Square-Free Words: Theme and Variations

Lucas Mol

THOMPSON RIVERS UNIVERSITY

October 7, 2021

"The three spheres continued to dance in my dream, a patternless, never-repeating dance. Yet, in the depths of my mind, the dance did possess a rhythm."

–from *The Three-Body Problem* by Cixin Liu

Plan

Theme

Variations

A Recent Variation

Alphabets and words

- An *alphabet* is a set of letters, treated simply as symbols, e.g.,
  - $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots, \mathtt{z}\}$ (the English alphabet)
  - $\{\mathtt{0}, \mathtt{1}\}$ (the binary alphabet)
  - $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$ (the alphabet of DNA strings)

Alphabets and words

- An *alphabet* is a set of letters, treated simply as symbols, e.g.,
  - $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots, \mathtt{z}\}$ (the English alphabet)
  - $\{\mathtt{0}, \mathtt{1}\}$ (the binary alphabet)
  - $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$ (the alphabet of DNA strings)
- A *word* is a sequence of letters taken from some alphabet, e.g.,
  - `apple`, `banana`, `clementine` (English words)
  - `0110100110010110` (a binary word)
  - `AAGATGCCGT` (a DNA string)

Alphabets and words

- An *alphabet* is a set of letters, treated simply as symbols, e.g.,
  - $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots, \mathtt{z}\}$ (the English alphabet)
  - $\{\mathtt{0}, \mathtt{1}\}$ (the binary alphabet)
  - $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$ (the alphabet of DNA strings)
- A *word* is a sequence of letters taken from some alphabet, e.g.,
  - `apple`, `banana`, `clementine` (English words)
  - `0110100110010110` (a binary word)
  - `AAGATGCCGT` (a DNA string)
- We are mostly interested in *long* words over *small* alphabets.

Alphabets and words

- An *alphabet* is a set of letters, treated simply as symbols, e.g.,
  - $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots, \mathtt{z}\}$ (the English alphabet)
  - $\{\mathtt{0}, \mathtt{1}\}$ (the binary alphabet)
  - $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$ (the alphabet of DNA strings)
- A *word* is a sequence of letters taken from some alphabet, e.g.,
  - `apple`, `banana`, `clementine` (English words)
  - `0110100110010110` (a binary word)
  - `AAGATGCCGT` (a DNA string)
- We are mostly interested in *long* words over *small* alphabets.
- Which patterns can be avoided, and which patterns must inevitably occur?

SQUARES AND SQUARE-FREE WORDS

- A *square* is a word of the form xx, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`

SQUARES AND SQUARE-FREE WORDS

- A *square* is a word of the form xx, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.

SQUARES AND SQUARE-FREE WORDS

- A *square* is a word of the form xx, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.
  - e.g. The word `0110` has factors:

    `0`,

SQUARES AND SQUARE-FREE WORDS

- A *square* is a word of the form xx, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.
  - e.g. The word `0110` has factors:

    `0`, `1`,

SQUARES AND SQUARE-FREE WORDS

- A *square* is a word of the form xx, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.
  - e.g. The word `0110` has factors:

    `0`, `1`, `01`,

SQUARES AND SQUARE-FREE WORDS

- A *square* is a word of the form xx, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.
  - e.g. The word `0110` has factors:

    `0`, `1`, `01`, `11`,

SQUARES AND SQUARE-FREE WORDS

- A *square* is a word of the form xx, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.
  - e.g. The word `0110` has factors:

    `0`, `1`, `01`, `11`, `10`,

SQUARES AND SQUARE-FREE WORDS

- A *square* is a word of the form *xx*, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.
  - e.g. The word `0110` has factors:

    `0`, `1`, `01`, `11`, `10`, `011`,

SQUARES AND SQUARE-FREE WORDS

- A *square* is a word of the form xx, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.
  - e.g. The word `0110` has factors:

    `0`, `1`, `01`, `11`, `10`, `011`, `110`,

Squares and square-free words

- A *square* is a word of the form xx, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.
  - e.g. The word `0110` has factors:

    `0`, `1`, `01`, `11`, `10`, `011`, `110`, and `0110`

SQUARES AND SQUARE-FREE WORDS

- A *square* is a word of the form xx, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.
  - e.g. The word `0110` has factors:
    `0`, `1`, `01`, `11`, `10`, `011`, `110`, and `0110`, but NOT `00`.

Squares and square-free words

- A *square* is a word of the form xx, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.
  - e.g. The word `0110` has factors:
    `0`, `1`, `01`, `11`, `10`, `011`, `110`, and `0110`, but NOT `00`.

SQUARES AND SQUARE-FREE WORDS

- A *square* is a word of the form xx, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.
  - e.g. The word `0110` has factors:
    `0`, `1`, `01`, `11`, `10`, `011`, `110`, and `0110`, but NOT `00`.
- A word is *square-free* if it contains no squares as factors.
  - `apple`

SQUARES AND SQUARE-FREE WORDS

- A *square* is a word of the form *xx*, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.
  - e.g. The word `0110` has factors:
    `0`, `1`, `01`, `11`, `10`, `011`, `110`, and `0110`, but NOT `00`.
- A word is *square-free* if it contains no squares as factors.
  - `apple` – not square-free

SQUARES AND SQUARE-FREE WORDS

- A *square* is a word of the form xx, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.
  - e.g. The word `0110` has factors:
    `0`, `1`, `01`, `11`, `10`, `011`, `110`, and `0110`, but NOT `00`.
- A word is *square-free* if it contains no squares as factors.
  - `apple` – not square-free
  - `banana`

Squares and square-free words

- A *square* is a word of the form *xx*, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.
  - e.g. The word `0110` has factors:
    `0`, `1`, `01`, `11`, `10`, `011`, `110`, and `0110`, but NOT `00`.
- A word is *square-free* if it contains no squares as factors.
  - `apple` – not square-free
  - `banana` – not square-free

Squares and square-free words

- A *square* is a word of the form xx, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.
  - e.g. The word `0110` has factors:
    `0`, `1`, `01`, `11`, `10`, `011`, `110`, and `0110`, but NOT `00`.
- A word is *square-free* if it contains no squares as factors.
  - `apple` – not square-free
  - `banana` – not square-free
  - `clementine`

SQUARES AND SQUARE-FREE WORDS

- A *square* is a word of the form *xx*, e.g.,
  - `murmur`, `hotshots`, `caracara`
  - `00`, `010212010212`
- The *factors* of a word are its contiguous subwords.
  - e.g. The word `0110` has factors:
    `0`, `1`, `01`, `11`, `10`, `011`, `110`, and `0110`, but NOT `00`.
- A word is *square-free* if it contains no squares as factors.
  - `apple` – not square-free
  - `banana` – not square-free
  - `clementine` – square-free

ARE SQUARES AVOIDABLE?

Q: Are there arbitrarily long square-free words?

Are squares avoidable?

Q: Are there arbitrarily long square-free words?
A: Sure, over an infinite alphabet!

$$a_0 a_1 a_2 a_3 \dots$$

Are squares avoidable?

Q: Are there arbitrarily long square-free words?
A: Sure, over an infinite alphabet!

$$a_0 a_1 a_2 a_3 \ldots$$

Revised Q: Are there arbitrarily long square-free words over a *finite* alphabet?

Are squares avoidable?

Q: Are there arbitrarily long square-free words?
A: Sure, over an infinite alphabet!

$$a_0 a_1 a_2 a_3 \dots$$

Revised Q: Are there arbitrarily long square-free words over a *finite* alphabet?
Revised A: Hmmmmm...

Are squares avoidable?

Q: Are there arbitrarily long square-free words?
A: Sure, over an infinite alphabet!

$$a_0 a_1 a_2 a_3 \ldots$$

Revised Q: Are there arbitrarily long square-free words over a *finite* alphabet?
Revised A: Hmmmmm...

- Over an alphabet of size one, say $\{0\}$?

Are squares avoidable?

Q: Are there arbitrarily long square-free words?
A: Sure, over an infinite alphabet!

$$a_0 a_1 a_2 a_3 \ldots$$

Revised Q: Are there arbitrarily long square-free words over a *finite* alphabet?
Revised A: Hmmmmm...

- Over an alphabet of size one, say $\{0\}$? No.

ARE SQUARES AVOIDABLE?

Q: Are there arbitrarily long square-free words?
A: Sure, over an infinite alphabet!

$$a_0 a_1 a_2 a_3 \ldots$$

Revised Q: Are there arbitrarily long square-free words over a *finite* alphabet?
Revised A: Hmmmmm. . .

- Over an alphabet of size one, say $\{0\}$? No.
- Over an alphabet of size two, say $\{0, 1\}$?

ARE SQUARES AVOIDABLE?

Q: Are there arbitrarily long square-free words?
A: Sure, over an infinite alphabet!

$$a_0 a_1 a_2 a_3 \ldots$$

Revised Q: Are there arbitrarily long square-free words over a *finite* alphabet?
Revised A: Hmmmmm...

- Over an alphabet of size one, say $\{0\}$? No.
- Over an alphabet of size two, say $\{0, 1\}$? No.

Are squares avoidable?

Q: Are there arbitrarily long square-free words?
A: Sure, over an infinite alphabet!

$$a_0 a_1 a_2 a_3 \dots$$

Revised Q: Are there arbitrarily long square-free words over a *finite* alphabet?
Revised A: Hmmmmm...

- Over an alphabet of size one, say $\{0\}$? No.
- Over an alphabet of size two, say $\{0, 1\}$? No.
- Over an alphabet of size three, say $\{0, 1, 2\}$?

BACKTRACKING

- This is called a *backtracking algorithm*.
- It is a depth-first search through the tree of all square-free words over $\{0, 1, 2\}$.
- We now have *strong evidence* that there are arbitrarily long square-free words over $\{0, 1, 2\}$.
- This motivates us to find a proof!

BACKTRACKING

- This is called a *backtracking algorithm*.
- It is a depth-first search through the tree of all square-free words over $\{0, 1, 2\}$.
- We now have *strong evidence* that there are arbitrarily long square-free words over $\{0, 1, 2\}$.
- This motivates us to find a proof!

letters
“The three ~~spheres~~ continued to dance in my dream, a patternless, never-repeating dance. Yet, in the depths of my mind, the dance did possess a rhythm.”

–from *The Three-Body Problem* by Cixin Liu

A CONSTRUCTION

A CONSTRUCTION

- Define a map h by
  - $h(\texttt{0}) = \texttt{012}$,
  - $h(\texttt{1}) = \texttt{02}$, and
  - $h(\texttt{2}) = \texttt{1}$.

A CONSTRUCTION

- Define a map h by
  - $h(\mathtt{0}) = \mathtt{012}$,
  - $h(\mathtt{1}) = \mathtt{02}$, and
  - $h(\mathtt{2}) = \mathtt{1}$.
- Extend h to all words over $\{\mathtt{0}, \mathtt{1}, \mathtt{2}\}$ in the obvious way:

$$h(\mathtt{0120}) = h(\mathtt{0})h(\mathtt{1})h(\mathtt{2})h(\mathtt{0}) = \mathtt{012021012}$$

A CONSTRUCTION

- Define a map h by
  - $h(\texttt{0}) = \texttt{012}$,
  - $h(\texttt{1}) = \texttt{02}$, and
  - $h(\texttt{2}) = \texttt{1}$.
- Extend h to all words over $\{\texttt{0}, \texttt{1}, \texttt{2}\}$ in the obvious way:

$$h(\texttt{0120}) = h(\texttt{0})h(\texttt{1})h(\texttt{2})h(\texttt{0}) = \texttt{012021012}$$

- We start with $\texttt{0}$, and repeatedly apply h.

A CONSTRUCTION

- Define a map h by
  - $h(\mathtt{0}) = \mathtt{012}$,
  - $h(\mathtt{1}) = \mathtt{02}$, and
  - $h(\mathtt{2}) = \mathtt{1}$.
- Extend h to all words over $\{\mathtt{0}, \mathtt{1}, \mathtt{2}\}$ in the obvious way:

$$h(\mathtt{0120}) = h(\mathtt{0})h(\mathtt{1})h(\mathtt{2})h(\mathtt{0}) = \mathtt{012021012}$$

- We start with $\mathtt{0}$, and repeatedly apply h.

$$h(\mathtt{0}) = \mathtt{012}$$

A CONSTRUCTION

- Define a map h by
  - $h(\mathtt{0}) = \mathtt{012}$,
  - $h(\mathtt{1}) = \mathtt{02}$, and
  - $h(\mathtt{2}) = \mathtt{1}$.
- Extend h to all words over $\{\mathtt{0}, \mathtt{1}, \mathtt{2}\}$ in the obvious way:

$$h(\mathtt{0120}) = h(\mathtt{0})h(\mathtt{1})h(\mathtt{2})h(\mathtt{0}) = \mathtt{012021012}$$

- We start with $\mathtt{0}$, and repeatedly apply h.

$$\begin{aligned} h(\mathtt{0}) &= \mathtt{012} \\ h^2(\mathtt{0}) &= \mathtt{012021} \end{aligned}$$

A CONSTRUCTION

- Define a map h by
  - $h(\mathtt{0}) = \mathtt{012}$,
  - $h(\mathtt{1}) = \mathtt{02}$, and
  - $h(\mathtt{2}) = \mathtt{1}$.
- Extend h to all words over $\{\mathtt{0}, \mathtt{1}, \mathtt{2}\}$ in the obvious way:

$$h(\mathtt{0120}) = h(\mathtt{0})h(\mathtt{1})h(\mathtt{2})h(\mathtt{0}) = \mathtt{012021012}$$

- We start with $\mathtt{0}$, and repeatedly apply h.

$$h(\mathtt{0}) = \mathtt{012}$$
$$h^2(\mathtt{0}) = \mathtt{012021}$$
$$h^3(\mathtt{0}) = \mathtt{012021012102}$$

A CONSTRUCTION

- Define a map h by
  - $h(\mathtt{0}) = \mathtt{012}$,
  - $h(\mathtt{1}) = \mathtt{02}$, and
  - $h(\mathtt{2}) = \mathtt{1}$.
- Extend h to all words over $\{\mathtt{0}, \mathtt{1}, \mathtt{2}\}$ in the obvious way:

$$h(\mathtt{0120}) = h(\mathtt{0})h(\mathtt{1})h(\mathtt{2})h(\mathtt{0}) = \mathtt{012021012}$$

- We start with $\mathtt{0}$, and repeatedly apply h.

$$\begin{aligned} h(\mathtt{0}) &= \mathtt{012} \\ h^2(\mathtt{0}) &= \mathtt{012021} \\ h^3(\mathtt{0}) &= \mathtt{012021012102} \\ h^4(\mathtt{0}) &= \mathtt{012021012102012021020121} \end{aligned}$$

A CONSTRUCTION

- Define a map h by
  - $h(\mathtt{0}) = \mathtt{012}$,
  - $h(\mathtt{1}) = \mathtt{02}$, and
  - $h(\mathtt{2}) = \mathtt{1}$.
- Extend h to all words over $\{\mathtt{0}, \mathtt{1}, \mathtt{2}\}$ in the obvious way:

$$h(\mathtt{0120}) = h(\mathtt{0})h(\mathtt{1})h(\mathtt{2})h(\mathtt{0}) = \mathtt{012021012}$$

- We start with $\mathtt{0}$, and repeatedly apply h.

$$\begin{aligned}
h(\mathtt{0}) &= \mathtt{012} \\
h^2(\mathtt{0}) &= \mathtt{012021} \\
h^3(\mathtt{0}) &= \mathtt{012021012102} \\
h^4(\mathtt{0}) &= \mathtt{012021012102012021020121} \\
&\vdots \\
h^\omega(\mathtt{0}) &= \mathtt{012021012102012021020121}\ldots
\end{aligned}$$

A CONSTRUCTION

Claim: $h^{\omega}(\mathtt{0}) = \mathtt{0120210121020120210201 21}\ldots$ is square-free.

- Proof Idea: Show that if w is square-free, then $h(w)$ is square-free.

A CONSTRUCTION

Claim: $h^\omega(\texttt{0}) = \texttt{012021012102012021020121}\ldots$ is square-free.

- Proof Idea: Show that if w is square-free, then $h(w)$ is square-free.
- Snag: $h(\texttt{010}) = \texttt{01202012}$ and $h(02120) = \texttt{0121021012}$

A CONSTRUCTION

Claim: $h^{\omega}(\mathtt{0}) = \mathtt{012021012102012021020121}\ldots$ is square-free.

- Proof Idea: Show that if w is square-free, then $h(w)$ is square-free.
- Snag: $h(\mathtt{010}) = \mathtt{01202012}$ and $h(02120) = \mathtt{0121021012}$
- Fix: Show that if w is square-free and does not contain $\mathtt{010}$ or $\mathtt{212}$, then $h(w)$ is square-free and does not contain $\mathtt{010}$ or $\mathtt{212}$.

A CONSTRUCTION

Claim: $h^\omega(\mathtt{0}) = \mathtt{01202101210201202102012 1}\ldots$ is square-free.

- Proof Idea: Show that if w is square-free, then $h(w)$ is square-free.
- Snag: $h(\mathtt{010}) = \mathtt{01202012}$ and $h(02120) = \mathtt{0121021012}$
- Fix: Show that if w is square-free and does not contain $\mathtt{010}$ or $\mathtt{212}$, then $h(w)$ is square-free and does not contain $\mathtt{010}$ or $\mathtt{212}$.

This is just one construction.

- In fact, the number of square-free words of length k over $\{\mathtt{0}, \mathtt{1}, \mathtt{2}\}$ grows exponentially in k.
- The best known bounds on the growth rate are due to Shur, from 2012.

THE ORIGIN OF COMBINATORICS ON WORDS

Axel Thue
(1863-1922)

- Thue was first to demonstrate that squares are *avoidable* over the alphabet $\{0, 1, 2\}$.
- This is recognized as the beginning of the field of *combinatorics on words*.
- Current work in the field is still directly inspired by his work.
- Perspectives can be algorithmic, combinatorial, algebraic, etc.
- Connections to number theory, abstract algebra, automata theory, logic, computability, etc.

My introduction to combinatorics on words

Five years ago, I met these folks.

James Currie

Narad Rampersad

Plan

Theme

Variations

A Recent Variation

Variation I – Patterns

- Example: The word `referee` is an instance of the pattern *xyzxz* with $x = \mathtt{re}$, $y = \mathtt{f}$, $z = \mathtt{e}$.
  - This pattern is *unavoidable* – over every finite alphabet, every sufficiently long word will contain an instance of it.
- There are nice characterizations of avoidable and unavoidable patterns (BEM 1979, Zimin 1982).
- For many avoidable patterns, we do not know their *avoidability index* – the size of the smallest alphabet over which they can be avoided.
- The avoidability index of every pattern with at most three variables is known (Cassaigne 1994).
- The largest known avoidability index is 5 (Clark 2001).
- Theorem (Ochem 2021): There are *patterns with reversal* of arbitrarily large avoidability index.

Variation II – Weaker Squares

An *abelian square* is a word of the form $x\tilde{x}$, where $\tilde{x}$ is an *anagram* of x.

- Examples: `mesosome`, `reappear`, `intestines`
- Theorem (Keränen 1992): Abelian squares are avoidable over four letters.

Variation II – Weaker Squares

An *abelian square* is a word of the form $x\tilde{x}$, where $\tilde{x}$ is an *anagram* of x.

- Examples: `mesosome`, `reappear`, `intestines`
- Theorem (Keränen 1992): Abelian squares are avoidable over four letters.
- The word $\sigma^{\omega}(\mathtt{0})$ avoids abelian squares, where

$$\sigma(0) = \mathtt{0120232123203231301020103101213121021232021013010203212320231210212320232132303132120}$$
$$\sigma(1) = \mathtt{1231303230310302012131210212320232132303132120121310323031302321323031303203010203231}$$
$$\sigma(2) = \mathtt{2302010301021013123202321323031303203010203231232021030102013032030102010310121310302}$$
$$\sigma(3) = \mathtt{3013121012132120230313032030102010310121310302303132101213120103101213121021232021013}$$

Variation II – Weaker Squares

An *additive square* is a word of the form $x\tilde{x}$, where x and $\tilde{x}$ have the same length and same sum.

- Examples: `021201`, `013202`
- Are these avoidable over some finite subset of $\mathbb{Z}$?

Variation II – Weaker Squares

An *additive square* is a word of the form $x\tilde{x}$, where x and $\tilde{x}$ have the same length and same sum.

- Examples: `021201`, `013202`
- Are these avoidable over some finite subset of $\mathbb{Z}$?
  - We. Don't. Know.

VARIATION II – WEAKER SQUARES

An *additive square* is a word of the form $x\tilde{x}$, where x and $\tilde{x}$ have the same length and same sum.

- Examples: `021201`, `013202`
- Are these avoidable over some finite subset of $\mathbb{Z}$?
  - We. Don't. Know.
- Theorem (Cassaigne, Currie, Schaeffer and Shallit 2014): Additive cubes are avoidable over $\{0, 1, 3, 4\}$.

VARIATION II – WEAKER SQUARES

An *additive square* is a word of the form $x\tilde{x}$, where x and $\tilde{x}$ have the same length and same sum.

- Examples: `021201`, `013202`
- Are these avoidable over some finite subset of $\mathbb{Z}$?
  - We. Don't. Know.
- Theorem (Cassaigne, Currie, Schaeffer and Shallit 2014): Additive cubes are avoidable over $\{0, 1, 3, 4\}$.
- The word $h^{\omega}(\mathtt{0})$ avoids additive cubes, where

$$h(\mathtt{0}) = \mathtt{03}$$
$$h(\mathtt{1}) = \mathtt{43}$$
$$h(\mathtt{3}) = \mathtt{1}$$
$$h(\mathtt{4}) = \mathtt{01}$$

VARIATION II – WEAKER SQUARES

An *additive square* is a word of the form $x\tilde{x}$, where x and $\tilde{x}$ have the same length and same sum.

- Examples: `021201`, `013202`
- Are these avoidable over some finite subset of $\mathbb{Z}$?
  - We. Don't. Know.
- Theorem (Cassaigne, Currie, Schaeffer and Shallit 2014): Additive cubes are avoidable over $\{0, 1, 3, 4\}$.
- The word $h^{\omega}(\texttt{0})$ avoids additive cubes, where

$$
\begin{aligned}
h(\texttt{0}) &= \texttt{03} \\
h(\texttt{1}) &= \texttt{43} \\
h(\texttt{3}) &= \texttt{1} \\
h(\texttt{4}) &= \texttt{01}
\end{aligned}
$$

- Theorem (Lietard and Rosenfeld 2021): Additive cubes are avoidable over every subset of $\mathbb{Z}$ of size 4 that is not equivalent to $\{0, 1, 2, 3\}$.

VARIATION II – WEAKER SQUARES

Decision Algorithms

- Theorem (Currie and Rampersad 2012): There is an algorithm which decides, under certain conditions on h, whether $h^{\omega}(\mathtt{0})$ contains abelian squares (cubes, 4th powers, etc.)

Variation II – Weaker Squares

Decision Algorithms

- Theorem (Currie and Rampersad 2012): There is an algorithm which decides, under certain conditions on h, whether $h^{\omega}(\texttt{0})$ contains abelian squares (cubes, 4th powers, etc.)
- Theorem (Rao and Rosenfeld 2018): Weaker conditions on h, less efficient algorithm.

Variation II – Weaker Squares

Decision Algorithms

- Theorem (Currie and Rampersad 2012): There is an algorithm which decides, under certain conditions on h, whether $h^\omega(\mathtt{0})$ contains abelian squares (cubes, 4th powers, etc.)
- Theorem (Rao and Rosenfeld 2018): Weaker conditions on h, less efficient algorithm.
- Theorem (Currie, Mol, and Rampersad 2021+++): Stronger conditions on h, more efficient algorithm that handles *additive* powers.

VARIATION III – FRACTIONAL REPETITIONS

- Examples:
  - `alfalfa` is a $7/3$-power
  - `salsa` is a $5/3$-power
  - `ingesting` is a $3/2$-power
- Dejean's Theorem, established gradually through the work of many authors, describes the exact threshold between avoidable and unavoidable powers over n letters:

$$\mathrm{RT}(n) = \begin{cases} 2, & \text{if } n = 2; \\ \frac{7}{4}, & \text{if } n = 3; \\ \frac{7}{5}, & \text{if } n = 4; \\ \frac{n}{n-1}, & \text{if } n \geq 5. \end{cases}$$

- Theorem (Currie, Mol, and Rampersad 2020): For all $n \geq 27$, the number of *threshold words* of length k over n letters grows exponentially in k.

VARIATION IV – SQUARE-FREE GRAPH COLOURING

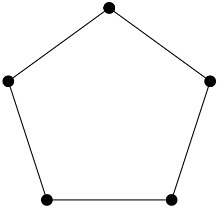
- Graph colouring: Assign colours to the vertices of a graph G so that no two adjacent vertices are given the same colour.

VARIATION IV – SQUARE-FREE GRAPH COLOURING

- Graph colouring: Assign colours to the vertices of a graph G so that no two adjacent vertices are given the same colour.
- The *chromatic number* of G, denoted $\chi(G)$, is the minimum number of colours that are needed.

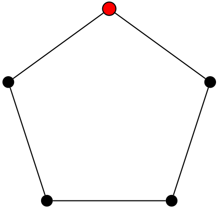
VARIATION IV – SQUARE-FREE GRAPH COLOURING

- Graph colouring: Assign colours to the vertices of a graph G so that no two adjacent vertices are given the same colour.
- The *chromatic number* of G, denoted $\chi(G)$, is the minimum number of colours that are needed.



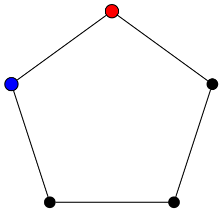
VARIATION IV – SQUARE-FREE GRAPH COLOURING

- Graph colouring: Assign colours to the vertices of a graph G so that no two adjacent vertices are given the same colour.
- The *chromatic number* of G, denoted $\chi(G)$, is the minimum number of colours that are needed.



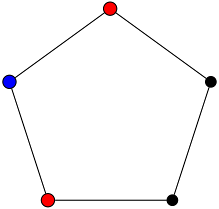
VARIATION IV – SQUARE-FREE GRAPH COLOURING

- Graph colouring: Assign colours to the vertices of a graph *G* so that no two adjacent vertices are given the same colour.
- The *chromatic number* of *G*, denoted $\chi(G)$, is the minimum number of colours that are needed.



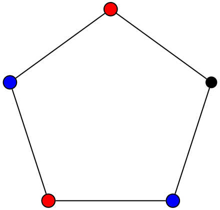
Variation IV – Square-free graph colouring

- Graph colouring: Assign colours to the vertices of a graph G so that no two adjacent vertices are given the same colour.
- The *chromatic number* of G, denoted $\chi(G)$, is the minimum number of colours that are needed.



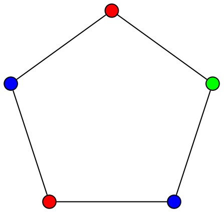
Variation IV – Square-free graph colouring

- Graph colouring: Assign colours to the vertices of a graph G so that no two adjacent vertices are given the same colour.
- The *chromatic number* of G, denoted $\chi(G)$, is the minimum number of colours that are needed.



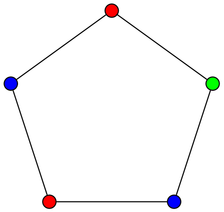
VARIATION IV – SQUARE-FREE GRAPH COLOURING

- Graph colouring: Assign colours to the vertices of a graph G so that no two adjacent vertices are given the same colour.
- The *chromatic number* of G, denoted $\chi(G)$, is the minimum number of colours that are needed.



Variation IV – Square-free graph colouring

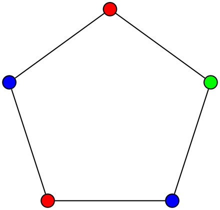
- Graph colouring: Assign colours to the vertices of a graph G so that no two adjacent vertices are given the same colour.
- The *chromatic number* of G, denoted $\chi(G)$, is the minimum number of colours that are needed.



- So $\chi(C_5) = 3$.

VARIATION IV – SQUARE-FREE GRAPH COLOURING

- Graph colouring: Assign colours to the vertices of a graph G so that no two adjacent vertices are given the same colour.
- The *chromatic number* of G, denoted $\chi(G)$, is the minimum number of colours that are needed.



- So $\chi(C_5) = 3$.
- If we start at the top and walk counterclockwise, we will have walked along a *square*: RBRB

Variation IV – Square-free graph colouring

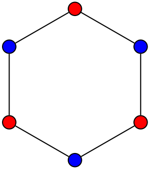
- Assign colours to the vertices of a graph G so that no squares can be obtained by walking a *simple path* in G.

Variation IV – Square-free graph colouring

- Assign colours to the vertices of a graph G so that no squares can be obtained by walking a *simple path* in G.
- The *nonrepetitive chromatic number* of G, denoted $\pi(G)$, is the minimum number of colours that are needed.

VARIATION IV – SQUARE-FREE GRAPH COLOURING

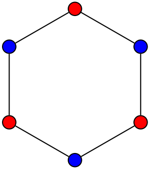
- Assign colours to the vertices of a graph G so that no squares can be obtained by walking a *simple path* in G.
- The *nonrepetitive chromatic number* of G, denoted $\pi(G)$, is the minimum number of colours that are needed.



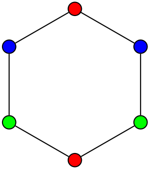
$\chi(C_6) = 2$

Variation IV – Square-free graph colouring

- Assign colours to the vertices of a graph G so that no squares can be obtained by walking a *simple path* in G.
- The *nonrepetitive chromatic number* of G, denoted $\pi(G)$, is the minimum number of colours that are needed.



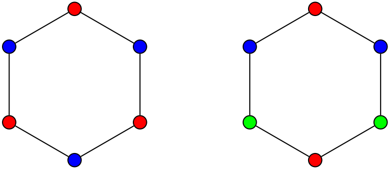
$\chi(C_6) = 2$



$\pi(C_6) = 3$

VARIATION IV – SQUARE-FREE GRAPH COLOURING

- Assign colours to the vertices of a graph G so that no squares can be obtained by walking a *simple path* in G.
- The *nonrepetitive chromatic number* of G, denoted $\pi(G)$, is the minimum number of colours that are needed.

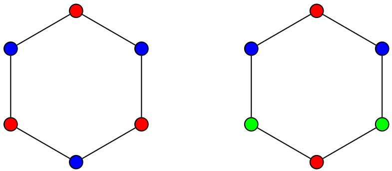


$\chi(C_6) = 2$ $\pi(C_6) = 3$

- Theorem (Appel and Hakken 1976): If G is planar, then $\chi(G) \leq 4$.

VARIATION IV – SQUARE-FREE GRAPH COLOURING

- Assign colours to the vertices of a graph G so that no squares can be obtained by walking a *simple path* in G.
- The *nonrepetitive chromatic number* of G, denoted $\pi(G)$, is the minimum number of colours that are needed.



$\chi(C_6) = 2$ $\qquad$ $\pi(C_6) = 3$

- Theorem (Appel and Hakken 1976): If G is planar, then $\chi(G) \leq 4$.
- Theorem (Dujmović et al. 2020): If G is planar, then $\pi(G) \leq 768$.

BONUS VARIATIONS

Combine some of variations I–IV and consider

- Avoiding patterns in the abelian sense
- Repetition thresholds for families of graphs
- Abelian square-free graph colourings
- An abelian repetition threshold
- etc.

Homework

- Are there patterns of arbitrarily large avoidability index? Or at least greater than 5?
- Are additive squares avoidable over some finite subset of $\mathbb{Z}$?
- Are additive cubes avoidable over $\{0, 1, 2, 3\}$?
- Show exponential growth of the number of threshold words in the remaining cases.
- Improve on the “768 colour theorem”.

Why I like combinatorics on words (and maybe you would too)

- There is a great mix of open problems; some concrete and approachable, some very deep.
- The barriers to entry are relatively low.
- Computers are extremely helpful for making conjectures.
- We use a wide variety of mathematical and computational tools.
- It is a small but dedicated worldwide community.
- Let me know if you would like to borrow some books, look at some papers, etc.

Thank you!

Backtracking... Revisited

- Our backtracking algorithm for square-free words over $\{0, 1, 2\}$ will never terminate.
- But suppose we don't ever want to "backtrack".
- If we can't add a letter at the end, we'll try adding a letter one position before the end.
- If that doesn't work, we'll keep moving towards the beginning, adding a letter as soon as possible.

Q: Does this *nonchalant* algorithm terminate?

Q: Are there any square-free words that cannot be extended to a longer square-free word by adding a single letter at some position?

- Asked by Grytczuk, Kordulewski, and Niewiadomski (2020).

Extensions of a word

- Let w be a word over a fixed alphabet Σ.
- An *extension* of w is a word obtained from w by adding a single letter from Σ at any position.
- Some extensions of the English word `pans` are:

`spans` `plans` `pawns` `pants` `pansy`

left extension internal extensions right extension

Extremal square-free words

- A square-free word w is called *extremal* if every extension of w contains a square.
  - These words would halt the nonchalant algorithm!
- The following is a shortest extremal square-free word over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$:

$$\mathtt{abcabacbcabcbabcabacbcabc}$$

EXTREMAL SQUARE-FREE WORDS

- A square-free word w is called *extremal* if every extension of w contains a square.
  - These words would halt the nonchalant algorithm!
- The following is a shortest extremal square-free word over $\{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$:

$$\texttt{aabcabacbcabcbabcabacbcabc}$$

Extremal square-free words

- A square-free word *w* is called *extremal* if every extension of *w* contains a square.
  - These words would halt the nonchalant algorithm!
- The following is a shortest extremal square-free word over $\{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$:

$$\texttt{babcabacbcabcbabcabacbcabc}$$

Extremal square-free words

- A square-free word w is called *extremal* if every extension of w contains a square.
  - These words would halt the nonchalant algorithm!
- The following is a shortest extremal square-free word over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$:

$$\underbrace{\color{red}{\mathtt{b}}\mathtt{abcabacbcabc}}\,\underbrace{\mathtt{babcabacbcabc}}$$

Extremal square-free words

- A square-free word w is called *extremal* if every extension of w contains a square.
  - These words would halt the nonchalant algorithm!
- The following is a shortest extremal square-free word over $\{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$:

$$\texttt{cabcabacbcabcbabcabacbcabc}$$

EXTREMAL SQUARE-FREE WORDS

- A square-free word *w* is called *extremal* if every extension of *w* contains a square.
  - These words would halt the nonchalant algorithm!
- The following is a shortest extremal square-free word over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$:

$$\underbrace{\mathtt{cab}}\,\underbrace{\mathtt{cab}}\,\mathtt{acbcabcbabcabacbcabc}$$

Extremal square-free words

- A square-free word *w* is called *extremal* if every extension of *w* contains a square.
  - These words would halt the nonchalant algorithm!
- The following is a shortest extremal square-free word over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$:

$$\mathtt{aabcabacbcabcbabcabacbcabc}$$

Extremal square-free words

- A square-free word *w* is called *extremal* if every extension of *w* contains a square.
  - These words would halt the nonchalant algorithm!
- The following is a shortest extremal square-free word over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$:

$$\mathtt{abbcabacbcabcbabcabacbcabc}$$

EXTREMAL SQUARE-FREE WORDS

- A square-free word w is called *extremal* if every extension of w contains a square.
  - These words would halt the nonchalant algorithm!
- The following is a shortest extremal square-free word over $\{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$:

$$\mathtt{acbcabacbcabcbabcabacbcabc}$$

Extremal square-free words

- A square-free word w is called *extremal* if every extension of w contains a square.
  - These words would halt the nonchalant algorithm!
- The following is a shortest extremal square-free word over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$:

$$\underbrace{\mathtt{a}{\color{red}\mathtt{c}}\mathtt{bcab}}\,\underbrace{\mathtt{acbcab}}\,\mathtt{cbabcabacbcabc}$$

EXTREMAL SQUARE-FREE WORDS

- A square-free word w is called *extremal* if every extension of w contains a square.
  - These words would halt the nonchalant algorithm!
- The following is a shortest extremal square-free word over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$:

$$\mathtt{abacabacbcabcbabcabacbcabc}$$

Extremal square-free words

- A square-free word w is called *extremal* if every extension of w contains a square.
  - These words would halt the nonchalant algorithm!
- The following is a shortest extremal square-free word over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$:

$$\underbrace{\mathtt{ab{\color{red}a}c}}\underbrace{\mathtt{abac}}\mathtt{bcabcbabcabacbcabc}$$

EXTREMAL SQUARE-FREE WORDS

- A square-free word w is called *extremal* if every extension of w contains a square.
  - These words would halt the nonchalant algorithm!
- The following is a shortest extremal square-free word over $\{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$:

abbcabacbcabcbabcabacbcabc

Extremal square-free words

- A square-free word *w* is called *extremal* if every extension of *w* contains a square.
  - These words would halt the nonchalant algorithm!
- The following is a shortest extremal square-free word over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$:

$$\mathtt{abccabacbcabcbabcabacbcabc}$$

Extremal square-free words

- A square-free word w is called *extremal* if every extension of w contains a square.
  - These words would halt the nonchalant algorithm!
- The following is a shortest extremal square-free word over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$:

$$\mathtt{abcabacbcabcbabcabacbcabc}$$

EXTREMAL SQUARE-FREE WORDS

- A square-free word w is called *extremal* if every extension of w contains a square.
  - These words would halt the nonchalant algorithm!
- The following is a shortest extremal square-free word over $\{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$:

$$\mathtt{abcabacbcabcbabcabacbcabc}$$

- Are there arbitrarily long extremal square-free words over $\{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$?

SHORT EXTREMAL SQUARE-FREE WORDS

1 None
2 None
3 None
4 None
5 None
6 None
7 None
8 None
9 None
10 None
11 None
12 None
13 None
14 None
15 None
16 None
17 None
18 None
19 None
20 None
21 None
22 None
23 None
24 None
25 0120102120121012010212012
26 None
27 None
28 None
29 None
30 None
31 None
32 None
33 None
34 None
35 None
36 None
37 None
38 None
39 None
40 None
41 01021012021020121021201021012021020121021
42 None
43 None
44 None
45 None
46 None
47 None
48 010212012102010212012101202120121020102120121020
49 None
50 01021201021012021020121012021201021012021020121020
51 None
52 None
53 None
54 None
55 None
56 None
57 None
58 None
59 None
60 None

Theorem (GKN 2020): There are arbitrarily long extremal square-free words over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$.

- The main idea is to use “nearly extremal” square-free words as building blocks.
- A square-free word is *nearly extremal* if it has at most two square-free extensions; one left extension, and one right extension.
- Lemma: If *u* and *v* are nearly extremal square-free, and *uv* is square-free, then *uv* is nearly extremal square-free.

BUILDING BLOCKS

- Fact: The word

  $$N = \texttt{abacbabcabacbcacbabcabacabcbabcabacbcabcb}$$

  is nearly extremal square-free.

Building blocks

- Fact: The word

  $$N = \texttt{abacbabcabacbcacbabcabacabcbabcabacbcabcb}$$

  is nearly extremal square-free.
- For every permutation π of the set $\{\texttt{a}, \texttt{b}, \texttt{c}\}$, let N_π denote the word obtained from N by applying π.

Building blocks

- Fact: The word

  $$N = \mathtt{abacbabcabacbcacbabcabacabcbabcabacbcabcb}$$

  is nearly extremal square-free.
- For every permutation π of the set $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, let N_π denote the word obtained from N by applying π.
- Let $N_{\tilde{\pi}}$ denote the reversal of the word N_π.

Building blocks

- Fact: The word

$$N = \mathtt{abacbabcabacbcacbabcabacabcbabcabacbcabcb}$$

  is nearly extremal square-free.
- For every permutation π of the set $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, let N_π denote the word obtained from N by applying π.
- Let $N_{\tilde{\pi}}$ denote the reversal of the word N_π.
- In this way, the word N gives rise to twelve distinct nearly extremal square-free words:

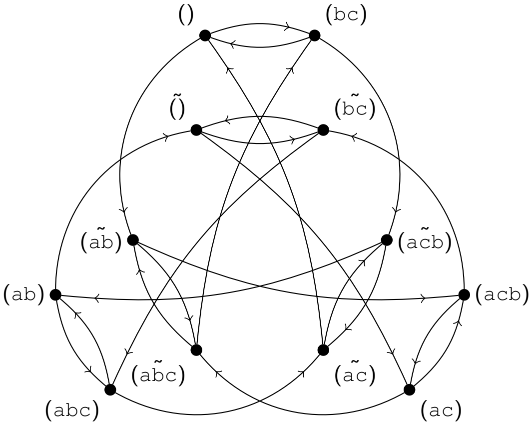
$$N_{()}, N_{(\mathtt{ab})}, N_{(\mathtt{ac})}, N_{(\mathtt{bc})}, N_{(\mathtt{abc})}, N_{(\mathtt{abc})}$$
$$N_{\tilde{()}}, N_{\widetilde{(\mathtt{ab})}}, N_{\widetilde{(\mathtt{ac})}}, N_{\widetilde{(\mathtt{bc})}}, N_{\widetilde{(\mathtt{abc})}}, N_{\widetilde{(\mathtt{abc})}}$$

A USEFUL DIGRAPH

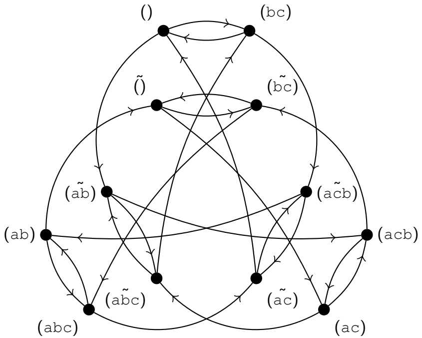
- Fact: The words $N_{()}N_{(\mathtt{bc})}$ and $N_{()}N_{(\tilde{\mathtt{a}}\mathtt{b})}$ are square-free.

A USEFUL DIGRAPH

- Fact: The words $N_{()}N_{(\mathtt{bc})}$ and $N_{()}N_{\widetilde{(\mathtt{ab})}}$ are square-free.
- We are led to the following digraph D.

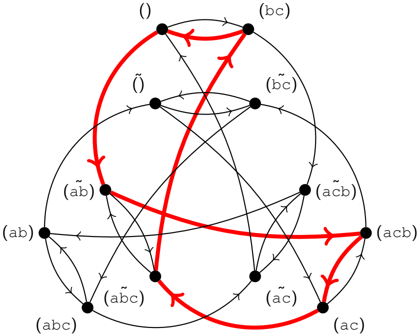


Long square-free walks



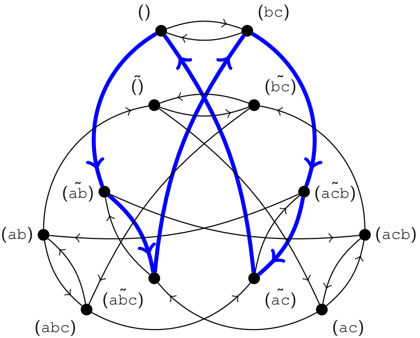
- Lemma: There are arbitrarily long square-free walks in D that begin and end at ().

LONG SQUARE-FREE WALKS



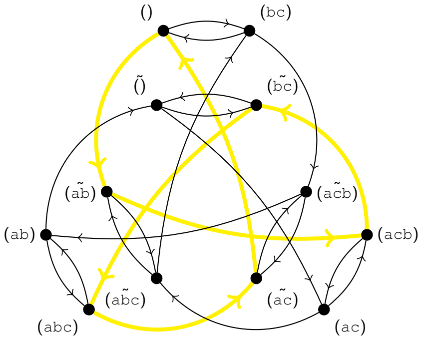
- Lemma: There are arbitrarily long square-free walks in D that begin and end at $()$.

Long square-free walks



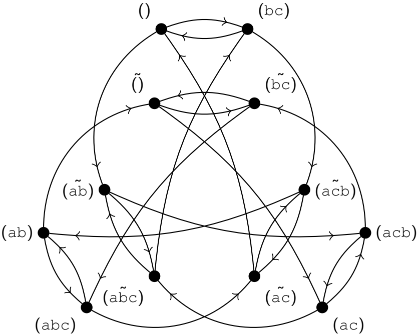
- Lemma: There are arbitrarily long square-free walks in *D* that begin and end at ().

LONG SQUARE-FREE WALKS



- Lemma: There are arbitrarily long square-free walks in D that begin and end at $()$.

LONG NEARLY EXTREMAL SQUARE-FREE WORDS



- Let f be the morphism that maps every vertex π to the corresponding word N_π.
- Lemma: If w is a square-free walk in D, then $f(w)$ is square-free, and hence nearly extremal square-free.

LONG EXTREMAL SQUARE-FREE WORDS

- Finally, there are two words P and S that will serve as "bookends":

$$P = \texttt{cbacbcabacbabcabacbcabcbacbc}$$
$$S = \texttt{acabcacbabcabacbcabcbacabacbcabcb}$$

- Lemma: If w is a square-free walk in D that begins and ends at (), then $Pf(w)S$ is extremal square-free.
- Conclusion: There are arbitrarily long extremal square-free words!

Further Questions

Q: Are there extremal square-free words of every sufficiently large length?

Further Questions

Q: Are there extremal square-free words of every sufficiently large length?

A: Yes.

- Theorem (Mol and Rampersad 2021): There is an extremal square-free word of length *k* over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ for every $k \geq 87$.
- In fact, there are many! The number grows exponentially in *k*.

Further Questions

Q: Are there extremal square-free words of every sufficiently large length?

A: Yes.

- Theorem (Mol and Rampersad 2021): There is an extremal square-free word of length k over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ for every $k \geq 87$.
- In fact, there are many! The number grows exponentially in k.

Q: Are there extremal square-free words over larger alphabets?

Further Questions

Q: Are there extremal square-free words of every sufficiently large length?

A: Yes.

- Theorem (Mol and Rampersad 2021): There is an extremal square-free word of length k over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ for every $k \geq 87$.
- In fact, there are many! The number grows exponentially in k.

Q: Are there extremal square-free words over larger alphabets?

A: Hmmmm...

Further Questions

Q: Are there extremal square-free words of every sufficiently large length?
A: Yes.

- Theorem (Mol and Rampersad 2021): There is an extremal square-free word of length k over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ for every $k \geq 87$.
- In fact, there are many! The number grows exponentially in k.

Q: Are there extremal square-free words over larger alphabets?
A: Hmmmm...

- Conjecture (GKN 2020): No.

FURTHER QUESTIONS

Q: Are there extremal square-free words of every sufficiently large length?
A: Yes.

- Theorem (Mol and Rampersad 2021): There is an extremal square-free word of length k over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ for every $k \geq 87$.
- In fact, there are many! The number grows exponentially in k.

Q: Are there extremal square-free words over larger alphabets?
A: Hmmmm...

- Conjecture (GKN 2020): No.
- Theorem (Hong and Zhang 2021): There are no extremal square-free words over alphabets of size 17 or greater.