Applications of Walnut to problems of local periodicity

Narad Rampersad

Department of Mathematics and Statistics
University of Winnipeg

A major theme in combinatorics on words is the relationship between local periodicity and global periodicity of an infinite word.

The local periodicity at a postion i of an infinite word can be measured in terms of repetitions

1. starting at position i,
2. ending at position i,
3. centred at position i.

We will consider 2) and 3).

First, what types of repetitions do we consider?

- a square is a non-empty repetition xx
- a cube is a non-empty repetition xxx
- in general, an α-power (or word with exponent α) is a word whose length divided by its minimal period is α

- Let's first consider repetitions ending at positions i.
- Let $\phi \approx 1.618$ be the golden ratio.
- Shallit conjectured that any infinite word with the property that every sufficiently large position ends with a word of exponent $\geq \phi^2 \approx 2.618$ is necessarily ultimately periodic.
- This was confirmed by Mignosi, Restivo, and Salemi.

Theorem (Mignosi, Restivo, Salemi 1993)

An infinite word $\mathbf{w}$ is ultimately periodic if and only if there is a repetition of exponent $\geq \phi^2$ ending at all sufficiently large positions.

- Mignosi, Restivo, and Salemi showed that the constant ϕ^2 cannot be replaced with any smaller number.
- The Fibonacci word

$$0100101001001010010100100 \cdots ,$$

generated by iterating the substitution $0 \to 01, \ 1 \to 0$, witnesses this optimality.
- For simiplicity, lets consider cubes, rather than ϕ^2 powers.
- This theorem implies that there are infinitely many positions of the Fibonacci word that do not end in a cube.
- Can we characterize these positions?

- Let $\mathbf{cubes_f}$ be the infinite word whose n-th term is

$$\begin{cases} 1 \text{ if a cube ends at position } n \text{ of } \mathbf{f}, \\ 0 \text{ otherwise.} \end{cases}$$

- For any $n \geq 0$, let $(n)_F$ denote the canonical representation of n in the Fibonacci (Zeckendorf) numeration system.
- This is the numeration system where the place values are the Fibonacci numbers $1, 2, 3, 5, 8, 13, \ldots$, the digit set is $\{0, 1\}$, and we forbid consecutive 1's.

Theorem

There are arbitrarily long runs of 1's in $\mathbf{cubes_f}$. More precisely, the runs of 1's in $\mathbf{cubes_f}$ are characterized by the following: If $(i)_F$ has the form

$$(i)_F \in (10)^+0(0+10)(00)^*0w,$$

where $w \in 0(10)^*(\epsilon + 1)$ then $\mathbf{cubes_f}$ contains a run of 1's of length

- $F_{2n+2} - 1$, if $|w| = 2n$ for some $n \geq 0$,
- $F_{2n+3} - 1$, if $|w| = 2n + 1$ for some $n \geq 0$,

beginning at position i.

Theorem

The runs of 0's in $\mathbf{cubes_f}$ have lengths 1, 2, 3, 7, 8, and 13. The only run of length 13 occurs at the beginning of $\mathbf{cubes_f}$. For each of the other lengths (1, 2, 3, 7, and 8), there are infinitely many runs of that length in $\mathbf{cubes_f}$.

We can obtain this result purely by computer using a program called Walnut (developed by Jeffrey Shallit's student Hamoon Mousavi). Suppose we are given

- A finite automaton reading input n in base-k (or Fibonacci, etc.) and outputing the n-th term of a sequence $\mathbf{s}$; and,
- A formula φ in first-order logic involving variables, constants, quantifiers, logical operations, ordering, addition and subtraction of natural numbers, and indexing into $\mathbf{s}$.

- We can also multiply by a constant (this is just repeated addition), but we can't multiply two variables.
- If φ has no free variables, Walnut will output either that φ is either TRUE or FALSE.
- If φ has free variables, Walnut will produce an automaton that accepts the base-k (or Fibonacci, etc.) representations of the values of the free variables that satisfy φ.

The formula

$$\exists i \exists n,\ (n > 1) \wedge (j = i + 3n - 1) \wedge (\forall k,\ k < 2n \Rightarrow \mathbf{f}_{i+k} = \mathbf{f}_{i+k+n})$$

is satisfied whenever j is the ending position of a cube in $\mathbf{f}$.

Translated to Walnut, the command

```
eval fib_end_cubes
    "?msd_fib Ei En n > 1 & j = i+3*n-1 &
    (Ak k < 2*n => F[i+k] = F[i+k+n])":
```

produces an automaton that accepts the Zeckendorf representations of the positions at which a cube ends in $\mathbf{f}$.

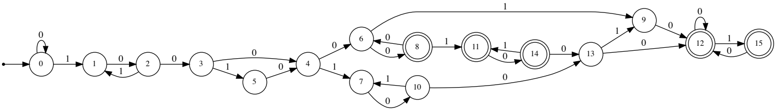

Figure: Automaton for ending positions of cubes in $\mathbf{f}$

The command

```
eval fib_end_cubes_run "?msd_fib n>=1 & (At t<n =>
    $fib_end_cubes(i+t)) & ~$fib_end_cubes(i+n) &
    (i=0|~$fib_end_cubes(i-1))":
```

produces an automaton that accepts the Zeckendorf representations of pairs (i, ℓ) such that there is a run of 1's in $\mathbf{cubes_f}$ of length ℓ starting at position i.

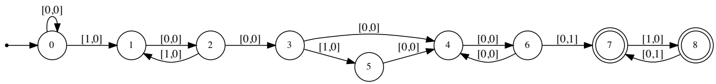

Figure: Automaton for runs of 1's in $\mathbf{cubes_f}$

- Now we examine the structure of this automaton:
- For an accepted pair (i, ℓ), we have

$$(i)_F = (10)^+ 0(0+10)(00)^* 0w, \text{ where } w \in 0(10)^*(\epsilon + 1).$$

- If $|w| = 2n$, then $(\ell)_F = (10)^n$ and so $\ell = F_{2n+2} - 1$.
- If $|w| = 2n+1$, then $(\ell)_F = (10)^n 1$ and so $\ell = F_{2n+3} - 1$.

We do the same for runs of 0's in $\mathbf{cubes_f}$:

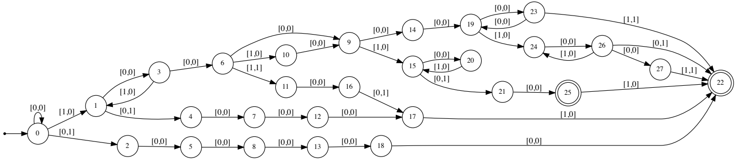

Figure: Automaton for runs of 0's in $\mathbf{cubes_f}$

We can project this automaton onto the second component of its input:

```
eval fib_no_cubes_run_length "?msd_fib Ei
    $fib_no_cubes_run(i,n)":
```

which produces

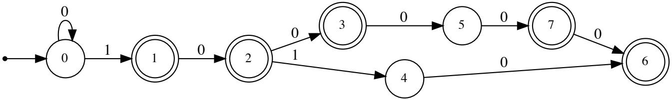

Figure: Automaton for lengths of runs of 0's in $\mathbf{cubes_f}$

- We see that the only possible run lengths are $\ell \in \{1, 2, 3, 7, 8, 13\}$.
- The command

```
eval tmp "?msd_fib Ai Ej j>i &
    $fib_no_cubes_run(j,1)":
```

  evaluates to TRUE, indicating that there are infinitely many runs of 0's of length 1.
- This is also the case for run lengths 2, 3, 7, and 8.
- For length 13 however, we get a result of FALSE.

The positions of the runs of length 7 and 8 have a simple structure:

Theorem

- The runs of 0's in $\mathbf{cubes_f}$ of length 8 begin at positions i where $(i)_F \in (10)^+0001$.
- The runs of 0's in $\mathbf{cubes_f}$ of length 7 begin at positions i where $(i)_F \in (10)^+01001$.

Theorem

The density of 0's in $\mathbf{cubes_f}$ is zero.

The following automaton gives the positions in $\mathbf{f}$ where no cube ends.

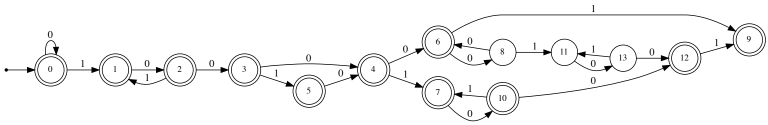

Figure: Automaton for positions in $\mathbf{f}$ where no cube ends

- It suffices to show that only polynomially many strings of length n are accepted.
- This can be seen directly from the structure of the automaton.
- This automaton does not have two cycles that can both mutually reach each other.
- Hence, the number of accepted strings of length n is polynomially bounded.

- What about other Sturmian words?
- Let $\mathbf{c}_\alpha$ be the characteristic Sturmian word with slope α.
- Let $\mathbf{cubes_w}$ be the binary word whose n-th term is 1 if $\mathbf{cubes_w}$ has a cube ending at position n, and 0 otherwise.
- Let $\mathrm{max\_no\_cubes}(\mathbf{w})$ denote the largest ℓ such that $\mathbf{cubes_w}$ contains infinitely many runs of 0's of length ℓ.

- We have see that $\mathrm{max\_no\_cubes}(\mathbf{f}) = 8$.
- Is it possible to determine $\mathrm{max\_no\_cubes}(\mathbf{c}_\alpha)$ from the continued fraction expansion of α?
- What is the least (resp. greatest) possible value of $\mathrm{max\_no\_cubes}(\mathbf{c}_\alpha)$ over all α? Is it 3 (resp. 8)?

Now let's consider repetitions centred at position i. The analogue of the theorem of Mignosi, Restivo, and Salemi is the following:

Theorem (Duval, Mignosi, Restivo 2001)

An infinite word $\mathbf{w}$ is ultimately periodic if and only if there is a square centred at all sufficiently large positions.

Again, the Fibonacci word is optimal: Duval, Mignosi, and Restivo showed that the Fibonacci word has a square centred at every position j, except when $j = F_n - 2$.

- In the second Lothaire book, Mignosi and Restivo (2002) use the following periodicity function to study repetitions centred at position i.
- Let $\mathbf{w} = w_0 w_1 w_2 \cdots$ be an infinite word.
- The periodicity function $p_{\mathbf{w}}(i)$ is the length of the shortest prefix u of $w_i w_{i+1} w_{i+2} \cdots$ such that either u is a suffix of $w_0 \cdots w_{i-1}$ (i.e., uu is a square centred at position i) or $w_0 \cdots w_{i-1}$ is a suffix of u, if such a word u exists.
- If no such u exists, then $p_{\mathbf{w}}(i) = \infty$.
- If $\mathbf{w}$ is recurrent, then $p_{\mathbf{w}}(i) < \infty$ for all i.

- Since the values of $p_{\mathbf{w}}(i)$ can fluctuate wildly, it is not that suitable as a complexity function.
- Mignosi and Restivo (2013) therefore defined the periodic complexity function $h_{\mathbf{w}}(i)$ as the average of the periodicity function:
- Let

$$P_{\mathbf{w}}(i) = \sum_{j=0}^{i-1} p_{\mathbf{w}}(j)$$

be the summatory function of $p_{\mathbf{w}}(i)$ and let

$$h_{\mathbf{w}}(i) = (1/i) P_{\mathbf{w}}(i)$$

for $i \geq 1$.

Mignosi and Restivo studied the periodicity function and the periodic complexity function for both the Thue–Morse word

$$\mathbf{t} = 0110100110010110\cdots$$

and the Fibonacci word

$$\mathbf{f} = 0100101001001010\cdots .$$

They proved that $h_{\mathbf{t}}(n) = \Theta(n)$ and $h_{\mathbf{f}}(n) = \Theta(\log n)$

The Thue–Morse word $\mathbf{t} = t_0 t_1 t_2 \cdots$ is defined by

$$t_i = \begin{cases} 0 & \text{if the number of } 1\text{'s in the binary} \\ & \text{representation of } i \text{ is even,} \\ 1 & \text{otherwise.} \end{cases}$$

Here are some initial values of $p_{\mathbf{t}}(i)$.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$p_{\mathbf{t}}(i)$	1	3	1	6	2	12	1	12	1	24	1	24	2	24	1	24

We can get a automaton that computes the binary representation of $p_{\mathbf{t}}(i)$ with the following Walnut commands (see Jeff Shallit's forthcoming book):

```
def tmEq "?msd_2 Ak (k<n) => T[i+k]=T[j+k]":

def tmRepWd "?msd_2 (i>=n & $tmEq(i,i-n,n)) |
    (n>i & $tmEq(0,n,i))":

def tmLocPer "?msd_2 (n>0) & $tmRepWd(i,n) &
    Am (m>0 & m<n) => ~$tmRepWd(i,m)":
```

This produces the automaton

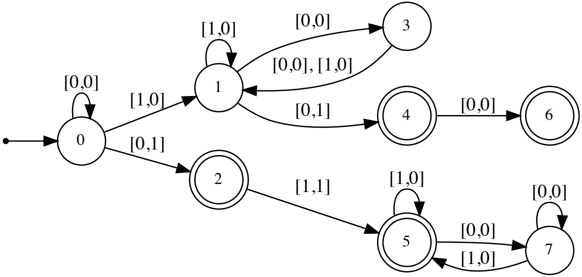

Figure: Automaton for the pair $(i, p_{\mathbf{t}}(i))$

By examining this automaton, one obtains the following:

Proposition

We have

- $p_{\mathbf{t}}(i) \in \{1, 2\}$ if i is even; and,
- $p_{\mathbf{t}}(i) = 3 \cdot 2^t$ if i is odd and $2^t \leq i < 2^{t+1}$.

We can then bound the summatory function of $p_{\mathbf{t}}(i)$.

Proposition

For $n \geq 1$, we have

$$\frac{3}{8}(n-1)^2 + \frac{n}{2} \leq P_{\mathbf{t}}(n) \leq \frac{3}{4}n^2 + n + 1.$$

We split the sum $P_{\mathbf{t}}(n) = \sum_{i=0}^{n-1} p(i)$ into even and odd indexed terms. For the even terms, we have

$$\frac{n}{2} \leq \sum_{\substack{i=0 \\ i \text{ even}}}^{n-1} p(i) \leq n+1.$$

For the odd terms, we have

$$\sum_{\substack{i=0 \\ i \text{ odd}}}^{n-1} p(i) \leq \sum_{\substack{i=0 \\ i \text{ odd}}}^{n-1} 3i \leq 3(n/2)^2 = \frac{3}{4}n^2.$$

and

$$\sum_{\substack{i=0 \\ i \text{ odd}}}^{n-1} p(i) \geq \sum_{\substack{i=0 \\ i \text{ odd}}}^{n-1} 3i/2 \geq (3/2)[(n-1)/2]^2 = \frac{3}{8}(n-1)^2.$$

Hence,

$$\frac{3}{8}(n-1)^2 + \frac{n}{2} \leq P_{\mathbf{t}}(n) \leq \frac{3}{4}n^2 + n + 1.$$

Dividing by n gives:

Theorem

For $n \geq 1$, we have

$$3n/8 - 1/4 \leq h_{\mathbf{t}}(n) \leq 3n/4 + 2.$$

In particular, we have $h_{\mathbf{t}}(n) = \Theta(n)$.

- In this case, we were fortunate that the automaton in for $p_{\mathbf{t}}(i)$ was rather simple.
- For more complicated sequences, this may not be the case, so next we explore other methods for analyzing the asymptotics of $P_{\mathbf{t}}(i)$.
- To apply these methods, we first need a linear representation for $p_{\mathbf{t}}(i)$.

A linear representation for $p_{\mathbf{t}}(i)$ is:

- an integer row vector v,
- an integer column vector w, and
- a pair of integer matrices M_0 and M_1, such that

$$p_{\mathbf{t}}(i) = vM_{i_{\ell-1}}M_{i_{\ell-2}}\cdots M_{i_0}w,$$

where $i_{\ell-1}i_{\ell-2}\cdots i_0$ is the binary representation of i.

Walnut can produce a linear representation for $p_{\mathbf{t}}(i)$ with the command

```
eval tmLocPer_enum i "?msd_2 En $tmLocPer(i,n) &
    m<n & ~$tmLocPer(i,m)":
```

The output of this command is a Maple worksheet containing the following values for v, w, M_0 and M_1.

$$v = [1, 0, 1, 0, 0, 0],$$

$$M_0 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}, \qquad M_1 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \end{bmatrix},$$

$$w = [1, 1, 0, 1, 1, 0]^T.$$

Next we determine the asymptotic behaviour of $P_{\mathbf{pd}}(n)$ and $h_{\mathbf{pd}}(n)$, where

$$\mathbf{pd} = 0100010101000100010001010100\cdots$$

is the period-doubling word, i.e., the fixed point of the morphism $0 \to 01$, $1 \to 00$.

Here are some initial values of $p_{\mathbf{pd}}(i)$:

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$p_{\mathbf{pd}}(i)$	1	2	4	1	1	8	2	2	2	2	16	1	1	4	4	1

Our goal is to show that $h_{\mathbf{pd}}(n) = \Theta(\log n)$ (i.e., its periodic complexity is rather more like that of the Fibonacci word than the Thue–Morse word).

We begin by using Walnut to compute the following linear representation for $p_{\mathbf{pd}}(i)$:

$$v = [1, 0, 0, 0, 0, 0],$$

$$M_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad M_1 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix},$$

$$w = [1, 1, 1, 1, 1, 1]^T.$$

Theorem

For $n \geq 1$, we have

$$P_{\mathbf{pd}}(n) \geq (1/3 \log_2 n - 1/18)n + 4/9,$$

$$P_{\mathbf{pd}}(n) \leq (4/3 \log_2 n + 22/9)n + 5/9,$$

and

$$1/3 \log_2 n - 1/18 \leq h_{\mathbf{pd}}(n) \leq 4/3 \log_2 n + 3.$$

Let $M = M_0 + M_1$. For $\ell \geq 0$, we have

$$\begin{aligned} P_{\mathbf{pd}}(2^\ell) &= \sum_{i<2^\ell} p(i) \\ &= \sum_{i_0,\dots,i_{\ell-1}\in\{0,1\}} vM_{i_{\ell-1}} \cdots M_{i_0} w \\ &= v(M_0 + M_1)^\ell w \\ &= vM^\ell w. \end{aligned}$$

To obtain a formula for $vM^\ell w$, we first compute the minimal polynomial of M:

$$m_M(x) = (x-2)^2(x+2)(x-1)(x+1).$$

It follows then that

$$vM^\ell w = (A + B\ell)2^\ell + C(-2)^\ell + D + E(-1)^\ell, \qquad (1)$$

for some constants $A, \ldots, E$.

We compute $P_{\mathbf{pd}}(2^\ell)$ for $\ell = 0, \dots, 4$, which gives the values $1, 3, 8, 21, 52$.

We then substitute these values into (1) to obtain a system of linear equations in the variables $A, \dots, E$, with solution

$$A = 5/9, B = 2/3, C = 0, D = 1/2, E = -1/18.$$

Thus, we have

$$P_{\mathbf{pd}}(2^\ell) = (5/9 + (2/3)\ell)2^\ell + 1/2 - 1/18(-1)^\ell,$$

and so

$$(5/9 + (2/3)\ell)2^\ell + 4/9 \leq P_{\mathbf{pd}}(2^\ell) \leq (5/9 + (2/3)\ell)2^\ell + 5/9.$$

Now write $2^\ell \leq n < 2^{\ell+1}$, so that $\ell \leq \log_2 n < \ell + 1$. We find

$$(1/3 \log_2 n - 1/18)n + 4/9 \leq P_{\mathbf{pd}}(n) \leq (4/3 \log_2 n + 22/9)n + 5/9,$$

and

$$1/3 \log_2 n - 1/18 \leq h_{\mathbf{pd}}(n) \leq 4/3 \log_2 n + 3.$$

We can determine the asymptotic growth of $h_{\mathbf{x}}(n)$ for other automatic sequences $\mathbf{x}$ by first using Walnut to compute a linear representation for $p_{\mathbf{x}}(i)$, and then applying techniques of Dumas or Heuberger–Krenn.

Let

$$\mathbf{rs} = r_0 r_1 r_2 \cdots = 0001001000011101 \cdots$$

be the Rudin–Shapiro sequence, defined by

$$r_i = \begin{cases} 0 & \text{if the number of } 11\text{'s in the binary} \\ & \text{representation of } i \text{ is even,} \\ 1 & \text{otherwise.} \end{cases}$$

The linear representation for $p_{\mathbf{rs}}(i)$ computed by Walnut gives 31×31 matrices M_0 and M_1.

The asymptotic techniques of Dumas or Heuberger–Krenn then give:

Theorem

We have

$$P_{\mathbf{rs}}(n) = n^2\Phi_{40}(\{\log_2 n\}) + O(n\log^2 n),$$

and

$$h_{\mathbf{rs}}(n) = n\Phi_{40}(\{\log_2 n\}) + O(\log^2 n),$$

for some 1-periodic continuous function Φ_{40}.

Walnut can be downloaded here:

`https://cs.uwaterloo.ca/~shallit/walnut.html`

The End