Avoiding repetitions in words III

Narad Rampersad

Department of Mathematics and Statistics
University of Winnipeg

Non-constructive methods

- We focus on non-constructive methods for proving results on avoidability in words.
- for instance: the probabilistic method,
- or other counting arguments

Finitary and infinitary results

- we are looking for an infinite word avoiding a set S
- many of the techniques we will see only give arbitrarily large, finite words avoiding S
- the existence of an infinite word avoiding S can be obtained by a standard argument
- sometimes presented topologically as a compactness argument
- or derived combinatorially from König's tree lemma

König's Lemma (reformulated)

Theorem (König's Lemma)

Let X be an infinite set of finite words over an alphabet Σ. Then there is an infinite word $\mathbf{x}$ over Σ such that every factor of $\mathbf{x}$ is a factor of infinitely many words in X.

Comments on König's lemma

- König's lemma is itself a non-constructive result.
- Even if X is given effectively, the result does not give an explicit construction of the word $\mathbf{x}$.
- The result can be strengthened somewhat.

Uniformly recurrent words

- a word is recurrent if each of its factors occurs infinitely often
- it is uniformly recurrent if for each factor, the distance between consecutive occurrences of that factor is bounded
- König's result can be improved to get a uniformly recurrent word

A stronger version of König

Theorem (Furstenburg 1981)

Let X be an infinite set of finite words over an alphabet Σ. Then there is a uniformly recurrent word $\mathbf{x}$ over Σ such that every factor of $\mathbf{x}$ is a factor of infinitely many words in X.

Proved by Furstenburg using ergodic theory; combinatorial proof given by others, such as Justin and Pirillo; topological proof by Currie and Linek.

An early use of the probabilistic method

One of the earliest uses of the probabilistic method in combinatorics on words was to prove:

Theorem (Beck 1981)

For any real $\epsilon > 0$, there exist an integer N_ϵ and an infinite binary word $\mathbf{w}$ such that for every factor x of $\mathbf{w}$ of length $n > N_\epsilon$, all occurrences of x in $\mathbf{w}$ are separated by a distance at least $(2-\epsilon)^n$.

The Lovász local lemma

- The proof is based on a lemma from probabilistic combinatorics known as the Lovász local lemma.
- allows one to give lower bounds on the probability of an intersection of several events when there are dependencies among the events

Entropy compression

- Moser and Tardos (2010) gave an algorithmic version of the Lovász local lemma based on an argument known as entropy compression.
- led to many improvements on earlier results proved using the local lemma
- well-suited for applications to avoidability in words
- easier to use than the local lemma
- gives sharper results

An application of the method

Theorem (Grytczuk, Kozik, and Micek 2013)

For every sequence $L_1, L_2, \ldots$ of 4-element sets, there exists a squarefree word $s_1 s_2 \cdots$ such that $s_i \in L_i$ for all $i \geq 1$.

Squarefree words over 4 letters

- let's apply the method to show the existence of an infinite squarefree word over the alphabet $\{1, 2, 3, 4\}$
- there are squarefree words over a 3-letter alphabet, so this result is not optimal
- the idea is to give a randomized algorithm that attempts to generate a squarefree word
- then show that some sufficiently long execution of the algorithm must generate a long squarefree word

The algorithm

Input: n

```
 1: $S = \epsilon, i = 1$
 2: while $i \leq n$ do
 3:     randomly choose $r \in \{1, 2, 3, 4\}$ and append $r$ to $S$
 4:     let $S = s_1 s_2 \cdots s_i$
 5:     if $s_1 s_2 \cdots s_i$ is squarefree then
 6:         set $i$ to $i + 1$
 7:     else $s_1 s_2 \cdots s_i$ ends with a square $xx$
 8:         delete the second occurrence of $x$
 9:         set $i$ to $i - |x|$
10:     end if
11: end while
```

Analyzing the algorithm

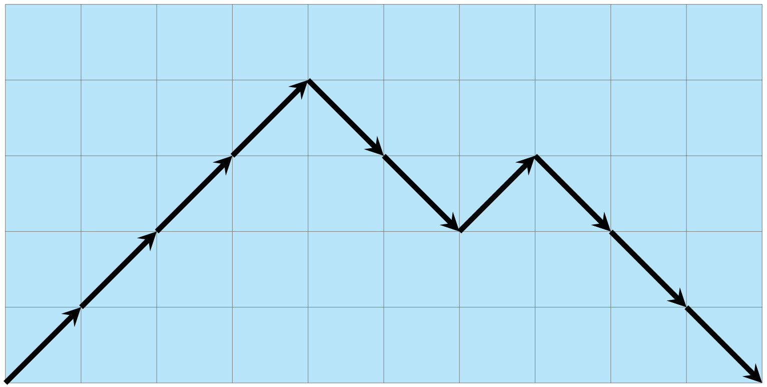
- fix n
- let M be the number of insertions (line 3) made during some execution of the algorithm
- let $r_1, r_2, \ldots, r_M$ be the sequence of random choices of letters inserted
- there are 4^M such sequences
- this sequence uniquely determines the execution of the algorithm

Another encoding of the execution

- we will describe the execution of the algorithm another way; i.e., by specifying:
  1. the sequence $S = s_1 s_2 \cdots s_i$ at the end of the execution, and
  2. the time and length of each deletion
- since each deletion consists of half of a square xx, the deleted block can be recovered from the first half still present in S
- with this information, we can describe the execution of the algorithm by running it backwards

Tracking the lengths of the deleted blocks

$1 \to 12 \to 121 \to 1212 \to 12 \to 123$



(if we terminate with a word of length i, we add i down-steps to the path)

Counting the number of paths

- the sequence of deletions is thus represented by a so-called Dyck path of length $2M$
- the number of such paths is the Catalan number

$$C_M = \binom{2M}{M}/(M+1)$$

- since the execution of the algorithm is uniquely determined by the final sequence S and this path, there are at most $(1/3)(4^{n+1} - 1)C_M$ executions

Getting a contradiction

- suppose the algorithm fails to produce a squarefree word of length n
- so for M arbitrarily large, every execution of the algorithm fails to terminate after M steps
- from our two different counts of the number of executions, we have $4^M \leq (1/3)(4^n - 1)C_M$

The inequality fails to hold

Thus,

$$\begin{aligned} 4^M &\leq \left(\frac{4^n - 1}{3}\right) C_M \\ &\ll 4^n \left(\frac{4^M}{M^{3/2}\sqrt{\pi}}\right), \end{aligned}$$

which is not possible for M sufficiently large.

The contradiction means that some execution of the algorithm terminates after producing a squarefree word of length n.

Getting an infinite squarefree word

- so for all n there is a squarefree word of length n over $\{1, 2, 3, 4\}$
- by König's lemma, there is an infinite squarefree word over 4 letters
- we can do better, since we know that there are ternary squarefree words, but the method is very useful for showing the avoidability of more complicated patterns

Avoiding repetitions in arithmetic progressions

Here are some other results proved using this method:

Theorem (Grytczuk, Kozik, Witkowski 2011)

Let $k \geq 1$. There are arbitrarily long sequences over $2k + 18\sqrt{k}$ symbols that avoid squares in every arithmetic subsequence whose common difference is in the set $\{1, \ldots, k\}$.

Non-repetitive games

- two players try to build a sequence
- on each turn, the player extends the sequence by one symbol
- Player 1 tries to avoid repetitions
- Player 2 tries to create repetitions
- Player 2 can always create a “trivial” repetition by repeating the last symbol played

Non-repetitive games

Theorem (Grytczuk, Kozik, Micek 2011)

Over a 6 symbol alphabet, Player 1 has a strategy to play forever while avoiding all except the “trivial” repetitions.

A criterion of Miller

Here is another criterion for avoidability.

Proposition (Miller 2011)

Let S be a set of non-empty words over a k-letter alphabet Σ. If there exists $c \in (1/k, 1)$ such that

$$\sum_{s \in S} c^{|s|} \leq kc - 1,$$

then there is an infinite word over Σ that avoids S.

Proof of Miller's Criterion

- Let $\Sigma = \{0, 1, \ldots, k-1\}$.
- Let
$$p = \sum_{s \in S} c^{|s|},$$
and suppose $p \le kc - 1$.

Proof of Miller's Criterion

- Let x be a word over Σ such that the quantity

$$w(x) := \sum_{s \in S} \sum \{c^{|u|} : |u| < |s| \text{ and } xu \text{ ends in } s\}$$

  is less than 1.
- If $x = \epsilon$ then the inner sum is empty and $w(x) = 0$.
- If $w(x) < 1$ then x avoids S.
- The quantity $w(x)$ is a measure of pending threats to an extension of x ending in some $s \in S$.

Proof of Miller's Criterion

- Given $w(x) < 1$ we show there is some $a \in \Sigma$ such that $w(xa) < 1$.
- The process can then continue indefinitely.
- Consider all possible extensions of x. We have

$$\sum_{0 \le a \le k-1} w(xa) = \frac{w(x)}{c} + \frac{p}{c}.$$

Proof of Miller's Criterion

- An extension a might match the first letter of an already pending threat.
- It might also be the first letter of a brand new threat.
- This increases the weight of the pending threats by a factor $1/c$ and adds p/c as the weight of the newly created threats.

Proof of Miller's Criterion

- We have $w(x) + p < 1 + p \leq kc$, so

$$\sum_{0 \leq a \leq k-1} w(xa) = \frac{w(x)}{c} + \frac{p}{c} < k.$$

- Therefore there exists a such that $w(xa) < 1$.
- We can therefore continue this process indefinitely to define an infinite word avoiding S.

A simple application of the criterion

Let S be any collection of binary words containing at most one word of length 5, one word of length 6, one word of length 7, etc. Then there is an infinite binary word that avoids S. We compute

$$\begin{aligned}\sum_{s \in S} c^{|s|} &\leq c^5 + c^6 + c^7 + \cdots \\ &= \frac{c^5}{1-c}.\end{aligned}$$

If $c = 0.6$ then $c^5/(1-c) = 0.1944 < 2c - 1 = 0.2$.

Showing the existence of squarefree words

As an example, let's show that there are infinite squarefree words over a 7 letter alphabet (a very weak result!).
Let $k = 7$ and let $S = \{xx : x \in \Sigma^*\}$. Let $c \in (1/7, 1)$ be a constant to be specified later.

$$\begin{aligned}\sum_{s \in S} c^{|s|} &= \sum_{i \geq 1} c^{2i} 7^i \\ &= \sum_{i \geq 1} (7c^2)^i \\ &= \frac{1}{1 - 7c^2} - 1.\end{aligned}$$

Satisfying the inequality

We need

$$\frac{1}{1-7c^2} - 1 \leq 7c - 1$$

If $c = 0.22$ then LHS is approx. 0.512 and RHS is 0.54.

By Miller's criterion there is an infinite squarefree word over 7 letters.

Avoiding long squarefree words

Similarly, we can show that long squares are avoidable over a binary alphabet.

Let $k = 2$ and let $S = \{xx : x \in \Sigma^*, |x| \geq 7\}$. Let $c \in (1/2, 1)$ be a constant to be specified later.

$$\begin{aligned}\sum_{s \in S} c^{|s|} &= \sum_{i \geq 7} c^{2i} 2^i \\ &= \sum_{i \geq 7} (2c^2)^i \\ &= \frac{128c^{14}}{1 - 2c^2}.\end{aligned}$$

Satisfying the inequality

We need

$$\frac{128c^{14}}{1-2c^2} \leq 2c-1$$

If $c = 0.54$ then LHS is approx. 0.055 and RHS is 0.08.

By Miller's criterion there is an infinite binary word containing no square xx with $|x| \geq 7$.

In fact, one can show constructively that 7 can be replaced by 3.

Words with high Kolmogorov complexity

One can also show that there are infinite words for which every factor has fairly high Kolmogorov complexity.

Random coin tosses

Which sequence of coin tosses is more random?

$$HTHHHTTHTTHHTHTHHTHTHT$$

$$HHHHHHHHHHHHHHHHHHHH$$

Each has the same probability: 2^{-20}.

Kolmogorov complexity

Define the Kolmogorov complexity $C(x)$ of a string x as the length of the shortest computer program (Turing machine) stored in binary that outputs x.
Choice of programming language (universal Turing machine) doesn't matter. Only affects the value of $C(x)$ by an additive constant.

Upper bound on $C(x)$

$$C(x) \leq |x| + O(1)$$

Here's a program that outputs x:

```
print 'x'
```

Its length is $|x|$ plus a constant.

Incompressible strings

- we view a program that outputs x as a "compressed" encoding of x
- a string x is incompressible (or Kolmogorov random) if $C(x) \geq |x|$

Existence of incompressible strings

Theorem

For each length n there is a string x of length n such that $C(x) \geq n$. (There are incompressible strings of every length.)

- there are 2^n binary strings of length n
- there are $1 + 2 + \cdots + 2^{n-1} = 2^n - 1$ binary strings of length $< n$
- at least one length n string has no shorter encoding

Prefix-free encodings

- problem with $C(x)$: $C(xy)$ can be less than $C(x)$
- prefix-free encodings: no encoding of one string is a prefix of the encoding of another string
- e.g., encode each binary string x as $1^{|x|}0x$
- takes up a little more space
- define the prefix Kolmogorov complexity $K(x)$ to be the length of the shortest prefix-free binary encoding of a computer program that outputs x

Subadditivity of prefix Kolmogorov complexity

$$K(xy) \leq K(x) + K(y) + O(1)$$

Kraft's Inequality

Theorem

Let X be a prefix-free collection of binary strings. Then

$$\sum_{x \in X} 2^{-|x|} \leq 1.$$

Proof of Kraft's Inequality

- for $x \in X$ let R_x be the subinterval of $[0, 1)$ consisting of all real numbers whose binary expansion begins with $0.x\ldots$
- R_x has length $2^{-|x|}$
- the prefix-freeness of X implies the R_x's are disjoint
- hence, the sum of their lengths is at most 1

Words with high Kolmogorov complexity

Theorem (Durand, Levin, Shen 2001)

Let $0 < \alpha < 1$. There is an infinite binary word $\mathbf{x}$ such that every factor u of $\mathbf{x}$ satisfies $K(u) > \alpha|u| - O(1)$, where $K(u)$ denotes the prefix Kolmogorov complexity of u.

Proving the Kolmogorov complexity result

We use Miller's criterion to avoid the low complexity words.

Define $b = -\log_2(1-\alpha) + 1$ and

$$S = \{s \in \{0,1\}^* : K(s) \leq \alpha|s| - b\}.$$

Set $c = 2^{-\alpha}$. Then

$$\sum_{s \in S} c^{|s|} = \sum_{s \in S} 2^{-\alpha|s|} \leq \sum_{s \in S} 2^{-K(s)-b} \leq 2^{-b} \sum_{s \in \{0,1\}^*} 2^{-K(s)} \leq 2^{-b},$$

where we have applied Kraft's Inequality in the last step.

We can verify that $2^{-b} < 2c - 1$, so by Miller's criterion there is an infinite word avoiding S.

Conclusion

- non-constructive methods have been very useful in proving avoidability results
- they have been applied to ordinary repetitions, fractional powers, patterns, approximate repetitions, "shuffle squares", etc.
- these methods do not seem applicable to abelian repetitions

The End