Using Walnut: Recent results in combinatorics on words and number theory

Narad Rampersad

Department of Mathematics and Statistics
University of Winnipeg

(Joint work with Jeffrey Shallit)

We explore the use of `Walnut`, a theorem prover for the class of automatic sequences (sequence computed by finite automata), to obtain some results in combinatorics on words and number theory.

We first show how `Walnut` can be used to obtain congruences for combinatorial sequences like the Catalan numbers

$$1, 1, 2, 5, 14, 42, 132, 429, 1430, \ldots$$

which count, among other things, the number of strings of properly nested parentheses of length $2n$, or the number of binary trees on n vertices.

We will be working with base-p expansions. If

$$n = n_0 + n_1 p + n_2 p^2 + \cdots + n_r p^r$$

we write

$$(n)_p = n_0 n_1 n_2 \cdots n_r$$

for the the base-p expansion of n written least-significant-digit first.

Let us start with the binomial coefficients:

Theorem (Lucas 1878)

Let p be prime and let

$$n = n_0 + n_1 p + n_2 p^2 + \cdots + n_r p^r$$
$$k = k_0 + k_1 p + k_2 p^2 + \cdots + k_r p^r.$$

Then

$$\binom{n}{k} \equiv_p \prod_{i=0}^{r} \binom{n_i}{k_i}.$$

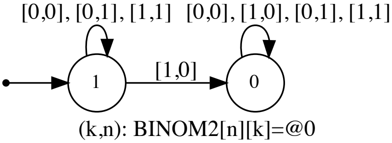
(By convention $\binom{n}{k} = 0$ if $n < k$.)

- Take $p = 2$. We see that $\binom{n}{k}$ is even exactly when there is some i such that $(k_i, n_i) = (1, 0)$.
- e.g.,

$$293930 = \binom{21}{12} \equiv_2 \binom{1}{0}\binom{0}{0}\binom{1}{1}\binom{\mathbf{0}}{\mathbf{1}}\binom{1}{0} \equiv_2 0$$

$$51895935 = \binom{29}{12} \equiv_2 \binom{1}{0}\binom{0}{0}\binom{1}{1}\binom{1}{1}\binom{1}{0} \equiv_2 1$$

- This can be checked with a finite automaton.

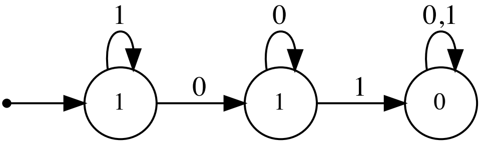


- The machine reads $(k, n)_2$, digit-by-digit, and follows the arcs labeled by each pair of digits read.
- If the machine ends in the state labeled 1, then $\binom{n}{k}$ is odd; otherwise it is even.

The sequence of Catalan numbers

$$C_n = \frac{1}{n+1}\binom{2n}{n} = \binom{2n}{n} - \binom{2n}{n-1}$$

modulo p can also be computed with a finite automaton: For $p = 2$ we get



Interpreting the automaton gives the following folklore theorem:

Theorem

C_n is odd iff $(n)_2 = 1^k 0^j$; i.e., iff $n = 2^k - 1$.

(Here 1^k means a string of k 1's and 0^j means a string of j 0's.)

- Rowland and Zeilberger and Rowland and Yassawi gave different algorithms to produce automata for the Catalan numbers modulo p, the Motzkin numbers modulo p, the Delannoy numbers modulo p, etc.
- Let's now look at the Catalan numbers C_n modulo 3. (Alter and Kubota (1973) studied the general case $C_n \bmod p$.)
- Let $\mathbf{c}_3 = (C_n \bmod 3)_{n\geq 0}$.

Theorem (Deutsch and Sagan 2006)

The runs of 0's in $\mathbf{c}_3$ begin at positions n where either

$$(n)_3 = 21^i \text{ or } (n)_3 = 21^i0\{0,1\}^j,\ i \geq 1,\ j \geq 0,$$

and have length $(3^{i+2} - 3)/2$.

Theorem cont'd. (Deutsch and Sagan 2006)

The blocks of non-zero values in $\mathbf{c}_3$ are given by the following:

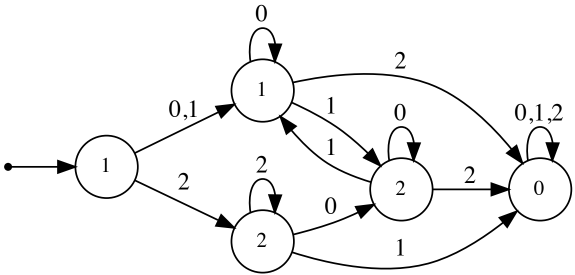
- The block 11222 occurs at position 0.
- The block 111222 occurs at all positions n where $(n)_3 = 2^i 0w$ for some $i \geq 2$ and some $w \in \{0,1\}^*$ that contains an odd number of 1's.
- The block 222111 occurs at all positions n where $(n)_3 = 2^i 0w$ for some $i \geq 2$ and some $w \in \{0,1\}^*$ that contains an even number of 1's.

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 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 representations of the values of the free variables that satisfy φ.

Applying the Rowland–Zeilberger method gives the automaton



for $\mathbf{c}_3$

which is rather more complicated than the modulo 2 automaton.

The formula

$$\varphi = \exists i \forall j((j \geq 0 \wedge j < 4) \Rightarrow \mathbf{c}_3(i+j) = 1)$$

asserts that there is a "run" of at least four 1's in $\mathbf{c}_3$.

In `Walnut`'s language, this is

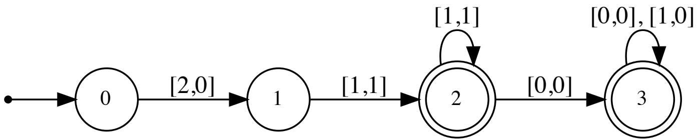
```
eval run4ones "?lsd_3 Ei Aj ((j>=0 & j<4) =>
    CAT3[i+j]=@1)":
```

and evaluates to "FALSE".

For the runs of 0's we use the `Walnut` command

```
eval cat3max0 "?lsd_3 n>=1 &
    (At t<n => CAT3[i+t]=@0) &
    CAT3[i+n]!=@0 & (i=0|CAT3[i-1]!=@0)":
```

which produces the automaton



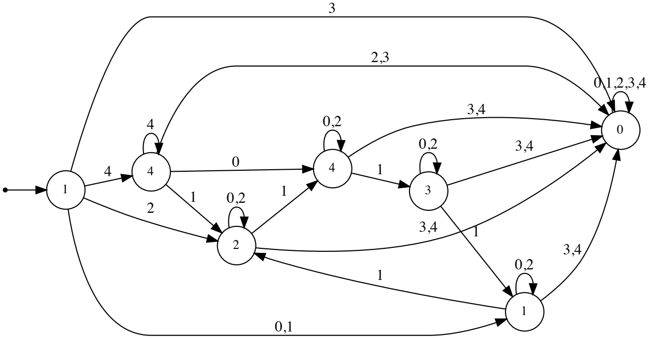
Examining the transition labels of the first component of the input gives the claimed representation for the starting positions of the runs of 0's

$$(i)_3 = 21^k \text{ or } (i)_3 = 21^k0\{0,1\}^j$$

and examining the transition labels of the second component gives the claimed length

$$(n)_3 = 01^k; \text{ i.e., } n = (3^{k+2} - 3)/2.$$

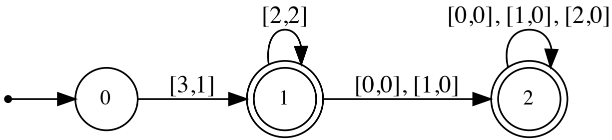
For $p = 5$, the Rowland–Zeilberger method gives the automaton



for

$\mathbf{c}_5 := C_n \bmod 5$.

Using Walnut, one can obtain the following automaton for the runs of 0's in $\mathbf{c}_5$:



From this automaton we derive:

Theorem

The runs of 0's in $\mathbf{c}_5$ begin at positions n where either

$$(n)_5 = 32^i \text{ or } (n)_5 = 32^i\{0,1\}\{0,1,2\}^j,\ i \geq 0,\ j \geq 0,$$

and have length $(5^{i+2} - 3)/2$.

We can easily characterize the non-zero blocks in $\mathbf{c}_5$ as well.

- We also obtained similar results for the Motzkin numbers modulo 3 and 5 as well.
- Ŵalnut can be used on any k-automatic sequence; i.e., any sequence whose n-th term can be computed by an automaton reading n in base-k as its input.
- Let's consider a new automatic sequence.
- In the rest of the talk, binary representations will be most-significant-digit first.

The Rudin-Shapiro coefficients

$$(a(n))_{n\geq 0} = (1, 1, 1, -1, 1, 1, -1, 1, \ldots)$$

form an infinite sequence of ± 1 defined recursively by the identities

$$a(2n) = a(n)$$
$$a(2n+1) = (-1)^n a(n)$$

and the initial condition $a(0) = 1$.

- The sequence $a(n)$ was introduced independently by Golay (1949), Rudin (1949), and Shapiro (1952).
- Rudin's motivation was the study of the absolute value of certain Fourier series; Golay was interested in optics.
- The function $a(n)$ can also be defined as $a(n) = (-1)^{r_n}$, where r_n counts the number of (possibly overlapping) occurrences of 11 in the binary representation of n.

Brillhart and Morton (1978) studied sums of these coefficients, and defined the two sums

$$s(n) = \sum_{0 \le i \le n} a(i) \qquad t(n) = \sum_{0 \le i \le n} (-1)^i a(i). \tag{1}$$

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$s(n)$	1	2	3	2	3	4	3	4	5	6	7	6	5	4
$t(n)$	1	0	1	2	3	2	1	0	1	0	1	2	1	2

Table: First few values of $s(n)$ and $t(n)$.

- Brillhart and Morton proved many properties of these sums; typically by a tedious induction.
- We show how to replace nearly all of these inductions with techniques from logic and automata theory.
- The Rudin-Shapiro sequence is 2-automatic and therefore also 4-automatic.

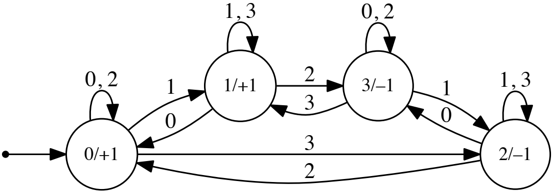


Figure: Base-4 automaton for the Rudin-Shapiro sequence

- States are labeled state number/output.
- The automaton reads the digits of the base-4 representation of n, starting with the most significant digit.
- Leading zeros in the inputs are allowed.

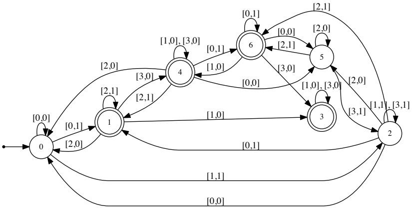
The main accomplishment of Brillhart and Morton's paper was proving the following inequalities:

Theorem (Brillhart & Morton)

For $n \geq 1$ we have

$$\sqrt{3n/5} \leq s(n) \leq \sqrt{6n}$$
$$0 \leq t(n) \leq \sqrt{3n}.$$

- To establish the inequalities for $s(n)$ and $t(n)$ we first determine automata that accept the pairs $(n, s(n))$ and $(n, t(n))$.
- In order for the automata to be able to process n and $s(n)$ in parallel, it turns out that we need to represent n in base-4 and $s(n)$ and $t(n)$ in base-2.



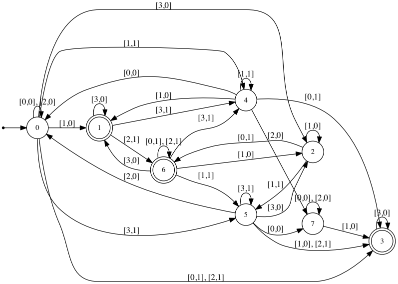


Figure: Synchronized automata for $s(n)$ (top) and $t(n)$ (bottom).

- To prove the inequalities we need to compare n to $s(n)$, but these numbers are now represented in different bases.
- We deal with this by defining a kind of "pseudo-square" function as follows: $m(n) = [(n)_2]_4$.
- In other words, m sends n to the integer obtained by interpreting the base-2 expansion of n as a number in base 4.
- We do this with the automaton `link42`:

```
reg link42 msd_4 msd_2 "([0,0]|[1,1])*":
```

- It's not hard to show that

$$(n^2 + 2n)/3 \leq m(n) \leq n^2.$$

We can now prove:

Lemma

For $n \geq 1$ we have $\frac{3n+7}{5} \leq m(s(n)) \leq 3n+1$, and the upper and lower bounds are tight.

We use the Walnut code

```
def maps "?msd_4 Ex $rss(n,x) & $link42(y,x)":
eval ms_lowerbnd "?msd_4 An,y (n>=1 & $maps(n,y))
  => y<=3*n+1":
eval ms_upperbnd "?msd_4 An,y (n>=1 & $maps(n,y))
  => 3*n+7<=5*y":
```

Tightness can be easily checked with Walnut.

Corollary

For $n \geq 1$ we have

$$s(n) \geq \sqrt{\frac{3n+7}{5}}.$$

- As a consequence, we get one of the claimed lower bounds.
- We simply put the bounds $m(s(n)) \leq s(n)^2$ and $\frac{3n+7}{5} \leq m(s(n))$ together to get $\frac{3n+7}{5} \leq s(n)^2$.
- Note that our lower bound is actually slightly stronger than that of Brillhart-Morton!

- The upper bound $s(n) \leq \sqrt{6n}$ is more difficult.
- If $m(s(n)) \leq 2n$, then the result follows immediately from the inequality $(n^2 + 2n)/3 \leq m(n)$.
- We can easily compute the exceptional set of n for which $m(s(n)) > 2n$: the binary representations of these n have the form

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

- The analysis of these exceptional values is somewhat technical (but still much easier than the original analysis of Brillhart and Morton!)

`Walnut` can be downloaded here:

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

The End