# Repetitions in infinite rich words

## A computational approach

**Aseem Baranwal**, Jeffrey Shallit

School of Computer Science
University of Waterloo
Canada

Email : aseemrb@gmail.com
Web : https://aseemrb.me

# Overview

## Rich words

- A finite word $w$ is palindrome-rich, or simply *rich* if it contains $|w|$ nonempty distinct palindromic factors.
- An infinite word is *rich* if all of its factors are rich.

## Examples

- The word **00010110** is rich – contains 8 distinct nonempty palindromes: 0, 00, 000, 1, 010, 101, 11, 0110.
- The word **00101100** is not rich – only 7 distinct palindromes.

# Repetition Threshold

- Repetition threshold can be studied for a limited class of infinite words – e.g., Rampersad et al. considered the class of balanced words.
- We study the class of infinite palindrome-rich words.

# Repetition Threshold

- Repetition threshold can be studied for a limited class of infinite words – e.g., Rampersad et al. considered the class of balanced words.
- We study the class of infinite palindrome-rich words.

**Problem**

What is the repetition threshold for infinite rich words over a $k$-letter alphabet?

Known results

## Known results

- Pelantová and Starosta (2013) showed that every infinite rich word contains a square, implying that the repetition threshold is greater than 2.

## Known results

- Pelantová and Starosta (2013) showed that every infinite rich word contains a square, implying that the repetition threshold is greater than 2.
- Vesti gave upper and lower bounds on the length of the longest square-free rich words (2017). He also proposed the open problem of determining the repetition threshold.

## Known results

- Pelantová and Starosta (2013) showed that every infinite rich word contains a square, implying that the repetition threshold is greater than 2.
- Vesti gave upper and lower bounds on the length of the longest square-free rich words (2017). He also proposed the open problem of determining the repetition threshold.

## Our contribution

## Known results

- Pelantová and Starosta (2013) showed that every infinite rich word contains a square, implying that the repetition threshold is greater than 2.
- Vesti gave upper and lower bounds on the length of the longest square-free rich words (2017). He also proposed the open problem of determining the repetition threshold.

## Our contribution

- We make the first progress on Vesti's problem by constructing a binary word that achieves the repetition threshold.

## Known results

- Pelantová and Starosta (2013) showed that every infinite rich word contains a square, implying that the repetition threshold is greater than 2.
- Vesti gave upper and lower bounds on the length of the longest square-free rich words (2017). He also proposed the open problem of determining the repetition threshold.

## Our contribution

- We make the first progress on Vesti's problem by constructing a binary word that achieves the repetition threshold.
- Our approach is automated and computation-based (`Walnut`).

## Morphisms

$$\phi: \begin{array}{l} 0 \mapsto 01 \\ 1 \mapsto 02 \\ 2 \mapsto 022 \end{array} \qquad \tau: \begin{array}{l} 0 \mapsto 0 \\ 1 \mapsto 01 \\ 2 \mapsto 011 \end{array}$$

## Theorem

The infinite word $\mathbf{r} = \tau(\phi^\omega(0)) = 00100110010\ldots$ is rich, and has the critical exponent $2 + \frac{\sqrt{2}}{2}$.

### Morphisms

$$\phi: \begin{array}{l} 0 \mapsto 01 \\ 1 \mapsto 02 \\ 2 \mapsto 022 \end{array} \qquad \tau: \begin{array}{l} 0 \mapsto 0 \\ 1 \mapsto 01 \\ 2 \mapsto 011 \end{array}$$

### Theorem

The infinite word $\mathbf{r} = \tau(\phi^\omega(0)) = 00100110010\ldots$ is rich, and has the critical exponent $2 + \frac{\sqrt{2}}{2}$.

- We conjectured this exponent to be the repetition threshold among the class of infinite rich words over $\Sigma_2 = \{0, 1\}$.
- Currie et al. have resolved our conjecture.

**Observation**

The lengths $L_i = |\tau(\phi^i(0))|$ follow the recurrence relation:
$L_0 = 1$, $L_1 = 3$, and $L_i = 2L_{i-1} + L_{i-2}$ for all $i \geq 2$.

This suggests that the word **r** might be Pell-automatic.

**Observation**

The lengths $L_i = |\tau(\phi^i(0))|$ follow the recurrence relation:
$L_0 = 1$, $L_1 = 3$, and $L_i = 2L_{i-1} + L_{i-2}$ for all $i \geq 2$.

This suggests that the word **r** might be Pell-automatic.

- We require an automaton producing the word **r** for `Walnut` to understand predicates involving **r**.
- We use the adder automaton given by Baranwal and Shallit to work in Walnut with predicates involving the word.

- We construct the automaton producing the word **r** using the L* algorithm by Angluin [1] to learn regular sets with queries.
- An induction based proof verifies that this automaton produces the same word as $\tau(\phi^{\omega}(0))$.

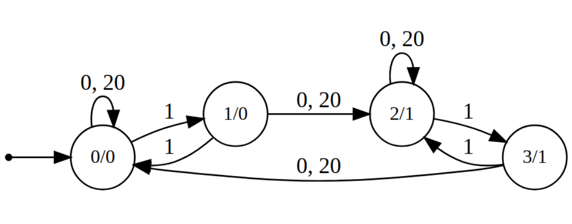



**Figure 1:** Automaton for the infinite word **r**.

**Notation**

**R** denotes the automaton for the word **r** indexed from 0.

**Notation**

R denotes the automaton for the word **r** indexed from 0.

**Procedure overview**

1. Construct automata for a set of fundamental predicates.

**Notation**

R denotes the automaton for the word **r** indexed from 0.

**Procedure overview**

1. Construct automata for a set of fundamental predicates.
2. Use them to construct a predicate for palindromic richness.

# Proving palindromic richness

**Notation**

R denotes the automaton for the word **r** indexed from 0.

**Procedure overview**

1. Construct automata for a set of fundamental predicates.
2. Use them to construct a predicate for palindromic richness.
3. Show that the predicate is true for all inputs.

Length-$n$ factors of R starting at indices $i$ and $j$ are equal.

**FactorEq**$(i, j, n)$

$\forall k \ (k < n) \implies \mathsf{R}[i + k] = \mathsf{R}[j + k]$

Length-$n$ factors of R starting at indices $i$ and $j$ are equal.

| **`FactorEq`**$(i, j, n)$ |
|---|
| $\forall k\ (k < n) \implies \mathrm{R}[i + k] = \mathrm{R}[j + k]$ |

Length-$n$ factor of R starting at index $i$ is a palindrome.

| **`Palindrome`**$(i, n)$ |
|---|
| $\forall j\ \forall k\ (k < n) \implies \mathrm{R}[i + k] = \mathrm{R}[n - 1 - k]$ |

Length-$n$ factors of R starting at indices $i$ and $j$ are equal.

| **FactorEq**$(i, j, n)$ |
|---|
| $\forall k\ (k < n) \implies \mathrm{R}[i + k] = \mathrm{R}[j + k]$ |

Length-$n$ factor of R starting at index $i$ is a palindrome.

| **Palindrome**$(i, n)$ |
|---|
| $\forall j\ \forall k\ (k < n) \implies \mathrm{R}[i + k] = \mathrm{R}[n - 1 - k]$ |

The word $\mathrm{R}[i..i + m - 1]$ occurs in the word $\mathrm{R}[j..j + n - 1]$.

| **Occurs**$(i, j, m, n)$ |
|---|
| $(m \leq n) \wedge (\exists k\ (k + m \leq n) \wedge \texttt{FactorEq}(i, j + k, m))$ |

Length-$n$ factors of R starting at indices $i$ and $j$ are equal.

| $\texttt{FactorEq}(i, j, n)$ | 67 states |
|---|---|
| $\forall k\ (k < n) \implies \mathrm{R}[i + k] = \mathrm{R}[j + k]$ | |

Length-$n$ factor of R starting at index $i$ is a palindrome.

| $\texttt{Palindrome}(i, n)$ | 27 states |
|---|---|
| $\forall j\ \forall k\ (k < n) \implies \mathrm{R}[i + k] = \mathrm{R}[n - 1 - k]$ | |

The word $\mathrm{R}[i..i + m - 1]$ occurs in the word $\mathrm{R}[j..j + n - 1]$.

| $\texttt{Occurs}(i, j, m, n)$ | 1715 states |
|---|---|
| $(m \leq n) \wedge (\exists k\ (k + m \leq n) \wedge \texttt{FactorEq}(i, j + k, m))$ | |

**Fact**

An infinite word is rich if and only if all its factors are rich.
We could also look at only the prefixes instead of all factors.

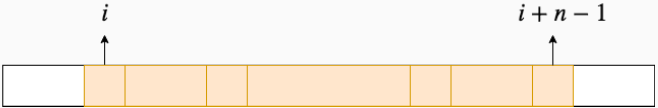
**Fact**

An infinite word is rich if and only if all its factors are rich.
We could also look at only the prefixes instead of all factors.

**Theorem**

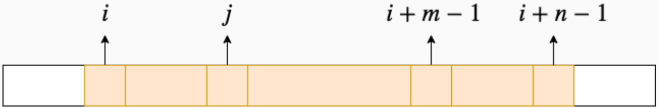
(Glen et al.) A finite word $w$ is rich if and only if every prefix of $w$ has a unioccurrent palindromic suffix.

Figure 2: Constructing the predicate RichFactor$(i, n)$.



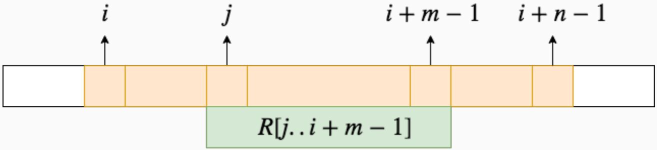

RichFactor$(i, n)$:

**Figure 2:** Constructing the predicate RichFactor($i, n$).



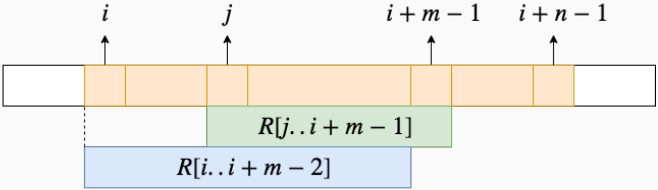

RichFactor($i, n$): $\forall m, (1 \leq m < n) \implies (\exists j, (i \leq j < i + m) \wedge$

**Figure 2:** Constructing the predicate RichFactor$(i, n)$.




RichFactor$(i, n)$: $\forall m, (1 \leq m < n) \implies (\exists j, (i \leq j < i + m) \wedge$
Palindrome$(j, i + m - j)$

**Figure 2:** Constructing the predicate RichFactor$(i, n)$.




RichFactor$(i, n)$: $\forall m, (1 \leq m < n) \implies (\exists j, (i \leq j < i + m) \wedge$ Palindrome$(j, i + m - j) \wedge \neg$Occurs$(j, i, i + m - j, m - 1))$

To determine if **r** is rich, we check if all its prefixes are rich.

**R_is_Rich**

$\forall n$ `RichFactor`$(0, n)$.

- In `Walnut`, this predicate evaluates to true.
- Conclusion – The infinite word **r** is rich.

**Problem**

It is difficult to write a first-order predicate to determine the critical exponent because it is an irrational number.

**Problem**

It is difficult to write a first-order predicate to determine the critical exponent because it is an irrational number.

**Solution**

Compute the critical exponent as the limit of a monotonic expression for exponents.

**Problem**

It is difficult to write a first-order predicate to determine the critical exponent because it is an irrational number.

**Solution**

Compute the critical exponent as the limit of a monotonic expression for exponents.

Overview

- Compute periods of high powers ($\geq 5/2$).
- Compute the maximal lengths associated with the high-power periods above.

### Computing the critical exponent

First, compute periods $p$ such that the word $\mathbf{r}$ has factors with period $p$ and exponent $> 5/2$.

**HighPowerPeriods**($p$)

$(p \geq 1) \wedge (\exists i \forall j \; (2j \leq 3p) \implies \mathrm{R}[i+j] = \mathrm{R}[i+j+p])$.

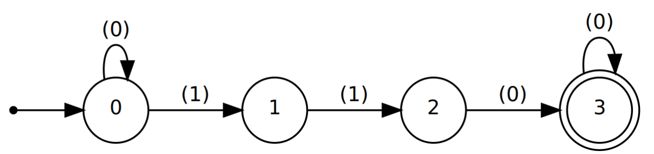



**Figure 3:** Automaton for the predicate HighPowerPeriods.

Compute pairs $(n, p)$ such that **r** has a factor of length $n + p$ with period $p$, which cannot be extended with the same period.

| **MaximalReps**$(n, p)$ | 17 states |
| --- | --- |
| $\exists i (\forall j \ (j < n) \implies \mathrm{R}[i + j] = \mathrm{R}[i + j + p]) \wedge (\mathrm{R}[i + n] \neq \mathrm{R}[i + n + p])$. | |

Compute pairs $(n, p)$ such that $\mathbf{r}$ has a factor of length $n + p$ with period $p$, which cannot be extended with the same period.

**MaximalReps**$(n, p)$ 17 states

$$\exists i (\forall j \ (j < n) \implies \mathrm{R}[i+j] = \mathrm{R}[i+j+p]) \land (\mathrm{R}[i+n] \neq \mathrm{R}[i+n+p]).$$

Compute pairs $(n, p)$ where $p$ satisfies HighPowerPeriods and $n + p$ is the longest length of any factor with period $p$.

**HighestPowers**$(n, p)$

$$\begin{aligned} &\texttt{HighPowerPeriods}(p) \land \\ &\texttt{MaximalReps}(n, p) \land \\ &(\forall m \ \texttt{MaximalReps}(m, p) \implies m \leq n). \end{aligned}$$

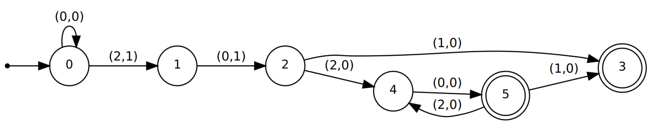



**Figure 4:** Automaton for the predicate `HighestPowers`.

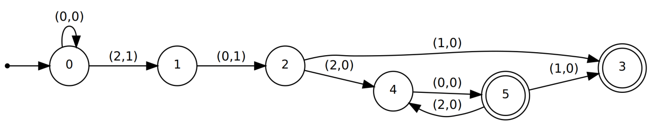



**Figure 4:** Automaton for the predicate `HighestPowers`.

This automaton accepts pairs $(n, p)$ of the following formats:

1. $\binom{0}{0}^* \binom{2}{1} \binom{0}{1} \binom{1}{0}$,
2. $\binom{0}{0}^* \binom{2}{1} \binom{0}{1} \binom{2}{0} \binom{0}{0} \{ \binom{2}{0} \binom{0}{0} \}^*$,
3. $\binom{0}{0}^* \binom{2}{1} \binom{0}{1} \binom{2}{0} \binom{0}{0} \{ \binom{2}{0} \binom{0}{0} \}^* \binom{1}{0}$.

- **Case 1** corresponds to: $n = (201)_P = 11$ and $p = (110)_P = 7$.

$$e = \frac{n+p}{p} = \frac{18}{7} \approx 2.57.$$

- **Case 2** corresponds to:

$$n = 2 \sum_{1 \le i \le k} P_{2k} = P_{2k+1} - 1, \quad p = P_{2k} + P_{2k-1}$$

- **Case 3** corresponds to:

$$n = 1 + 2 \sum_{1 \le i \le k} P_{2k+1} = P_{2k+2} - 1, \quad p = P_{2k+1} + P_{2k}$$

Substituting $m = 2k - 1$ in case 2, and $m = 2k$ in case 3, we notice that the expressions for the exponent coincide:

$$\begin{aligned} e &= \frac{P_{m+2} + P_{m+1} + P_m - 1}{P_{m+1} + P_m} \\ &= 2 + \frac{P_{m+1} - 1}{P_{m+1} + P_m}. \end{aligned}$$

This expression is increasing with $m$, and tends to $2 + \sqrt{2}/2$ as $m \to \infty$. Thus, the critical exponent,

$$E(\mathbf{r}) = 2 + \frac{\sqrt{2}}{2}.$$

- With backtracking search, we had found that

$$2.700 \leq RRT(2) \leq 2 + \frac{\sqrt{2}}{2} \doteq 2.707,$$

  and conjectured that the upper bound is exact.
- Used the data structure EERTREE given by Rubinchik and Shur [5] to efficiently verify richness.

- With backtracking search, we had found that

$$2.700 \leq RRT(2) \leq 2 + \frac{\sqrt{2}}{2} \doteq 2.707,$$

  and conjectured that the upper bound is exact.
- Used the data structure EERTREE given by Rubinchik and Shur [5] to efficiently verify richness.
- Recently, Currie, Mol, and Rampersad have resolved our conjecture.

**Morphisms $f$, $g$ and $h$**

$$f: \begin{array}{l} 0 \mapsto 0 \\ 1 \mapsto 01 \\ 2 \mapsto 011 \end{array} \qquad g: \begin{array}{l} 0 \mapsto 011 \\ 1 \mapsto 0121 \\ 2 \mapsto 012121 \end{array} \qquad h: \begin{array}{l} 0 \mapsto 01 \\ 1 \mapsto 02 \\ 2 \mapsto 022 \end{array}$$

**Lemma (Currie et al.)**

The critical exponent of $f(g(h^\omega(0)))$ is at least $2 + \sqrt{2}/2$.

**Theorem (Currie, Mol, Rampersad)**

The repetition threshold for infinite rich words over the alphabet $\Sigma_2 = \{0, 1\}$ is $2 + \sqrt{2}/2$.

**Proof sketch**

**Theorem (Currie, Mol, Rampersad)**

The repetition threshold for infinite rich words over the alphabet $\Sigma_2 = \{0, 1\}$ is $2 + \sqrt{2}/2$.

**Proof sketch**

- If $w \in \Sigma_2^\omega$ is an infinite rich word avoiding powers $\geq 14/5$ then

**Theorem (Currie, Mol, Rampersad)**

The repetition threshold for infinite rich words over the alphabet $\Sigma_2 = \{0, 1\}$ is $2 + \sqrt{2}/2$.

**Proof sketch**

- If $w \in \Sigma_2^\omega$ is an infinite rich word avoiding powers $\geq 14/5$ then
  - $w$ contains all factors of $f(h^\omega(0))$, or

# Proof by Currie et al.

**Theorem (Currie, Mol, Rampersad)**

The repetition threshold for infinite rich words over the alphabet $\Sigma_2 = \{0, 1\}$ is $2 + \sqrt{2}/2$.

**Proof sketch**

- If $w \in \Sigma_2^\omega$ is an infinite rich word avoiding powers $\geq 14/5$ then
  - $w$ contains all factors of $f(h^\omega(0))$, or
  - $w$ contains all factors of $f(g(h^\omega(0)))$.

**Theorem (Currie, Mol, Rampersad)**

The repetition threshold for infinite rich words over the alphabet $\Sigma_2 = \{0, 1\}$ is $2 + \sqrt{2}/2$.

**Proof sketch**

- If $w \in \Sigma_2^\omega$ is an infinite rich word avoiding powers $\geq 14/5$ then
  - $w$ contains all factors of $f(h^\omega(0))$, or
  - $w$ contains all factors of $f(g(h^\omega(0)))$.
- Our word $\mathbf{r} = f(h^\omega(0))$ has critical exponent $2 + \sqrt{2}/2$.

# Proof by Currie et al.

**Theorem (Currie, Mol, Rampersad)**

The repetition threshold for infinite rich words over the alphabet $\Sigma_2 = \{0, 1\}$ is $2 + \sqrt{2}/2$.

**Proof sketch**

- If $w \in \Sigma_2^\omega$ is an infinite rich word avoiding powers $\geq 14/5$ then
  - $w$ contains all factors of $f(h^\omega(0))$, or
  - $w$ contains all factors of $f(g(h^\omega(0)))$.
- Our word $\mathbf{r} = f(h^\omega(0))$ has critical exponent $2 + \sqrt{2}/2$.
- By Lemma, the critical exponent of $f(g(h^\omega(0)))$ is $\geq 2 + \sqrt{2}/2$.

# Proof by Currie et al.

**Theorem (Currie, Mol, Rampersad)**

The repetition threshold for infinite rich words over the alphabet $\Sigma_2 = \{0, 1\}$ is $2 + \sqrt{2}/2$.

**Proof sketch**

- If $w \in \Sigma_2^\omega$ is an infinite rich word avoiding powers $\geq 14/5$ then
  - $w$ contains all factors of $f(h^\omega(0))$, or
  - $w$ contains all factors of $f(g(h^\omega(0)))$.
- Our word $\mathbf{r} = f(h^\omega(0))$ has critical exponent $2 + \sqrt{2}/2$.
- By Lemma, the critical exponent of $f(g(h^\omega(0)))$ is $\geq 2 + \sqrt{2}/2$.

**The repetition threshold for binary rich words**, James D. Currie, Lucas Mol, Narad Rampersad, Arxiv preprint: https://arxiv.org/abs/1908.03169.

# Future work

**Fact (Edita Pelantová)**

Our word **r** is a complementary symmetric Rote word [4], and hence by the works of Massé, Pelantová and others [2, 3], it follows that **r** is rich.

- Repetition threshold for larger alphabets. Our backtracking search shows that $RRT(3) \geq 9/4$.
- Construct Rote words associated with Sturmian substitutions over larger alphabets ($k$) – check if they achieve $RRT(k)$.

D. Angluin.
**Learning regular sets from queries and counterexamples.**
*Information and Computation*, 75(2):87–106, 1987.

A. Blondin Massé, S. Brlek, S. Labbé, and L. Vuillon.
**Palindromic complexity of codings of rotations.**
*Theoret. Comput. Sci.*, 412:6455–6463, 2011.

E. Pelantová and Š. Starosta.
**Constructions of words rich in palindromes and pseudopalindromes.**
*Discrete Math. & Theoret. Comput. Sci.*, 18:Paper #16, 2016.
Available at `https://dmtcs.episciences.org/2202`.

G. Rote.
**Sequences with subword complexity $2n$.**
*J. Number Theory*, 46:196–213, 1994.

M. Rubinchik and A. M. Shur.
**EERTREE: An efficient data structure for processing palindromes in strings.**
In Z. Lipták and W. F. Smyth, editors, *Combinatorial Algorithms*, pages 321–333, Cham, 2016. Springer International Publishing.

Thank you.