Towards a solution to the "runs" conjecture

MAXIME CROCHEMORE[1], LUCIAN ILIE[2], LIVIU TINTA[2]

[1]King's College London, UK

[2]University of Western Ontario, CANADA

Repetitions

- repetition = substring with adjacent occurrences
- square – two occurrences
  - ATAGGATAGG $= (\text{ATAGG})^2$
  - hotshots $= (\text{hots})^2$
- general (fractional) repetitions
  - alfalfa $= (\text{alf})^{\frac{7}{3}}$
  - otavaltavaltavaltavaltiolta $= \text{o}(\text{taval})^{\frac{21}{5}}\text{iolta}$
  - CGATATATATATATACGGC $= \text{CG}(\text{AT})^{\frac{13}{2}}\text{CGGC}$
- for a repetition w^α
  - α is the exponent
  - the number of letters of w is the period

Repetitions – Problems

- interesting combinatorial problems
  - count the number of various types of repetitions
  - squares already investigated by [Thue, 1906, 1912]
- useful combinatorial algorithms
  - find various types of repetitions
  - find all repetitions in linear time
- many applications
  - text algorithms
  - data compression
  - analysis of biological sequences

Finding all repetitions

- Goal: find all repetitions in linear time
  - problem – too many repetitions
  - solution – encode all more compactly
- [Crochemore, 1981, 1983] – linear for one square, $\mathcal{O}(n \log n)$ for all primitively-rooted maximal integer rep.
- [Apostolico, Preparata, 1983] – $\mathcal{O}(n \log n)$ for all right-maximal
- [Main, Lorentz, 1985] – $\mathcal{O}(n \log n)$ for all maximal
- [Main, 1989] – linear for all maximal leftmost
- [Iliopoulos, Moore, Smyth, 1997] – linear for all in Fibonacci

Runs

- run – repetition that is
  - fractional
  - non-extendable (maximal)
  - primitively-rooted
    - $\mathsf{otavaltavaltavaltavaltiolta} = \mathsf{o(taval)}^{\frac{21}{5}}\mathsf{iolta}$
    - $\mathsf{CGATATATATATATACGGC} = \mathsf{CG(AT)}^{\frac{13}{2}}\mathsf{CGGC}$
    - $00101010000010 = 00(01)^{\frac{5}{2}}00010$
    - $00010101000010 = 0^3 1010^4 10$

Runs – linear bound

- all repetitions are encoded in runs
- number of runs in a string of length n
  - $\mathcal{O}(n)$ – [Kolpakov, Kucherov, 1998]
  - compute all repetitions in linear time
    - modified Main's algorithm
    - based on Lempel–Ziv factorization – LZ77
    - most important breakthrough – the bound
  - no constant could be derived from the proof
- "runs" conjecture
  - $\text{runs}(n) \leq n$ – supported by computations
- [Rytter, 2006] – first bound: $5n$

Runs – simple proof

- Kolpakov and Kucherov's proof – very complicated
  - 8 (large) pages
  - contains case 2.1.2.2
- Rytter's proof – 9 pages
  - highly and weakly periodic runs
- [Crochemore, Ilie, 2006] – simple proof
  - 1.5 pages
  - improved Rytter's idea of "neighbor runs"

Runs – sum of exponents

- $\mathcal{O}(n)$ – [Kolpakov, Kucherov, 1999]
  - sum of exponents – applications to analysis of algorithms
  - proof very complicated – 9 (large) pages, contains case 2.1.2.3.2
- [Crochemore, Ilie, 2006] – simple proof – 0.5 pages

Number of runs – upper bounds

- analysis of any algorithm computing all repetitions
- upper bounds
  - [Rytter, 2006] – $5n$
  - [Puglisi, Simpson, Smyth, 2006] – $3.48n$
  - [Rytter, 2006] – $3.44n$
  - [Crochemore, Ilie, 2006] – $1.6n$

Sum of exponents – upper bounds

- conjecture
  - [Kolpakov, Kucherov, 1999] – $\leq 2n$
- [Crochemore, Ilie, 2006] – $5.6n$
  - could be improved by computer verification to $2.9n$ or lower
  - the first explicit bound for the sum of exponents
  - from Rytter's paper – $25n$ – "unsatisfactory"

Proof ideas

- [Rytter, 2006]
  - runs counted at the start – logarithmically many
  - short period runs
  - long period runs
    - weakly periodic
    - highly periodic

abaababaabaababaababaabaababaabaa

Proof ideas

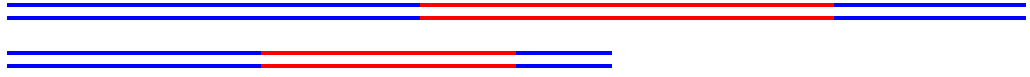
- [Rytter, 2006]
  - runs counted at the start – logarithmically many
  - short period runs
  - long period runs
    - weakly periodic
    - highly periodic

abaababaabaababaababaabaababaabaa

Proof ideas

- [Rytter, 2006]
  - runs counted at the start – logarithmically many
  - short period runs
  - long period runs
    - weakly periodic
    - highly periodic

abaababaabaababaababaabaababaabaa



Proof ideas

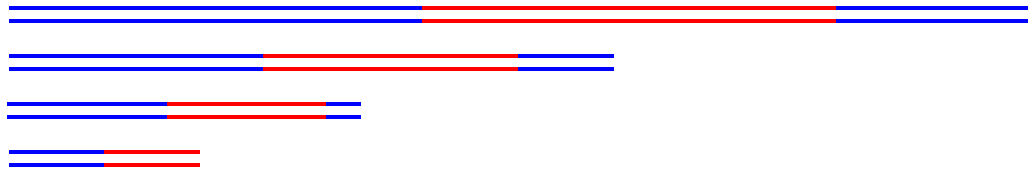
- [Rytter, 2006]
  - runs counted at the start – logarithmically many
  - short period runs
  - long period runs
    - weakly periodic
    - highly periodic

abaababaabaababaababaabaababaabaa

Proof ideas

- [Rytter, 2006]
  - runs counted at the start – logarithmically many
  - short period runs
  - long period runs
    - weakly periodic
    - highly periodic

abaababaabaababaababaabaababaabaa



Proof ideas (2)

- [Crochemore, Ilie, 2006]
  - runs counted at the center – linearly many
    - center = beginning of second period
  - short period runs (microruns)
  - long period runs – 10 times better bound
  - for periods ≥ 87
    - we have $0.06897n$ compared to $0.67n$

abaabaabaabaabaababaabaabaabaabaaba

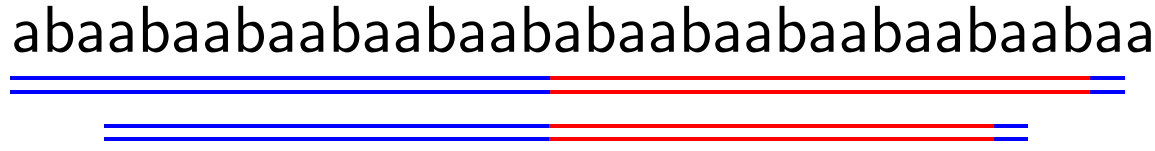
Proof ideas (2)

- [Crochemore, Ilie, 2006]
  - runs counted at the center – linearly many
    - center = beginning of second period
  - short period runs (microruns)
  - long period runs – 10 times better bound
  - for periods ≥ 87
    - we have $0.06897n$ compared to $0.67n$

abaabaabaabaabaababaabaabaabaabaaba

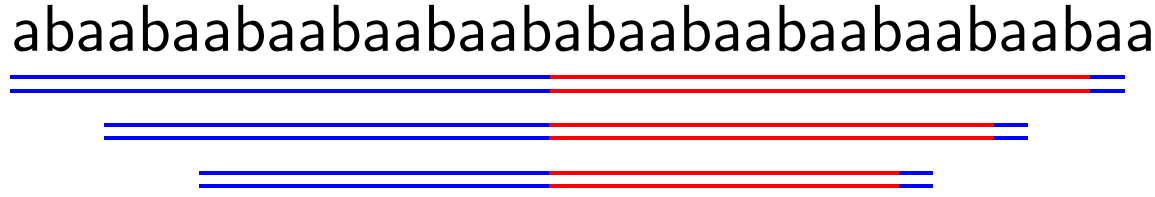
Proof ideas (2)

- [Crochemore, Ilie, 2006]
  - runs counted at the center – linearly many
    - center = beginning of second period
  - short period runs (microruns)
  - long period runs – 10 times better bound
  - for periods ≥ 87
    - we have $0.06897n$ compared to $0.67n$



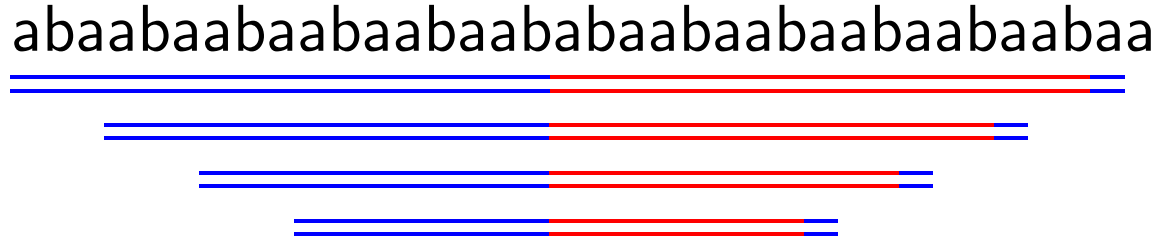
Proof ideas (2)

- [Crochemore, Ilie, 2006]
  - runs counted at the center – linearly many
    - center = beginning of second period
  - short period runs (microruns)
  - long period runs – 10 times better bound
  - for periods ≥ 87
    - we have $0.06897n$ compared to $0.67n$



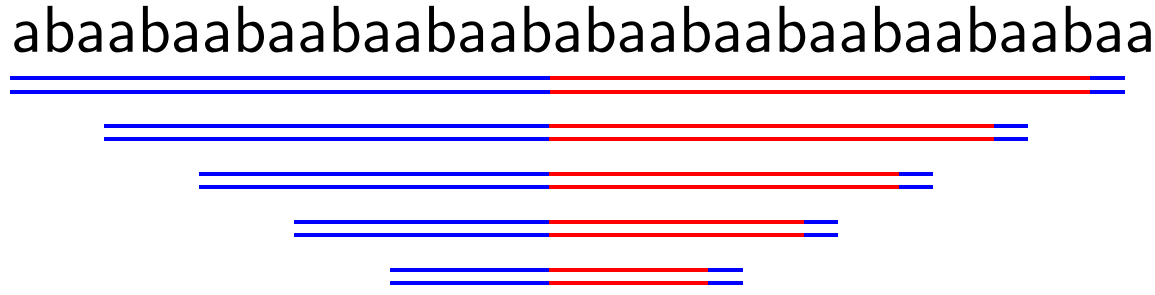
Proof ideas (2)

- [Crochemore, Ilie, 2006]
  - runs counted at the center – linearly many
    - center = beginning of second period
  - short period runs (microruns)
  - long period runs – 10 times better bound
  - for periods ≥ 87
    - we have $0.06897n$ compared to $0.67n$



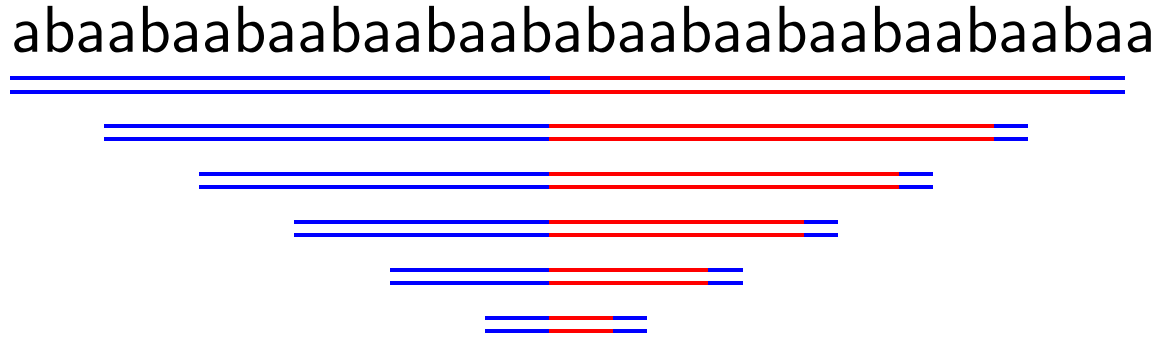
Proof ideas (2)

- [Crochemore, Ilie, 2006]
  - runs counted at the center – linearly many
    - center = beginning of second period
  - short period runs (microruns)
  - long period runs – 10 times better bound
  - for periods ≥ 87
    - we have $0.06897n$ compared to $0.67n$



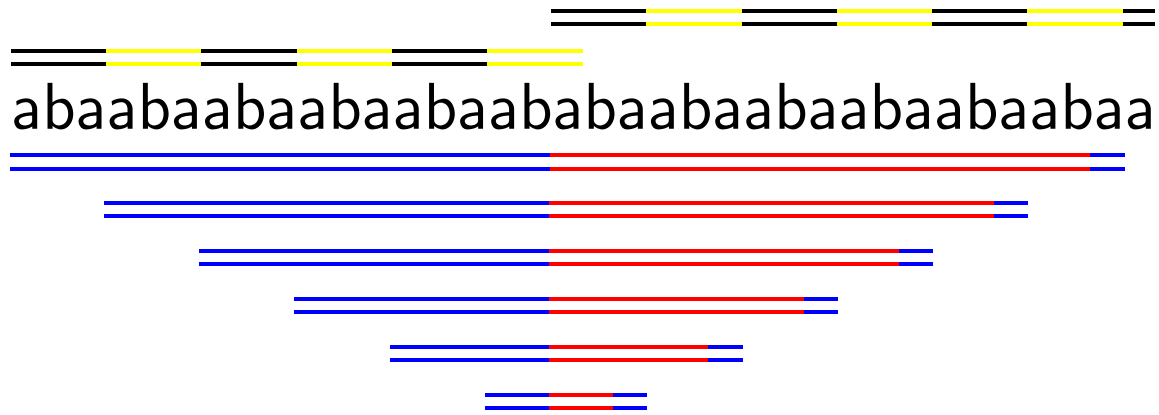
Proof ideas (2)

- [Crochemore, Ilie, 2006]
  - runs counted at the center – linearly many
    - center = beginning of second period
  - short period runs (microruns)
  - long period runs – 10 times better bound
  - for periods ≥ 87
    - we have $0.06897n$ compared to $0.67n$



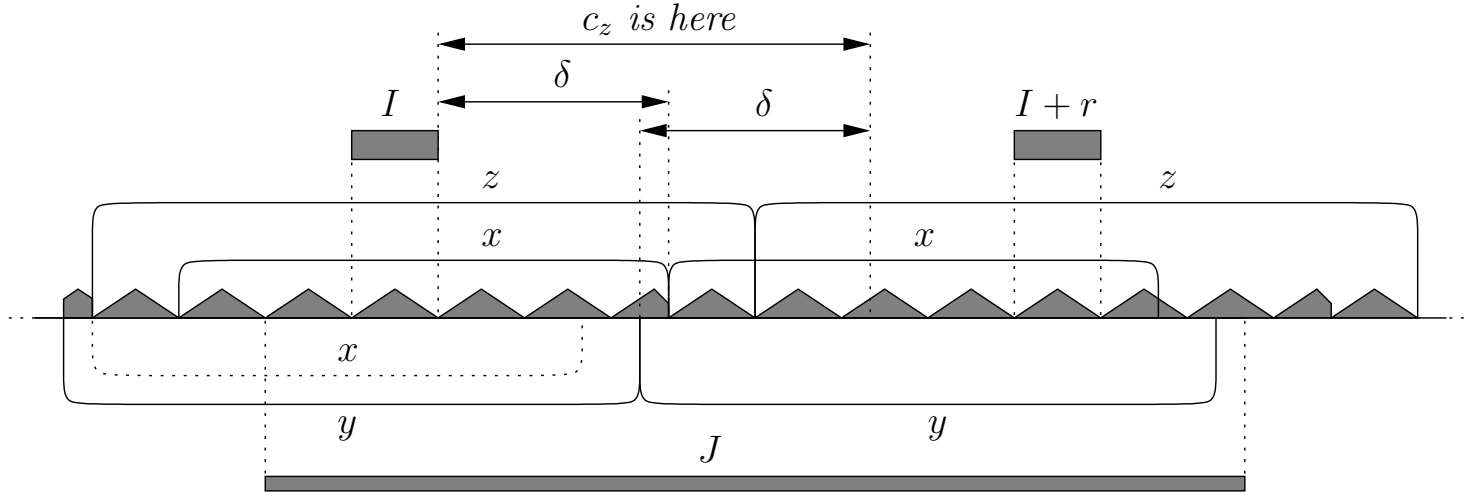
Proof ideas (2)

- [Crochemore, Ilie, 2006]
  - runs counted at the center – linearly many
    - center = beginning of second period
  - short period runs (microruns)
  - long period runs – 10 times better bound
  - for periods ≥ 87
    - we have $0.06897n$ compared to $0.67n$



Short periods need separate reasoning

- runs with close centers imply high local periodicity
- for short periods it is not possible

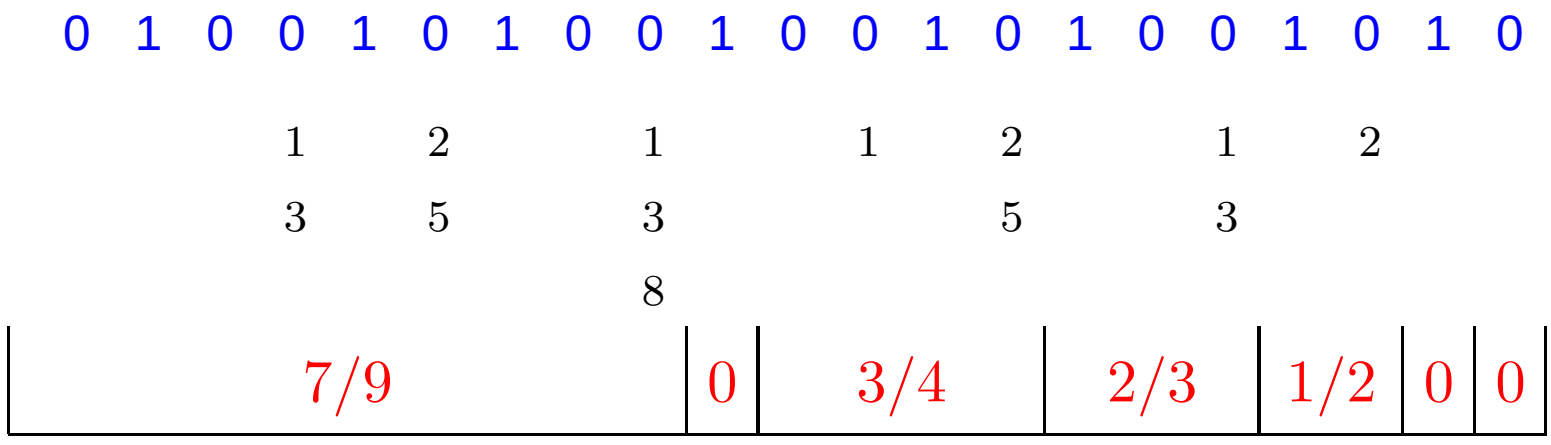


Long and short periods

- [Crochemore, Ilie, 2006]
- runs with period p or larger
  - **Theorem 1** $\boxed{\text{runs}_{\geqslant p}(n) \leq \frac{6}{p}n}$
- runs with period p or smaller (microruns)
  - **Theorem 2** $\boxed{\text{runs}_{\leqslant p}(n) \leq bn}$
- **Bound:** $\text{runs}(n) \leq (\frac{6}{p+1} + b)n$
  - $p = 9$, $b = 1$ gives $\text{runs}(n) \leq 1.6n$
  - higher p and/or lower b give better bounds

Microruns – better upper bounds

- centers of microruns – not uniformly distributed
- amortized for large enough intervals
- example: $p = 8$, $b = 0.8$

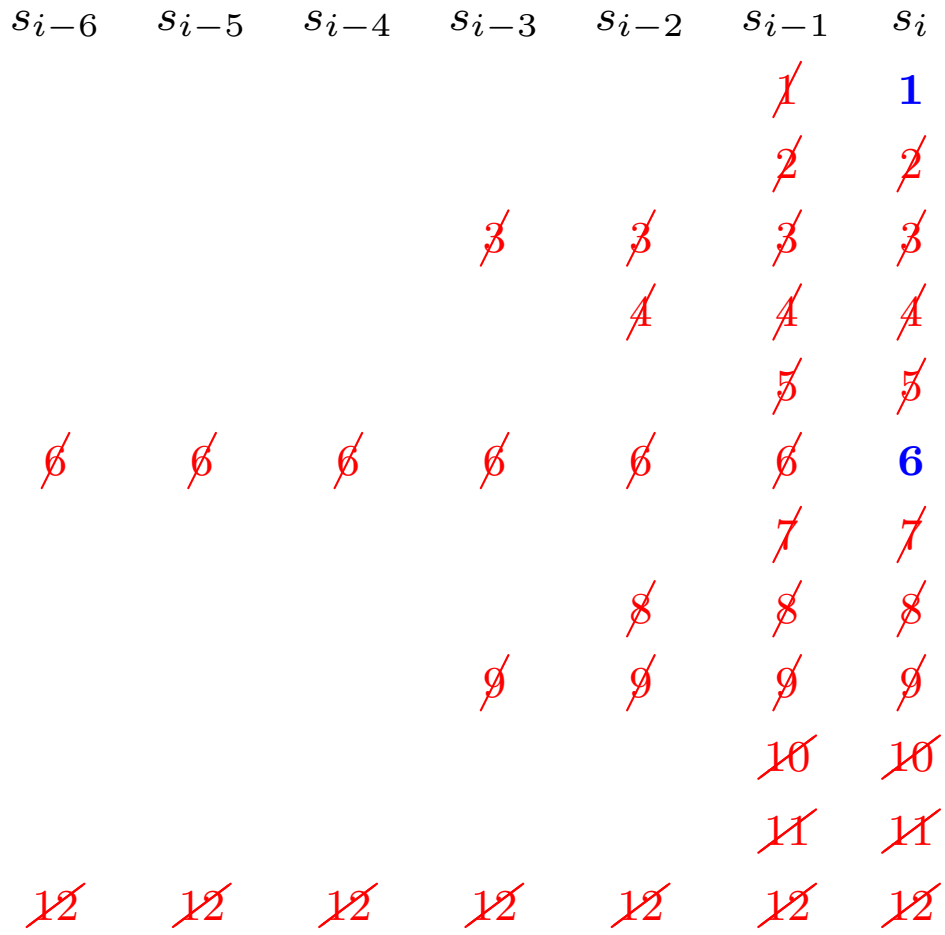


Microruns – better upper bounds (2)

- given p and b, try all possibilities
  - if all are amortized, then $\text{runs}_{\leqslant p}(n) \leq bn$
- trying all strings
  - for arbitrary alphabets – not clear how to do it
  - for fixed alphabets – too much to compute
    - 2 letters, length of amortizing interval 100: 2^{100}
- trying all combinations of periods
  - works for arbitrary alphabets
  - too much to compute
    - $p = 100$, length of amortizing interval 100: 2^{10000}
  - solution: try only the possible combinations

Microruns – better upper bounds (3)

- example: if $\{1, 6\}$ at some position i in a string s and $p = 12$, then some periods are forbidden



Microruns – better upper bounds (4)

- all possible sets of periods at the same position for $p = 9$
  - $\emptyset$,
    $\{1\}, \ldots, \{9\}$,
    $\{1,3\}, \ldots, \{1,9\}, \{2,5\}, \ldots, \{2,9\}, \{3,7\}, \ldots \{3,9\}, \{4,9\}, \{5,8\}$,
    $\{1,3,5\}, \{1,3,7\}, \ldots, \{1,3,9\}, \{1,4,7\}, \{1,4,9\}, \{1,5,9\}, \{2,5,8\}$,
    $\{1,3,5,7\}$,
    $\{1,3,5,7,9\}$
- there are 37 sets out of potential $2^9 = 512$

Current best bound

- obtained using SHARCNET – `www.sharcnet.ca`

p	b	solutions	amortize	runs$(n) \leqslant$	CPU time (s)
10	0.85	900	100	$\mathbf{1.396n}$	40
15	0.89	5275	27	$\mathbf{1.265n}$	600
20	0.89	34833	97	$\mathbf{1.176n}$	1920
25	0.91	135457	153	$\mathbf{1.141n}$	8640
30	0.91	471339	153	$\mathbf{1.104n}$	50,400
35	0.93	1455422	82	$\mathbf{1.097n}$	230,400
40	0.93	3907110	84	$\mathbf{1.077n}$	1,130,400
50	0.93	22635894	139	$\mathbf{1.048n}$	27,388,800

Number of runs – lower bounds

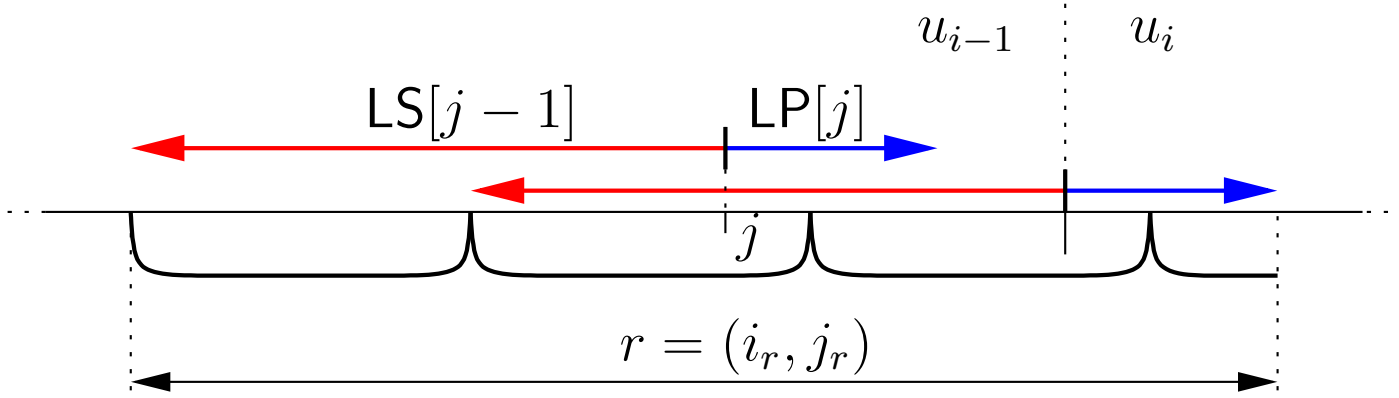
- [Franek, Simpson, Smyth, 2003]
  - $\frac{3}{2\phi}n = 0.927..n$
  - conjectured optimal
- [Kusano, Matsubara, Ishino, Bannai, Shinohara, 2008]
  - $0.944542n$

Further research

- $0.944n \leq \text{runs}(n) \leq 1.048n$
- good enough for practical purposes
- the optimal value seems to be below n
  - (the "runs" conjecture is probably true)
- average number of runs
  - [Puglisi, Simpson, 2008] – also linear
- algorithms for computing runs
  - only one idea – use Lempel–Ziv factorization
  - (same for linear on average)
  - different (simpler) algorithms
  - could bring new insight into the structure of strings

Linear-time algorithm idea

- Lempel–Ziv factorization – a.b.b.a.abb.baa.ab.ab
- runs that cross an LZ-factorization point



- those which do not are only translated