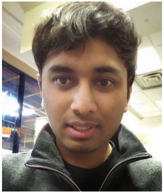
# Determining Repetition Thresholds via Logic and Numeration Systems

Jeffrey Shallit
School of Computer Science
University of Waterloo
Waterloo, ON N2L 3G1
Canada
shallit@uwaterloo.ca
https://cs.uwaterloo.ca/~shallit/

# My co-authors



Aseem Baranwal

James Currie

Lucas Mol

Narad Rampersad

Elise Vandomme

# Repetitions in words

- Repetitions in words: a long-studied topic (since at least 1906) with many applications to other areas of mathematics and computer science
- Most basic forms of repetitions: squares, cubes, overlaps:
  - A *square* is a nonempty word of the form $xx = x^2$, like the German word `nennen`.
  - A word is said to *contain* a square if some factor (contiguous block) is a square. So the German word `Strebausbausteuerung` contains the square `bausbaus`.
  - A *cube* is a nonempty word of the form $xxx = x^3$. The English sort-of-word `shshsh` is a cube.
  - An *overlap* is a word of the form $axaxa$, where $a$ is a single letter and $x$ is a (possibly empty) word. The German word `mehrerer` ends with the overlap `rerer`.
- A word *avoids* squares (or is *squarefree*) if it contains no factor that is a square. So the word `square` is squarefree, but the word `squarefree` is not.

# Thue's work on repetition in words

We say that *squares are avoidable* over an alphabet $\Sigma$ if there exists an infinite squarefree word over $\Sigma$.

Axel Thue (1863–1922)

- It is easy to see that squares are **not** avoidable over $\{0, 1\}$.
  (The longest such word is of length 3.)
- Thue proved in 1906 that squares **are** avoidable over $\{0, 1, 2\}$.
- Similarly, overlaps and cubes are avoidable over $\{0, 1\}$.

# Thue's 1912 construction

- Consider the morphism $\mu$ mapping

$$\begin{aligned} 0 &\to 01 \\ 1 &\to 10 \end{aligned}$$

- We can iterate this morphism, obtaining

$$\begin{aligned} \mu(0) &= 01 \\ \mu^2(0) &= 0110 \\ \mu^3(0) &= 01101001 \\ &\vdots \end{aligned}$$

- In the limit, we get $\mu^\omega(0) = \mathbf{t} = 0110100110010110\cdots$, the infinite *Thue-Morse word*. It is a fixed point of $\mu$.
- This word avoids overlaps.

# Periods, exponents, and more general notions of repetition

- A (finite or infinite) word $w$ has period $p$ if $w[i] = w[i + p]$ for all meaningful $i$.
- Example: the word `alfalfa` has period 3.
- A word can have multiple periods. For example, `aabaaaba` has periods 4 and 7.
- The shortest period is called *the* period.
- The *exponent* of a word is its length divided by its shortest period.
- For example, the German word `schematische` has length 12 and period 8, so its exponent is 3/2.
- Not surprisingly, squares have exponent 2, and cubes have exponent 3.

# Critical exponents

- The *critical exponent* of a word $w$, written $\mathrm{ce}(w)$, is the supremum, over all factors $f$ of $w$, of the exponent of $f$.
  - It measures the largest repetition occurring in a word.
- The critical exponent of the German word `verwendenden` is $8/3$, corresponding to the factor `endenden` .
- In the case of infinite words, this critical exponent can be attained, or not.
- Example: the infinite Fibonacci word $\mathbf{f}$, the fixed point of the morphism $\varphi : 0 \to 01$ and $1 \to 0$:

  $$\mathbf{f} = 01001010010010100101001 \cdots ,$$

  has critical exponent $2 + \tau \doteq 3.61803$, where $\tau = (1 + \sqrt{5})/2$, the golden ratio.

# Repetition threshold

- The *repetition threshold* for a set $S$ of words is the infimum, over all words $w \in S$, of $\mathrm{ce}(w)$.
- This infimum can be attained by a member of $S$, or not.
- Essentially, the repetition threshold measures the *largest unavoidable repetition* over all words in $S$.

# Repetition threshold for small alphabets

Let RT($k$) denote the repetition threshold for the set of all words over a $k$-letter alphabet.

- Squares are not avoidable over a 2-letter alphabet, but overlaps are.
- So RT(2) $= 2$.
- Dejean (1972) proved that the fixed point of the morphism

$$\begin{aligned} 0 &\rightarrow 0120212012102120210 \\ 1 &\rightarrow 1201020120210201021 \\ 2 &\rightarrow 2012101201021012102 \end{aligned}$$

  has critical exponent $7/4$. This is best possible. So RT(3) $= 7/4$.

# Dejean's theorem

- Dejean conjectured that $\mathrm{RT}(4) = 7/5$ and $\mathrm{RT}(k) = k/(k-1)$ for $k \geq 5$.
- This difficult and deep result was finally proven in 2009, by the combined work of many authors (Pansiot, Moulin Ollagnier, Mohammad-Noori, Currie, Carpi, Rampersad, Rao).

# Outline of this talk

In this talk I will discuss computing repetition thresholds for three different classes of words:

- *balanced words* over a $k$-letter alphabet;
- *rich words* over a binary alphabet; and
- *binary words avoiding antisquares*.

What makes this appropriate for NUMERATION 2019 is our proof technique, which is based on numeration systems.

# Balanced words

- A word $w$ is *balanced* if all factors of $w$ of the same length have roughly the same frequencies of letters.
- More precisely, let $|w|_a$ be the number of occurrences of the letter $a$ in $w$. Then a word is balanced if $||x|_a - |y|_a| \leq 1$ for all $a \in \Sigma$ and all factors $x, y$ of $w$ of the same length.
- Example: the German word `Steuerbefehle` is balanced.
- The German word `unausgewogen` is not balanced.

# Binary balanced words and Sturmian words

There is a nice characterization of the infinite binary balanced words:

They coincide with the set of *Sturmian words*: words of the form

$$(\lfloor \alpha(n+1)+\beta \rfloor - \lfloor \alpha n + \beta \rfloor)_{n \geq 1}$$

for real numbers $0 \leq \alpha, \beta < 1$ (or the same thing with floor replaced by ceiling).

# Balanced words over larger alphabets

For larger alphabets, the balanced words are the words obtainable from Sturmian words by replacing the 0's (respectively, the 1's) with a word of the form $x^\omega$, where $x$ is a constant-gap word (due to Pascal Hubert and Ron Graham, independently):

We say $x$ is a *constant-gap* word if two consecutive occurrences of the same letter in $x^\omega = xxx\cdots$ are always separated by the same number of symbols.

For example, 0102 is a constant-gap word, but 0120 is not.

# Repetition thresholds for balanced words

**Theorem.** (Rampersad-JOS-Vandomme; Baranwal-JOS)
The repetition threshold RTBAL($k$) for balanced words over a $k$-letter alphabet is as follows:

| $k$ | RTBAL($k$) |
|---|---|
| 2 | $2 + \tau \doteq 3.61803$ |
| 3 | $2 + \sqrt{2}/2 \doteq 2.7071$ |
| 4 | $1 + \tau/2 \doteq 1.8090$ |
| 5 | $3/2 = 1.5$ |

In addition, we conjecture that RTBAL($k$) $= (k-2)/(k-3)$ for $k \geq 5$.

# Rich words

- A *palindrome* is a word that reads the same forwards and backwards, like the German word `neben`.
- Droubay, Justin, and Pirillo proved that a length-$n$ word contains at most $n$ distinct nonempty palindromes as factors.
- A word that contains exactly $n$ distinct palindromes is called *rich*.
- An example is the German word `besessen`: it contains the palindromes `b, e, s, n, ss, ese, ses, esse`.
- Rich words have been studied extensively, but they are still a bit mysterious.

# Repetition threshold for rich words

- Pelantová and Starosta proved (2013) that every infinite rich word contains a square.
- Vesti (2017) gave upper and lower bounds on the length of a longest square-free rich word over a $k$-letter alphabet.
- Vesti also proposed the problem of determining the repetition threshold for infinite rich words.

# Repetition threshold for rich words

**Theorem.** (Baranwal-JOS) The repetition threshold for infinite binary rich words is between 2.700 and $2 + \frac{\sqrt{2}}{2} = 2.7071 \cdots$, and there is an infinite binary rich word with the latter critical exponent.

# Avoiding antisquares

An *antisquare* is a binary word of the form $x\overline{x}$, where $\overline{x}$ is the binary complement of $x$, mapping 0 to 1 and 1 to 0.

Antisquares can only be avoided trivially over a binary alphabet, by the words $0^\omega$ and $1^\omega$.

Similarly, if we allow only a single antisquare, the only possibilities are $0\,1^\omega$ and $1\,0^\omega$.

However, if we allow exactly two antisquares — 01 and 10, then things become much more interesting.

# Avoiding all antisquares (except 01 and 10)

**Proposition.** Every word in $(1000 + 10000)^*$ and $(1000 + 10000)^\omega$ avoids all antisquares (except 01 and 10).

So there are exponentially many such length-$n$ words, and uncountably many such infinite words.

**Theorem.** (Baranwal-Currie-Mol-Rampersad-JOS) The repetition threshold for binary words avoiding all antisquares (except 01 and 10) is $2 + \tau \doteq 3.61803$.

# Enter numeration systems!

Numeration systems can be used to prove many of the results discussed so far! Basic idea:

- Find a morphism $h$ generating a suitable candidate word $\mathbf{w}$ whose critical exponent matches the repetition threshold
- Construct an appropriate numeration system $S$ so that $\mathbf{w}$ is *$S$-automatic*
  - This means that there is a deterministic finite automaton with output (DFAO) that, given a valid $S$-representation for the integer $n$, computes $\mathbf{w}[n]$
- Express some claim about the properties of $\mathbf{w}$ as a first-order logical formula $\varphi$
- Finally, use a decision procedure to prove or disprove $\varphi$ via a computer program.
- One still has to argue that this candidate word is actually best.

# The easiest case — uniform morphisms

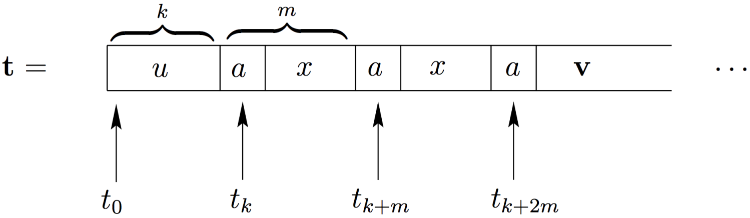
- If the morphism is $k$-uniform (the image of every letter is of length $k$), then the appropriate numeration system is ordinary base-$k$ representation
- The corresponding DFAO can be constructed directly from $h$.

# Expressing overlaps in first-order logic

Example: let the Thue-Morse sequence be given by

$$\mathbf{t} = t_0 t_1 t_2 \cdots = 01101001 \cdots .$$

Suppose $\mathbf{t}$ has an overlap *axaxa* beginning at position $k$ with $|ax| = m \geq 1$. Then we have



So a first-order formula expressing the assertion that $\mathbf{t}$ has an overlap is

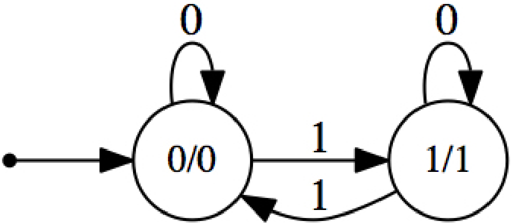
$$\exists k, m \; (m \geq 1) \wedge \forall i \; (i \leq m) \Longrightarrow \mathbf{t}_{k+i} = \mathbf{t}_{k+i+m}.$$

# Automaton for the Thue-Morse word

The corresponding DFAO for the Thue-Morse sequence

$$\mathbf{t} = t_0 t_1 t_2 \cdots = 01101001 \cdots$$

is as follows:



# Walnut

Walnut is free software, written by Hamoon Mousavi, that can evaluate the truth of first-order formulas on sequences defined by automata, using ideas of Büchi, Bruyère, and others.

The basic idea is that the automaton for the sequence is transformed into an automaton accepting the representations (in some numeration system) of the values of the free variables making the formula true.

If there are no free variables, the system answers either `true` or `false`.

# Good news and bad news

Bad news first: the decision procedure used by `Walnut` has enormously bad worst-case running time: it is

$$2^{2^{\cdot^{\cdot^{\cdot^{2^{p(N)}}}}}},$$

where the number of 2's in the exponent is equal to the number of quantifier alternations, $p$ is a polynomial in the length of the particular statement being decided, and $N$ is the number of automaton states needed to describe the underlying sequence.

Good news: even so, we have been successful on something like 90% of the queries we've tried, even with as many as 5 quantifier alternations.

# Reproving Thue's result on overlaps

We take the first-order formula

$$\exists k, m \ (m \geq 1) \wedge \forall i \ (i \leq m) \Longrightarrow \mathbf{t}_{k+i} = \mathbf{t}_{k+i+m}$$

expressing the assertion that **t** has an overlap, and translate it into `Walnut` as follows:

```
E k,m (m >= 1) & Ai (i <= m) => T[k+i]=T[k+i+m]
```

Here `T` represents the two-state DFAO generating the Thue-Morse sequence.

When we enter this into `Walnut`, it answers `false`, so we have proved that **t** is overlap-free.

# Reproving Dejean's result for the three-letter alphabet

More generally, every $k$-uniform morphism over an $s$-letter alphabet (each letter's image has length $k$) corresponds trivially to an $s$-state DFAO with transitions on $0, 1, \ldots, k-1$ taking integer inputs represented in base $k$.

Dejean's morphism $\delta$ given by

$$\begin{aligned}
0 &\to 0120212012102120210 \\
1 &\to 1201020120210201021 \\
2 &\to 2012101201021012102
\end{aligned}$$

corresponds to a 3-state automaton in base 19.

# Reproving Dejean's result for the three-letter alphabet

If we call the automaton D, then the Walnut formula asserting that $\delta^\omega(0)$ has a repetition of exponent $> 7/4$ is as follows:

```
E i,p (p >= 1) & Aj (4*j <= 3*p) => D[i+j] = D[i+j+p]
```

Unfortunately, this runs out of memory (more than 50 Gigs are needed)!

So instead make the variable substitution $k = i + j$ and say

```
E i,p (p >= 1) & Aj,k ((k=i+j)&(4*j <= 3*p) =>
                D[k] = D[k+p]
```

which returns false. Thus we've proved Dejean's theorem for alphabet size 3. This computation took 170 seconds and 15 gigs of storage on an x86_64 GNU/Linux machine.

# Balanced words: a proof with `Walnut`

If a candidate word is given by a suitable morphism, we can check whether it is balanced with a first-order logic formula.

This is not obvious! There is no obvious general way to count the number of occurrences of a letter in a factor of an infinite word with a first-order formula.

However, for binary words, there is an alternative characterization of the balance property that can be expressed in first-order logic: **a word $x$ is unbalanced iff it has two factors of the form $0w0$ and $1w1$.**

# Working with non-uniform morphisms

If the morphism is non-uniform, then the corresponding numeration system is not base-$k$ representation; it depends on the structure of the morphism.

Our decision procedure crucially depends on the ability to implement addition in the numeration system based on the morphism.

But not all numeration systems have this property!

# Fibonacci numeration

One nice system that does is Fibonacci numeration: numbers are represented in the form $\sum_{i\geq 2} e_i F_i$, where $F_2 = 1$, $F_3 = 2$, $F_n = F_n + F_{n-1}$ are the Fibonacci numbers. Here the $e_i$ are in $\{0,1\}$ and no two consecutive $e_i$ are 1.

There is an automaton that decides, on input $x, y, z$ in the Fibonacci numeration system, whether $z = x + y$.

More generally, adders can be implemented for Pisot numeration systems (see work of Christiane Frougny et al.).

# Critical exponents

Given a candidate word $\mathbf{w}$, we'd like an algorithm to determine the critical exponent $e$ of $\mathbf{w}$.

For words defined by uniform morphisms, we know how to do this (Schaeffer-JOS), but for more general morphisms this is not yet known in all cases.

For many numeration systems, the following often works: we write a logical formula corresponding to the period lengths of "large repetitions" close to $e$.

Usually the possible period lengths $p$ will be rare and easy to describe.

We then write a logical formula corresponding to the possible periods $p$, and specifying words of maximal length $\ell$ with this period. Again, usually these will be easy to specify. With some knowledge about the numeration system, we can often compute the supremum of $\ell/p$.

# Balanced words

Using these ideas, we were able to determine the repetition threshold for balanced words over alphabets of size $2, 3, 4, 5$.

For alphabet size $2, 3, 4$ more work was needed to argue that our candidate words are actually best.

For alphabet size 5, since the repetition threshold is $3/2$, a breadth-first search suffices to show that the exponent is optimal over all balanced words.

# Rich words

Recall that a length-$n$ word is rich if it has $n$ distinct nonempty palindromes as factors.

It turns out that this is equivalent to the following: a (finite or infinite) word is rich if every nonempty prefix has a palindromic suffix that does not appear earlier in the word.

This can be expressed in first-order logic. So we can verify that a candidate word is actually rich.

For alphabet size 2, our candidate word is given by $g(h^\omega(0))$:

$$\begin{aligned} h : 0 &\rightarrow 01 \\ 1 &\rightarrow 02 \\ 2 &\rightarrow 022 \end{aligned} \qquad\qquad \begin{aligned} g : 0 &\rightarrow 0 \\ 1 &\rightarrow 01 \\ 2 &\rightarrow 011 \end{aligned}$$

# Rich words II

This word is automatic for the Pell numeration system, built on the terms of the linear recurrence $P_1 = 1$, $P_2 = 2$, and $P_n = 2P_{n-1} + P_{n-2}$.

So we can compute its critical exponent (which is $2 + \sqrt{2}/2 \doteq 2.707$) and verify that it is rich, using `Walnut`.

Breadth-first search shows there is no infinite binary rich word with critical exponent $< 2.700$.

We still do not know if our bound $2 + \sqrt{2}/2$ is optimal.

# Words avoiding antisquares

Idea #1: consider binary words avoiding all antisquares (except 01 and 10), and restrict attention to those with small critical exponent, say $< e$. It turns out we can take $e = 11/3$.

Idea #2: restrict attention to those words $x$ with more 0's than 1's.

These words have a nice property: up to short prefixes and suffixes, we can factor such words into blocks of 0001 and 01.

In other words, such words can be written in the form $x = x_1 h(x_2) x_3$, with $|x_1|$ and $|x_3|$ short, and $h(0) = 0001$, $h(1) = 01$.

Now we can do another factorization on $x_2$, this time in the form $x_2 = y_1 g(y_2) y_3$, where $g(0) = 001$ and $g(1) = 01$.

Furthermore, $y_2$ has such a factorization in terms of $g$ itself.

# Words avoiding antisquares

It follows that sufficiently large $x$ have a factor of the form $h(g^i(0))$ for large $i$, and hence the critical exponent of all infinite words avoiding antisquares (except 01 and 10) is at least that of $h(g^\omega(0))$.

Now we can use `Walnut` to determine this critical exponent, and also verify that $h(g^\omega(0))$ avoids all antisquares (except 01 and 10).

We get

**Theorem.** (Baranwal-Currie-Mol-Rampersad-JOS) Every binary infinite word avoiding all antisquares (except 01 and 10) has critical exponent at least $2 + \tau \doteq 3.61803$. This is best possible, since $h(g^\omega(0))$ has critical exponent exactly $2 + \tau$.

# Future prospects

- Try to extend the computability of critical exponents to non-standard numeration systems (beyond base-$k$)
- Implement a more general decision procedure for deciding properties of Sturmian sequences, based on ideas of Hieronymi, Schaeffer, and others.

# Open Problems

1. Prove that the repetition threshold for balanced words, RTBAL($k$) satisfies RTBAL($k$) $= (k-2)/(k-3)$ for $k \geq 5$. (Known to hold for $k = 5$.)
2. How many length-$n$ rich binary words are there? Upper and lower bounds are known, but they are widely separated.
3. Prove or disprove that the repetition threshold for binary rich words is indeed $2 + \sqrt{2}/2$, and prove analogous results for larger alphabets.
4. Is there a first-order logic characterization of balanced words over alphabets of size $\geq 3$?