A Different Proof of Crochemore-Ilie Lemma Concerning Microruns[1]

Frantisek Franek and Jan Holub

ABSTRACT. We present a different computational proof of the estimate of the number of microruns established in a recent Crochemore-Ilie paper. The original proof in Crochemore-Ilie paper relies on computational means, and thus our proof provides an independent verification of the fact. We also introduce a notion of R-cover that is essential to our approach. The hope is that a further analysis of R-covers will lead to a non-computational proof of the upper bound of the number of microruns.

1 Introduction

An important structural characteristic of a string over an alphabet is its periodicity. Repetitions (tandem repeats) have always been in the focus of the research into periodicities. The notion of runs captures maximal repetitions which themselves are not repetitions and allows for a succinct notation ([10]). Even though it had been known that there could be $O(n \log n)$ repetitions in a string of length n ([1]), it was shown in 1997 by Ilioupoulos, Moore, and Smyth that number of runs in Fibonnaci strings is linear ([7]). In 2000, Kolpakov and Kucherov proved that number of runs was linear in the length of the input string ([8]). Their proof was existential and thus did not specify the constants of linearity. The behaviour of the **maxrun function** $\boldsymbol{\rho}(n) = \max\{\boldsymbol{r}(\boldsymbol{x}) \mid \textit{all strings } \boldsymbol{x} \textit{ of length } n\}$, where $\boldsymbol{r}(\boldsymbol{x})$ denotes the number of runs in a string $\boldsymbol{x}$, became an interest of study to many. In several papers (e.g. [4], [11], [3]) several conjectures about $\boldsymbol{\rho}(n)$ were put forth:

(1) $\boldsymbol{\rho}(n) < n$,

(2) $\lim_{|\boldsymbol{x}|\to\infty} \frac{\boldsymbol{\rho}(\boldsymbol{x})}{|\boldsymbol{x}|} = \frac{3}{1+\sqrt{5}}$

[1]Supported in part by a grant from the Natural Sciences & Engineering Research Council of Canada, a grant from the Ministry of Education, Youth and Sports of Czech Republic, and a grant from the Czech Science Foundation.

(3) $\boldsymbol{\rho}(n+1) \leq \boldsymbol{\rho}(n)+2$,

(4) for any n, there is a cube-free binary string $\boldsymbol{x}$ so that $\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{\rho}(\boldsymbol{x})$.

[4] introduced a construction of an increasing sequence $\{\boldsymbol{x}_n \ : \ n < \infty\}$ of binary strings "rich in runs" so that $\lim_{n\to\infty} \frac{\boldsymbol{r}(\boldsymbol{x_n})}{|\boldsymbol{x_n}|} = \alpha$, where $\alpha = \frac{3}{1+\sqrt{5}} \approx 0.927$. The technique was used by Franek and Yang to provide an asymptotic lower bound for $\boldsymbol{\rho}(n)$ ([5]). This proof was significantly simplified by Giraud ([6]). Just recently, [9] improved the lower bound, falsifying the conjecture (2). The current value of the lower bound 0.944565 (not published yet) can be found at the web site of one of the authors at

`http://www.shino.ecei.tohoku.ac.jp/runs/`

An explicit upper bound $6.3n$ was first given by Rytter in 2006 and immediately improved by him to $5n$ (see [12]), later improved more to $3.44n$. Crochemore and Ilie ([2]) lowered the upper bound to $1.6n$ using a different method. The current value of the upper bound standing at $1.048n$ (not published yet) can be found at the web site of Ilie at

`http://www.csd.uwo.ca/~ilie/runs.html`

Crochemore-Ilie approach relies in two estimates: the first is an estimate of the number of so-called δ-runs, and the other is an estimate of the number of microruns, i.e. runs with period ≤ 9. The first estimate is proven in the paper and states that in average, each interval of length δ contains at most one center of δ-run. The estimate of the number of microruns (Lemma 2 in the paper) states that *the number of microruns is bounded by the length of the string*. As a sketch of the proof, one of 512 different cases is analyzed. The supposedly complete and exhaustive list of all cases was generated using computer. So, the estimate of the number of microruns is established using computational means.

Since it is important for computational results to have independent verification, we present a totally different approach that establishes by computational means the estimate of the number of microruns. We also introduce a notion of R-cover that is essential to our approach. The hope is that a further analysis of R-covers will lead to a non-computational proof of the number of microruns.

2 Preliminaries and definitions

Definition 1. $\boldsymbol{x}[s..(s+ep+t)]$ is a **run** in a string $\boldsymbol{x}[1..n]$
if $\boldsymbol{x}[s..(s+p-1)] = \boldsymbol{x}[(s+p)..(s+2p-1)] = \cdots$
$= \boldsymbol{x}[(s+(e-1)p)..(s+ep-1)]$ and
$\boldsymbol{x}[(s+(e-1)p)..(s+(e-1)p+t)] = \boldsymbol{x}[(s+ep)..(s+ep+t)]$,

where $0 \leq s < n$ is the **starting position** of the run, $1 \leq p < n$ is the **period** of the run, $e \geq 2$ is the **exponent** (or **power**) of the run, and $0 \leq t < p$ is the **tail** of the run. Moreover, it is required that either $s = 0$ or that $\boldsymbol{x}[s{-}1] \neq \boldsymbol{x}[s{+}2p{-}1]$ (in simple terms it means that it is a leftmost repetition) and that $\boldsymbol{x}[s{+}(ep){+}t{+}1] \neq \boldsymbol{x}[s{+}e{+}1)p{+}t{+}1]$ (in simple terms it means that the tail cannot be extended to the right). It is also required, that the **generator** of the run, $\boldsymbol{x}[s..(s{+}p{-}1)]$ is **primitive**, i.e. not a repetition itself.
$\boldsymbol{x}[s..(s{+}2p{-}1)]$ is referred to as the **leftmost square** of the run, and $\boldsymbol{x}[(s{+}(e{-}2)p{+}t{+}1)..(s{+}ep{+}t)]$ as the **rightmost square** of the run (*for illustration see Figure 1*).

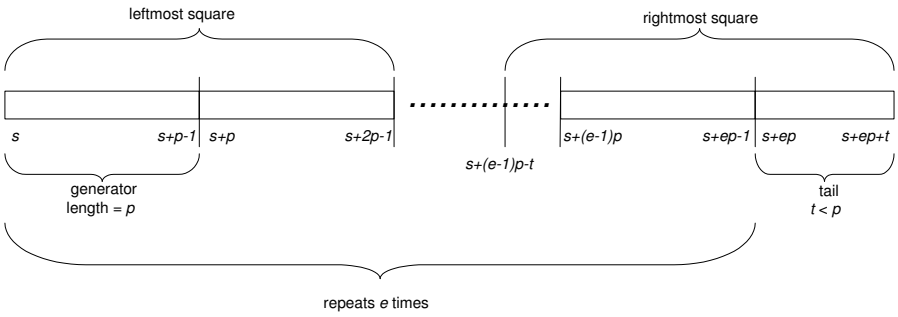

Figure 1. Illustration of a run

Note that each run can be uniquely encoded by the four-tuple (s, p, e, t).

The core of a run is an auxiliary notion used to construct R-covers (see Lemma 1). Intuitively, it is a set of positions which "destroys" the run if we split the run there into two parts.
We employ the convention that splitting a string $\boldsymbol{x}[1..n]$ in position k means breaking it into $\boldsymbol{x}[1..k]$ and $\boldsymbol{x}[k{+}1..n]$.
For instance, a run `aaa` cannot be destroyed by splitting: `a|aa` preserves a run `aa` from the original run `aaa`, `aa|a` does likewise, and thus it has an empty core. On the other hand, `ababab` can be split into `aba|bab` and the run is destroyed, so position 3 is in the core, while 2 is not since `ab|abab` preserves a run `abab` from the original run `ababab` (*for illustration using a more complex run* `abaabaa` *see Figure 2*).

Definition 2. The **core** of a run $\boldsymbol{r} = (s, p, e, t)$ is the set of positions where the leftmost and the rightmost squares of the run overlap less the last index,

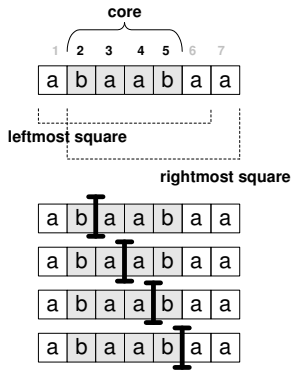

Figure 2. Core of a run and the "destruction" of the run by splitting it in a core position

or more precisely

$\{i : s \leq i < s{+}2p{-}1 \ \&\ s{+}(e{-}1)p{-}t \leq i < s{+}ep{+}t\}$.

Note: Any run with power ≥ 4 has an empty core regardless its period, so does the cube with period 1 (`aaa`), in a sense these runs are indestructible by splitting. Cubes with higher periods have non-empty cores, and so do squares. If a square has no tail, the core actually contains all positions (except the last one), that is a maximal core – such run can be destroyed by splitting it anywhere, for instance `abab`: `a|bab` or `ab|ab` or `aba|b`, none of the splitting preserves anything of the run.

It seems intuitively clear that in a string with a maximum number of runs, the runs must be distributed "uniformly" and "densely". The following notion of R-cover is an attempt to describe a dense distribution of runs in a string (*for illustration, see Figure 3*).

Definition 3. A set $\{\boldsymbol{r}_i : 1 \leq i < m\}$ of squares in a string $\boldsymbol{x}$ is an **R-cover** of $\boldsymbol{x}$ if

1. the union of all squares $\boldsymbol{r}_i$ is the whole of $\boldsymbol{x}$;
2. each square $\boldsymbol{r}_i$ has a primitive generator;
3. each square $\boldsymbol{r}_i$ is leftmost (i.e. cannot be shifted left);
4. the starting position of $\boldsymbol{r}_i <$ the starting position of $\boldsymbol{r}_{i+1}$, and the end position of $\boldsymbol{r}_i <$ the end position of $\boldsymbol{r}_{i+1}$;

5. for any run $\boldsymbol{r}$ in $\boldsymbol{x}$, the leftmost square of $\boldsymbol{r}$ is a substring of some $\boldsymbol{r}_i$.

In the following, for the sake of simplicity, a **microrun** (**microsquare**) indicates a run (square) with period ≤ 9. A **micro-R-cover** is an R-cover consisting of microsquares. $\mu(\boldsymbol{x})$ denotes the number of microruns in $\boldsymbol{x}$.

Lemma 1. Let $\boldsymbol{x} = \boldsymbol{x}[1..n]$ be a string. If every $1 \leq i < n$ is in the core of some microrun in $\boldsymbol{x}$, then there exists a micro-R-cover of $\boldsymbol{x}$.

Proof. Among all microruns that have 1 in its core, choose the one with the largest period, call it R_1. Set r_1 to the leftmost square of R_1. We proceed by induction.

Assume to have constructed $\{R_1 : 1 \leq m\}$ and $\{r_i : 1 \leq m\}$ such that $\{r_i : 1 \leq m\}$ satisfies 2-5 from Definition 3 and each r_i is the leftmost square of R_i. If $\bigcup_{1 \leq i \leq m} r_i = \boldsymbol{x}$, then condition 1 from Definition 3 is satisfied and $\{r_i : 1 \leq m\}$ is an R-cover and the proof is complete.
Otherwise pick the leftmost position $k \in \{1, \cdots, n-1\}$ that is not covered by $\bigcup_{1 \leq i \leq m} r_i$. Among microruns that have k covered by its leftmost square (at least one such must exists, since there is at least one that has k in its core), choose the leftmost ones, and among those, choose the run with the largest period, it is R_{m+1}. Set r_{m+1} to the leftmost square of R_{m+1}.
Since k is not covered by any r_i, $1 \leq i \leq m$, it is not in the core of any of the microruns R_i, $1 \leq i \leq m$, in fact k is to the right of the core of any R_i, $1 \leq i \leq m$. Since k is either in the core of R_{m+1} or to the left of the core of R_{m+1}, R_{m+1} is distinct from all R_i, $1 \leq i \leq m$. □ ■

The following definition of cut is another auxiliary notion. It allows to carry induction over number of microruns: if a string $\boldsymbol{x}[1..n]$ has a cut k, then $\mu(\boldsymbol{x}[1..n]) \leq k + \mu(\boldsymbol{x}[k+1..n])$.

Definition 4. A position $k < n$ in a string $\boldsymbol{x}[1..n]$ is a cut, if the number of all microruns with starting position $\leq k$ is $\leq k$, and it is a smallest such k.

The following lemma is crucial for our proof, it guarantees that under some conditions, a cut exists.

Lemma 2. Let $\boldsymbol{x}$ be an arbitrary string with a micro-R-cover $\{r_i : 1 \leq i \leq m\}$. Let r_1 have not tail. Let there be another microsquare s with a primitive generator of size $<$ the period of r_1, with no tail, and starting at position 1. Further assume that $|\boldsymbol{x}| > 35$. Then $\boldsymbol{x}$ has a cut.

Proof. Note that due to the size of $\boldsymbol{x}$, $m \geq 2$.
The proof is computational and was carried out by the following steps.

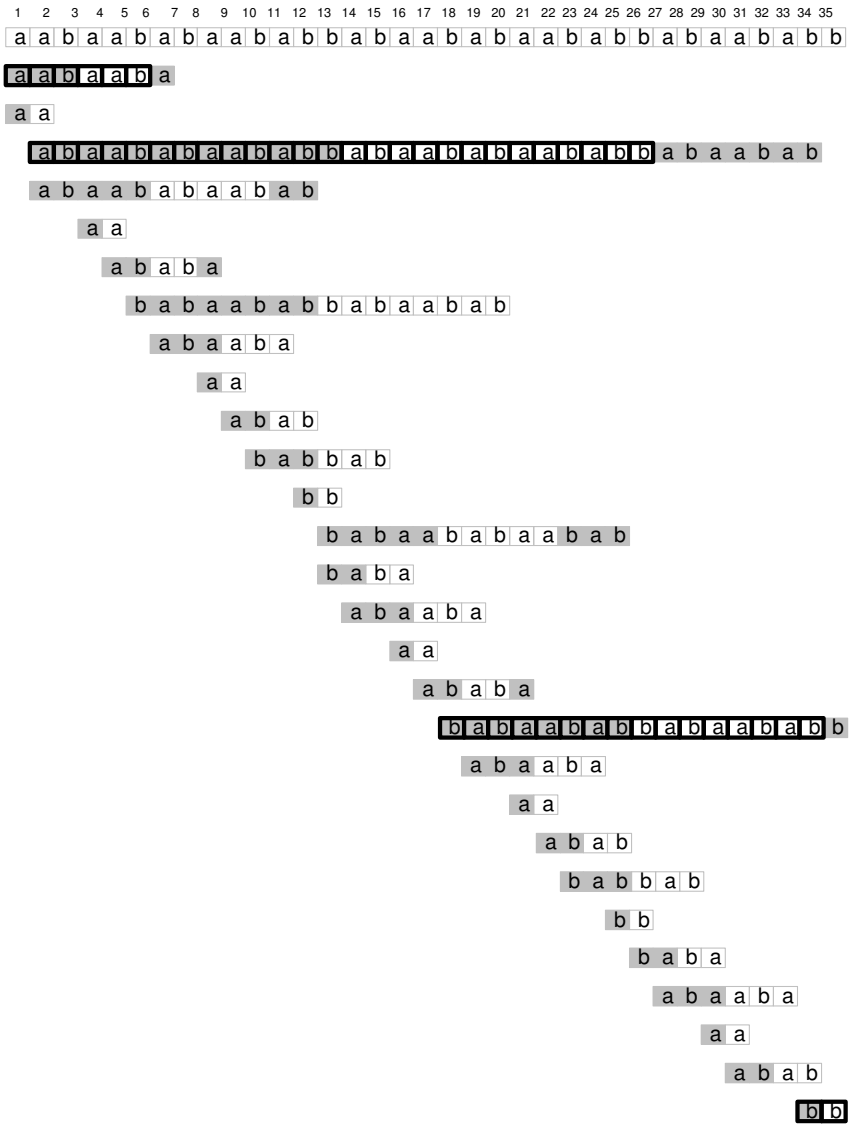

Figure 3. A string with all its runs, its R-cover is highlighted

1) s was generated, then it was extended to r_1.

2) The cut k_1 for r_1 was computed (that the cut must exist follows from the fact that $\rho(n) < n$ for all $n \leq 35$).
If r_2 starting position $> k_1$, then k_1 is a cut for $r_1 \cup r_2$ as well. Thus we tried to generate a "bad" r_2 with a starting position $\leq k_1$.
For most r_1 generated, only "good" r_2 could be generated, and so k_1 was the cut for $r_1 \cup r_2$ and thus the cut for $\bigcup_{1 \leq i \leq m} r_i = \boldsymbol{x}$ as well.
In a few cases a "bad" r_2 was generated. The configuration r_1, r_2 was then processed further.

3) The cut k_2 for $r_1 \cup r_2$ was computed (it was always successful).
If r_3 starting position $> k_2$, then k_2 is a cut for $r_1 \cup r_2 \cup r_3$ as well. Thus we tried to generate a "bad" r_3 with a starting position $\leq k_2$. Only "good" r_3 could be generated, and so k_2 was the cut for $r1 \cup r_2 \cup r_3$ and thus the cut for $\bigcup_{1 \leq i \leq m} r_i = \boldsymbol{x}$ as well. □

■

3 The main theorem and its proof

Theorem 1. For any string $\boldsymbol{x}$, $\mu(\boldsymbol{x}) \leq |\boldsymbol{x}|$.

Proof. It is known from various computational results, including the computations carried by the authors of this paper, that $\rho(n) < n$ for all $n \leq 35$, and so $\mu(\boldsymbol{x}) < |\boldsymbol{x}|$ for all strings $\boldsymbol{x}$ of size ≤ 35.
So we can assume that the size of $\boldsymbol{x}[1..n]$ is bigger than 35. We proceed by induction. At each stage, we discuss two cases.

Case 1: *there exists k, $1 \leq k < n$, that is not in the core of any microrun.*
Then $\mu(\boldsymbol{x}[1..n]) \leq \mu(\boldsymbol{x}[1..k]) + \mu(\boldsymbol{x}[k+1..n]$. By the induction hypothesis, $\mu(\boldsymbol{x}[1..n]) \leq \mu(\boldsymbol{x}[1..k]) + \mu(\boldsymbol{x}[k+1..n] \leq k + (n-k) = n$.

Case 2: *for any k, $1 \leq k < n$, k is in the core of some microrun.*
Then by Lemma 1, $\boldsymbol{x}$ has a micro-R-cover $\{r_i : 1 \leq i \leq m\}$. If the position 1 is not in the core of at least two microruns, then $\mu(\boldsymbol{x}[1..n]) \leq 1 + \mu(\boldsymbol{x}[2..n])$ and so by the induction hypothesis $\mu(\boldsymbol{x}[1..n]) \leq 1 + \mu(\boldsymbol{x}[2..n]) \leq 1 + (n-1) = n$.
So we can assume that position 1 is in the core of at least two microruns. It follows that r_1 must be a microsquare with no tail and that there is a microsquare s with a period $<$ the period of r_1, starting at position 1, and no tail. Thus the conditions of Lemma 2 are fulfilled and so there is a cut k. It follows that $\mu(\boldsymbol{x}) \leq k + \mu(\boldsymbol{s}[k+1..n) = k + (n-k) = n$. □ ■

4 Conclusion and further research

We presented an alternative computational proof of the estimate of the number of microrun. The method presented does not scale up well for higher periods – though Lemma 2 holds as is for periods ≤ 10 – for higher periods more than just two initial squares of the R-cover are needed before the cut is guaranteed.

However, the most interesting aspect of R-covers was not fully exploited here: for a given string $\boldsymbol{x}[1..n]$, if there is a position k in $\boldsymbol{x}$ that is not in the core of at least two microruns, than $\mu(\boldsymbol{x}[1..n]) \leq \mu(\boldsymbol{s}[1..k-1])+1+\mu(\boldsymbol{x}[k+1..n])$ and so by the induction hypothesis $\mu(\boldsymbol{x}) \leq n$. This indicates that induction breaks down only if a string has two micro-R-covers, one a refinement of the other. There is a hope (and computational results carried to date provide some evidence), that such double covers are not possible. The future research will thus focus on a non-computational proof that such double covers do not exist providing a route to a non-computational estimate of the number of microruns.

BIBLIOGRAPHY

[1] M. Crochemore: *An optimal algorithm for computing the repetitions in a word.* Inform. Process. Lett., 5 (5) 1981, pp. 297–315.

[2] M. Crochemore and L. Ilie: *Maximal repetitions in strings.* to appear in J. Comput. Syst. Sci.

[3] Fan Kangmin and W. F. Smyth: *A new periodicity lemma.* to appear in SIAM J. of Discr. Math.

[4] F. Franek, J. Simpson, and W. F. Smyth: *The maximum number of runs in a string* in Proceedings of 14th Australasian Workshop on Combinatorial Algorithms AWOCA 2003, Seoul National University, Seoul, Korea, July 13-16 2003.

[5] F. Franek and Q. Yang: *An asymptotic lower bound for the maximal number of runs in a string* Int. Journ. of Foundations of Computer Science, 1 (19) 2008, pp. 195–203.

[6] M. Giraud: *Not so many runs in strings* The proceedings of the LATA 2008, Tarragona, Spain, March 2008.

[7] C. S. Iliopoulos, D. Moore, and W. F. Smyth: *A characterization of the squares in a Fibonacci string* Theoretical Computer Science 172 (1997) 281-291.

[8] R. Kolpakov and G. Kucherov: *On maximal repetitions in words.* J. of Discrete Algorithms, (1) 2000, pp. 159–186.

[9] R. K. Kusano, W. Matsubara, A. Ishimo, H. Bannai, and A. Shinohara *New lower bounds for the maximum number of runs in a string*, CoRR, abs/0804.1214, May 2008, http://arxiv.org/abs/0804.1214,

[10] M. G. Main: *Detecting leftmost maximal periodicities.* Discrete Applied Maths., (25) 1989, pp. 145–153.

[11] S. J. Puglisi, W. F. Smyth, and A. Turpin: *Some restrictions on periodicity in strings*, in Proceedings of 16th Australasian Workshop on Combinatorial Algorithms AWOCA 2005, University of Ballarat, Victoria, Australia, September 18-21 2005, pp. 263–268.

[12] W. Rytter: *The number of runs in a string: Improved analysis of the linear upper bound*, in Proceedings of 23rd Annual Symposium on Theoretical Aspects of Computer Science STACS 2006, Marseille, France, February 23-25 2006, pp. 184–195.

Frantisek Franek
Department of Computing & Software
McMaster University
Hamilton, Ontario, Canada L8S 4K1
Email: franek@mcmaster.ca
http://www.cas.mcmaster.ca/~franek

Jan Holub
Department of Computer Science and Engineering
Faculty of Electrical Engineering
Czech Technical University in Prague
Karlovo Namesti 13, 121 35 Prague 2, Czech Republic
Email: holub@fel.cvut.cz
http://cs.felk.cvut.cz/~holub