
<th>( 10 )</th>
<th>( 11 )</th>
<th>( 12 )</th>
<th>( 13 )</th>
<th>( 14 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>17</td>
<td>12</td>
<td>8</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>18</td>
<td>13</td>
<td>9</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>19</td>
<td>14</td>
<td>9</td>
<td>12</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>20</td>
<td>15</td>
<td>10</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>21</td>
<td>15</td>
<td>10</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>11</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>23</td>
<td>17</td>
<td>11</td>
<td>15</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>12</td>
<td>16</td>
<td>16</td>
<td>17</td>
<td>17</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>25</td>
<td>19</td>
<td>12</td>
<td>17</td>
<td>17</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>26</td>
<td>20</td>
<td>13</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>27</td>
<td>21</td>
<td>13</td>
<td>18</td>
<td>18</td>
<td>19</td>
<td>19</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>22</td>
</tr>
<tr>
<td>28</td>
<td>22</td>
<td>14</td>
<td>19</td>
<td>19</td>
<td>20</td>
<td>20</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>29</td>
<td>23</td>
<td>14</td>
<td>20</td>
<td>20</td>
<td>21</td>
<td>21</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>30</td>
<td>24</td>
<td>15</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>22</td>
<td>22</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>32</td>
<td>26</td>
<td>16</td>
<td>22</td>
<td>22</td>
<td>23</td>
<td>23</td>
<td>24</td>
<td>24</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>33</td>
<td>27</td>
<td>16</td>
<td>23</td>
<td>23</td>
<td>24</td>
<td>24</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>34</td>
<td>27</td>
<td>17</td>
<td>24</td>
<td>24</td>
<td>25</td>
<td>25</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>35</td>
<td>28</td>
<td>17</td>
<td>24</td>
<td>24</td>
<td>25</td>
<td>25</td>
<td>26</td>
<td>26</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>28</td>
<td>28</td>
<td>28</td>
</tr>
</tbody>
</table>

Table 4: Values of \( \rho_p(n) \) for binary strings. For each \( n \), the value in bold shows the smallest period \( p \) such that \( \rho_p(n) = \rho(n) \).
References


