On a lower bound for the maximum number of runs

Joint work: Q. Yang and F. Franek

Department of Computing and Software
Faculty of Engineering
McMaster University
Hamilton, Ontario, Canada

London Algorithmic Workshop, February 2007

1. notion of runs, maxrun function

2. known facts and conjectures

3. building sequences of strings "rich in runs" -- (a) method by *Simpson, Smyth,* and *F.*, (b) method by *Yang* and *F.*

5. constructing asymptotic lower bound from the sequences

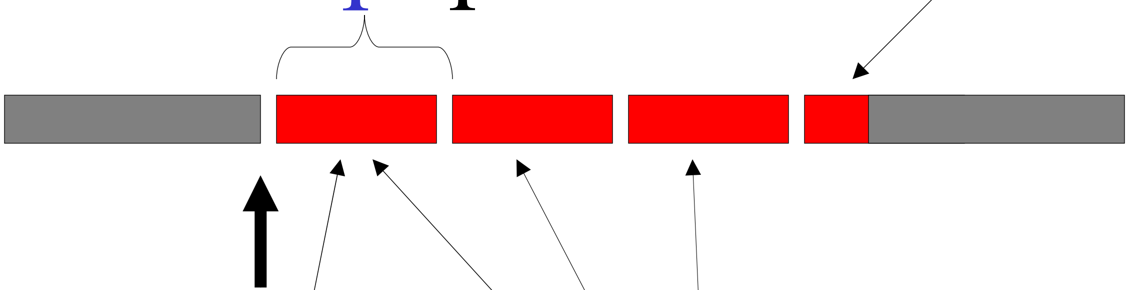
6. Further research

A **run** captures the notion of a maximal non-extendible repetition in a string ***x***

(s,p,e,t)

p period

t tail (rightmost)



s starting position (leftmost)

e power, exponent

irreducible generator

$$\boldsymbol{\rho(n) = \max \{ R(x) \mid |x|=n \}}$$

maxrun function

where $\mathbf{R(x)}$ is number of runs in $\mathbf{x}$

P1: $\rho(n+1) \geq \rho(n)$

P2: $\rho(n+2) \geq \rho(n)+1$

P3: $\rho(n+1) \leq \rho(n)+\left\lfloor \frac{n}{2} \right\rfloor$

P4: $\rho(n+1) = \rho(n)$ for some n

[$\rho(33)=\rho(34)=27$,

is it asymptotic?]

P5: $\rho(n+1) \geq \rho(n)+2$ for some n

[$\rho(13)=8$, $\rho(14)=10$,

is it asymptotic?]

Values of $\rho(n)$ computed by Kolpakov & Kucherov for $n \leq 32$

Franek & Smyth computed all run-maximal strings up to $n = 35$

Trivial lower bound: $\rho(n) \geq 0.5\, n$

CONJECTURES
(Smyth et al)

C1: $\rho(n) < n$

C2: $\rho(n+1) \leq \rho(n)+2$

C3: $\rho(n)$ attained by a binary cube-free string of length n

C1' : $$\underbrace{\lim_{n\to\infty} \frac{\rho(n)}{n}}_{\exists ?} = \underbrace{\frac{3}{1+\sqrt{5}}}_{\alpha} \cong 0.927$$

2000 *Kolpakov & Kucherov*

$$\rho(n) \le k_1 n - k_2 \log_2 n \sqrt{n}$$

2003 *Franek*, *Simpson*, *Smyth*

A recursive construction of an infinite sequence $\{x_n\}$ of binary strings of increasing length so that

$$\lim_{n \to \infty} \frac{R(x_n)}{|x_n|} = \alpha$$

2006 *Rytter*

$\rho(n) \le 5n$ 3.5n 3.44n 1.6n 1.18n ?

2006 *Franek*, *Yang*

Franek-Simpson-Smyth method can be used to get a family of asymptotic lower bounds:

$$(\forall \varepsilon > 0)(\exists\, N)(\forall n \ge N)(\ \rho(n) \ge (\alpha - \varepsilon)n\ )$$

2006 *Franek*, *Yang*

A recursive construction (based on a different philosophy) of an infinite sequence $\{x_n\}$ of binary strings of increasing length so that

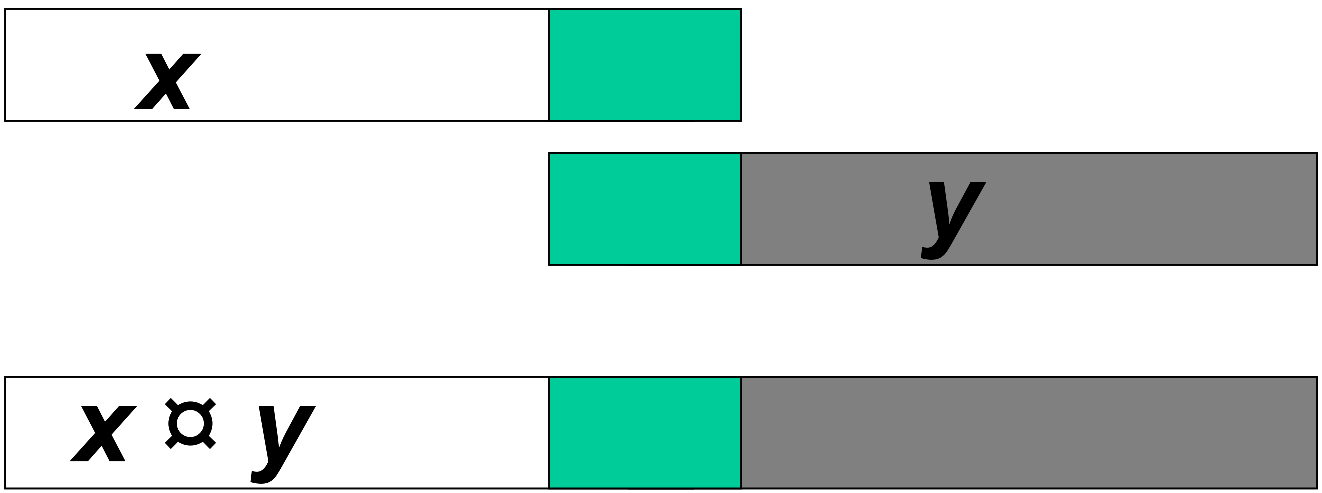
$$\lim_{n \to \infty} \frac{R(xn)}{|xn|} = \alpha$$

This result strengthens the case for C1'

Franek-Simpson-Smyth

The motivation for concatenation

x* ¤ *y



All the runs from ***x*** and ***y*** are preserved

$\boldsymbol{x}[1..n] ¤ \boldsymbol{y}[1..m] =$

$$= \begin{cases} \boldsymbol{x}[1..n]\boldsymbol{y}[2..m] & \text{if } \boldsymbol{x}[n]=\boldsymbol{y}[1] \\ \boldsymbol{x}[1..n-1]\boldsymbol{y}[2..m] & \text{if } \boldsymbol{x}[n]\neq\boldsymbol{y}[1] \end{cases}$$

we are working with two patterns p_0=010010 and p_1=101101

010010¤101101=0100101101

we preserved all the runs, gained one run, while shortening the length by one or two characters

$$\boldsymbol{g}(\boldsymbol{x}[1..n]) = \begin{cases} p_0 = 010010 & \text{if } \boldsymbol{x}[1]=0 \text{ \& } n=1 \\ p_1 = 101101 & \text{if } \boldsymbol{x}[1]=1 \text{ \& } n=1 \\ \boldsymbol{g}(\boldsymbol{x}[1]) ¤ \boldsymbol{g}(\boldsymbol{x}[2]) ¤ \ldots ¤ \boldsymbol{g}(\boldsymbol{x}[n]) & \end{cases}$$

Fact1: $|\boldsymbol{g}(\boldsymbol{x})| = 4|\boldsymbol{x}| + \lambda(\boldsymbol{x}) + 2$ where $\lambda(\boldsymbol{x})$ is the number of 00 and 11

Fact2: $\lambda(\boldsymbol{g}(\boldsymbol{x})) = |\boldsymbol{x}|$

Fact3: $R(\boldsymbol{g}(\boldsymbol{x})) = R(\boldsymbol{x}) + 3|\boldsymbol{x}| - 1$

This gives a recursive construction:

$\boldsymbol{x}_0$ an arbitrary binary string

$\boldsymbol{x}_{n+1} = \boldsymbol{g}(\boldsymbol{x}_n)$

and working out the recurrence relations for $|\ \ |$, $\lambda(\ )$, and $R(\ )$ lead to

$$\lim_{n \to \infty} \frac{R(\boldsymbol{x}_n)}{|\boldsymbol{x}_n|} = \alpha$$

How quickly it converges?

i	$\|x_i\|$	$R(x_i)$	$\lambda(x_i)$	$\frac{R(x_i)}{\|x_i\|}$
0	1	0	0	0
1	6	2	1	0.3333
2	27	19	6	0.7037
3	116	99	27	0.8534
4	493	463	116	0.9047
5	2090	1924	493	0.9206
6	8855	8193	2090	0.9252
7	35712	34757	8855	0.9266

Franek-Yang

according to C3 we should be "playing" with cube-free strings

Hence we define ¤ so to make sure that cubes of period 1 (000 or 111) and 2 (010101, or 101010) are eliminated during concatenation. We call such strings *loose cube-free*.

$\boldsymbol{x}[1..n] ¤ \boldsymbol{y}[1..m] =$

$$= \begin{cases} \boldsymbol{x}[1..n]\boldsymbol{y}[1..m] & \text{if } \boldsymbol{x}[n]=\boldsymbol{y}[1] \\ \boldsymbol{x}[1..n]\boldsymbol{y}[1]\boldsymbol{y}[1..m] & \text{if } \boldsymbol{x}[n]\neq\boldsymbol{y}[1] \end{cases}$$

we are working with two patterns p_0=0101 and p_1=1010

0101¤0101=010110101

we preserved all the runs, gained two runs, while increasing the length by 1

0101¤1010=01011010

we preserved all the runs, gained two runs, while preserving the length

$$\boldsymbol{g}(\boldsymbol{x}[1..n]) = \begin{cases} p_0 = 0101 & \text{if } \boldsymbol{x}[1]=0 \ \&\ n=1 \\ p_1 = 1010 & \text{if } \boldsymbol{x}[1]=1 \ \&\ n=1 \\ \boldsymbol{g}(\boldsymbol{x}[1]) ¤ \boldsymbol{g}(\boldsymbol{x}[2]) ¤ \ \dots\ ¤ \boldsymbol{g}(\boldsymbol{x}[n]) & \end{cases}$$

Fact1: If $\boldsymbol{x}$ is loose cube-free, so is $\boldsymbol{g}(\boldsymbol{x})$

Fact2: $|\boldsymbol{g}(\boldsymbol{x})| = 4|\boldsymbol{x}| + \lambda(\boldsymbol{x})$

Fact3: $\lambda(\boldsymbol{g}(\boldsymbol{x})) = |\boldsymbol{x}| - 1$

Fact4: $R(\boldsymbol{g}(\boldsymbol{x}))=R(\boldsymbol{x})+3|\boldsymbol{x}|-2-R_{bad}(\boldsymbol{x})$

Bad run: 

is lost during transformation by $\boldsymbol{g}()$

However, we can control it, so that $|R_{bad}(\boldsymbol{x})| < 2$.

This again gives a recursive construction:

$\boldsymbol{x}_0$ a "properly" chosen loose cube-free string, $\boldsymbol{x}_{n+1} = \boldsymbol{g}(\boldsymbol{x}_n)$

and working out the recurrence relations for $|\ \ |$, $\lambda(\ )$, and $R(\ )$ lead to

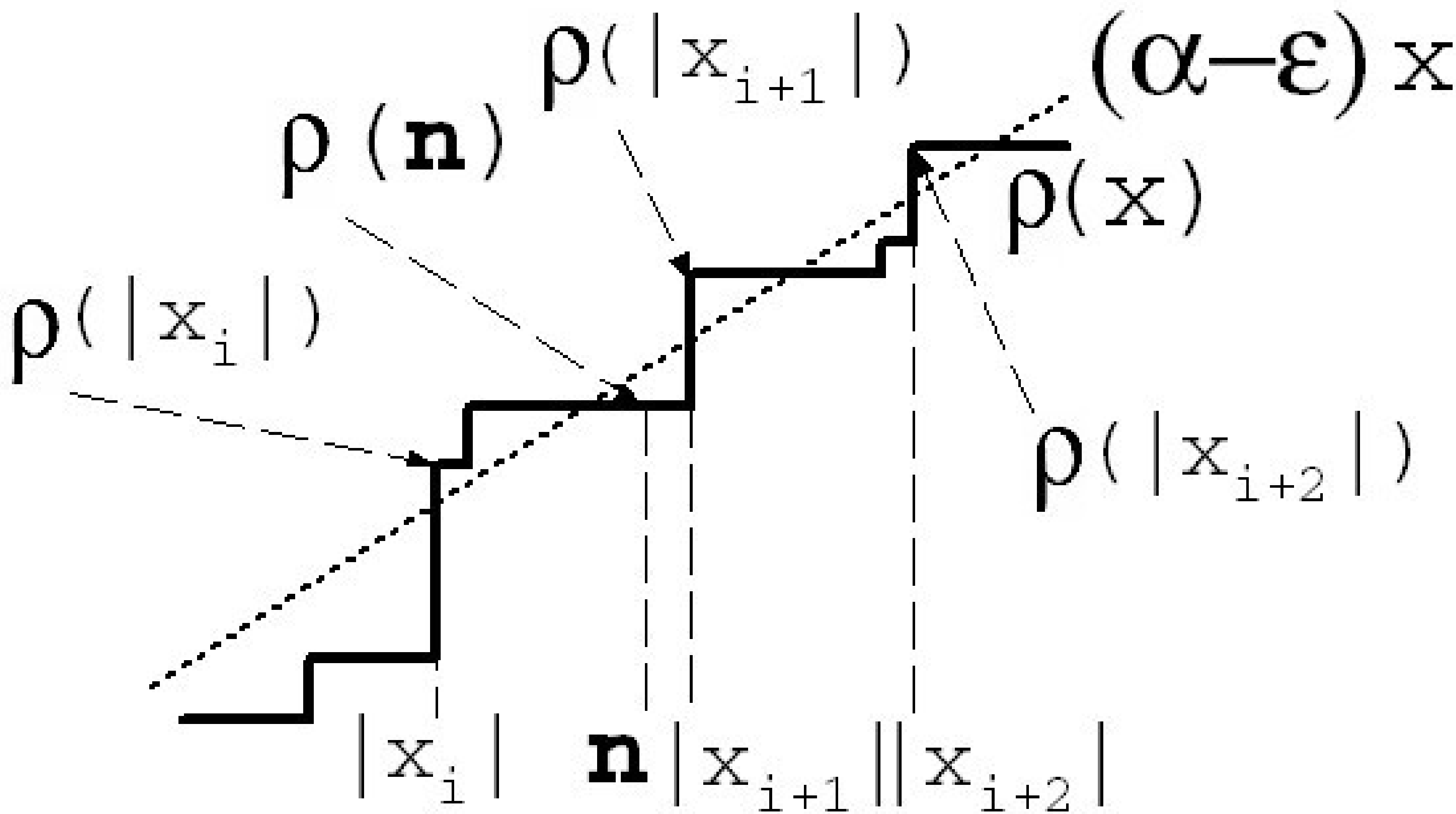
$$\lim_{n \to \infty} \frac{R(\boldsymbol{x}_n)}{|\boldsymbol{x}_n|} = \alpha$$

How quickly it converges?

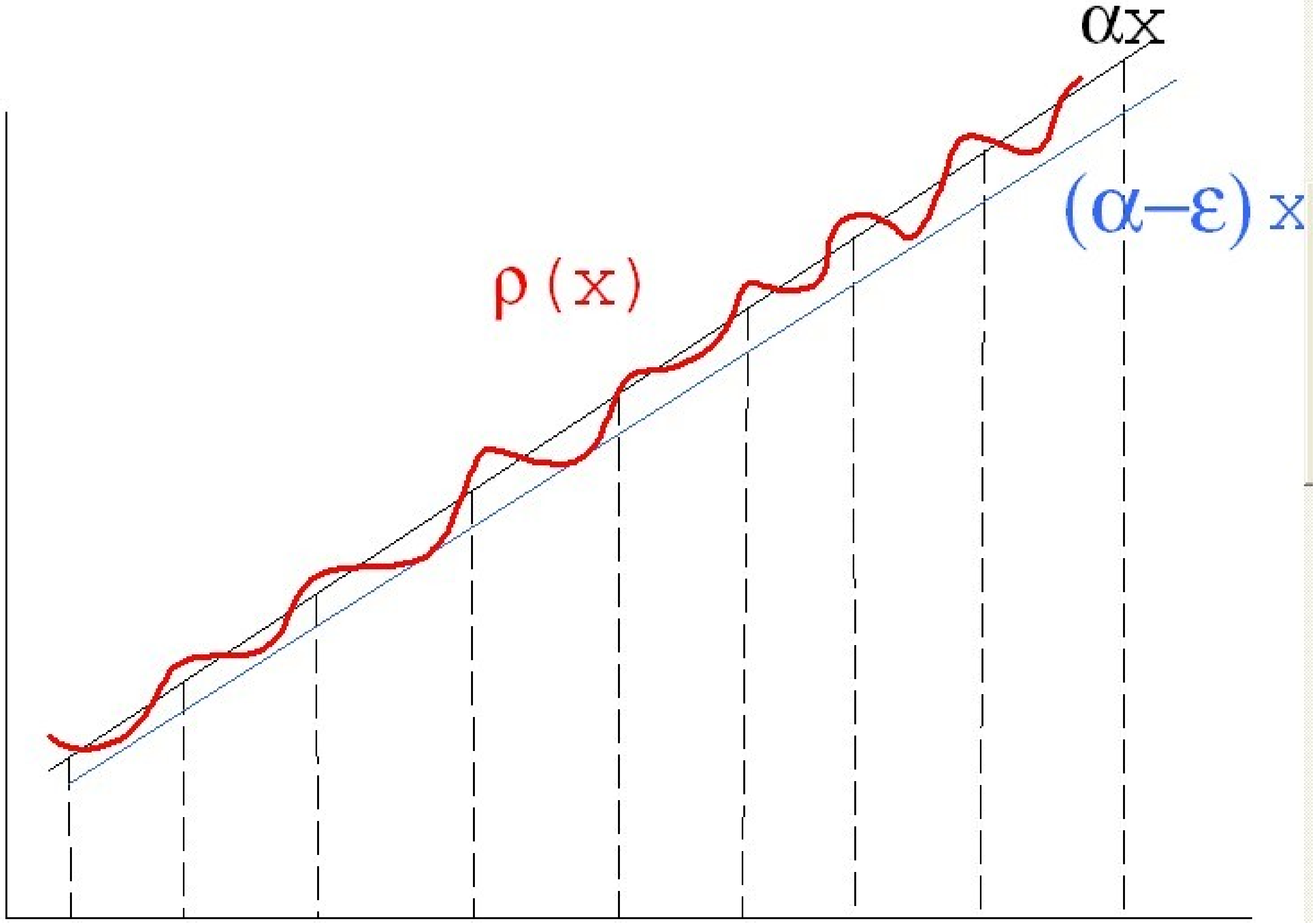
i	$\lvert x_i \rvert$	$R(x_i)$	$\lambda(x_i)$	$\frac{R(x_i)}{\lvert x_i \rvert}$
0	1	0	0	0
1	4	1	0	0.25
2	16	11	3	0.6875
3	67	56	15	0.8358
4	283	254	66	0.8975
5	1198	1100	282	0.9182
6	5074	4691	1197	0.9245
7	21493	19910	5073	0.9263

Franek-Yang method converges faster than *Franek-Simpson-Smyth*, however to the same limit.

Such a sequence is not enough to establish a lower bound, not even an asymptotic lower bound:


ρ(|x_{i+1}|)
(α−ε)x
ρ(n)
ρ(x)
ρ(|x_i|)
ρ(|x_{i+2}|)
|x_i|
n
|x_{i+1}|
|x_{i+2}|

The strategy -- let us put in several sequences, so that the distances between two points on the x-axis are small enough (depending on a given ε) so that $\rho(x)$ does not dip below $(\alpha-\varepsilon)x$.

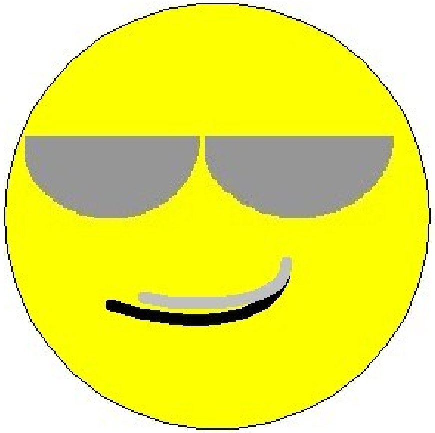

αx
(α−ε) x
ρ (x)

The proof is technical and requires that the size of the strings during the recursive construction is divisible by certain parameters. That requires small modifications of the presented constructions and a careful selection of a finite number of "starting" strings.

This research is a result of an effort to settle the conjectures C1-C3. This effort has not been completed yet, as none of the conjectures has been settled. If the upper bound for $\rho()$ can really be pushed as low as 1.5n, then we do know something about $\rho()$:

$$0.92n \le \rho(n) \le 1.6n$$

So, the future research will continue with attempts to settle the conjectures. Among the conjectures, C3 is the most interesting, for it is the only one that describes structural properties of run-maximal strings. This is the ultimate goal -- to describe structurally run-maximal strings and method how to generate them (could be very useful for testing of many algorithms)



http://www.cas.mcmaster/~franek