# Rudin-Shapiro Sums Via Automata Theory and Logic

Narad Rampersad

Department of Mathematics and Statistics
University of Winnipeg

(Joint work with Jeffrey Shallit)

The Rudin-Shapiro coefficients

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

form an infinite sequence of $\pm 1$ 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

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

| $n$ | 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.

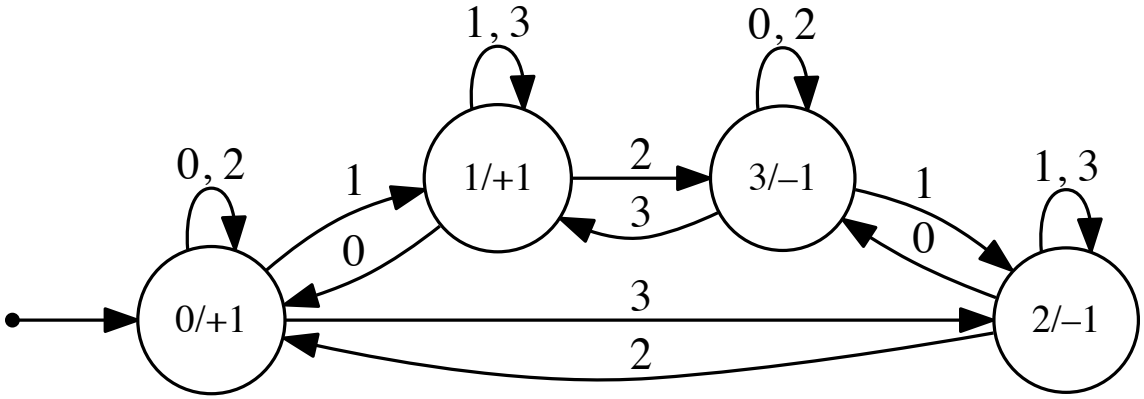


Figure: DFAO computing the Rudin-Shapiro function, in base $4$.

- 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.

Our main tool is `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 $\varphi$ 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 $\varphi$ has no free variables, `Walnut` will output either that $\varphi$ is either TRUE or FALSE.
- If $\varphi$ has free variables, `Walnut` will produce an automaton that accepts the base-$k$ representations of the values of the free variables that satisfy $\varphi$.

- To deal with $s(n)$ and $t(n)$ we first determine DFAO's that accept the relations $(n, s(n))$ and $(n, t(n))$.
- To do this, we represent $n$ in base-$4$ and $s(n)$ and $t(n)$ in base-$2$.
- We say that $s(n)$ and $t(n)$ are $(4, 2)$-synchronized sequences.

- Why use base-$4$ for $n$ and base-$2$ for the values of $s$ and $t$?
- It turns out that $s(n)$ and $t(n)$ grow like $\sqrt{n}$;
- hence, for an automaton to process $n$ and $s(n)$ in parallel, length considerations show that the base of representation for $n$ must be the square of that for $s(n)$ and $t(n)$.

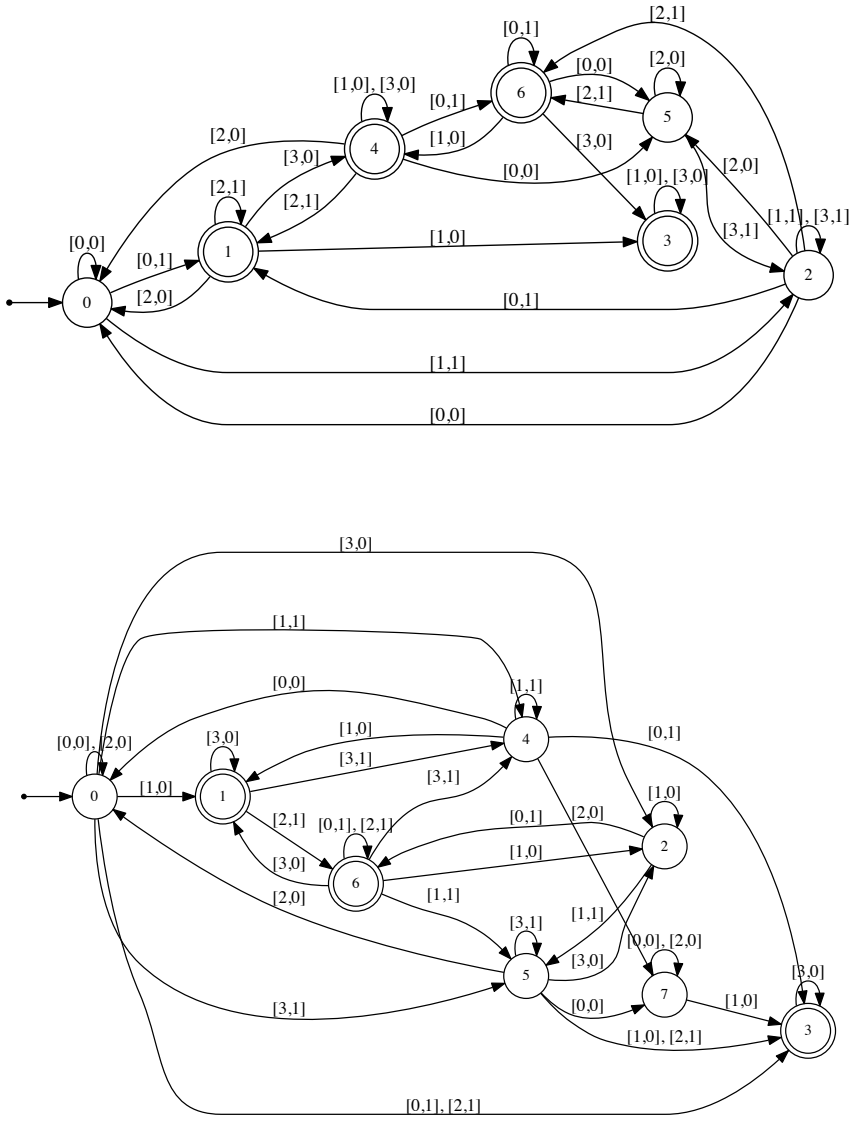


Figure: Synchronized automata for $s(n)$ (top) and $t(n)$ (bottom).

- The automata are "guessed" using the Myhill–Nerode Theorem.
- We prove the correctness of each automaton by induction on $n$, using Walnut itself to verify the induction step.
- We also use Walnut to verify that the relations computed by these automata are actually functions and that $t(n) \geq 0$ for all $n$.

The Walnut commands to verify the correctness of the automaton `rss` for $s(n)$ are:

```
eval test1 "?msd_4 An,y ($rss(n,y) & RS4[n+1]=@1)
  => $rss(n+1,y+1)":
eval test2 "?msd_4 An,y ($rss(n,y) & RS4[n+1]=@-1)
  => $rss(n+1,y-1)":
```

Brillhart & Morton begin by proving the following identities:

## Theorem (Brillhart & Morton)

We have

$$s(2n) = s(n) + t(n-1), \qquad (n \geq 1); \tag{2}$$

$$s(2n+1) = s(n) + t(n), \qquad (n \geq 0); \tag{3}$$

$$t(2n) = s(n) - t(n-1), \qquad (n \geq 1); \tag{4}$$

$$t(2n+1) = s(n) - t(n), \qquad (n \geq 0). \tag{5}$$

We use the following Walnut commands:

```
eval eq2 "?msd_4 An,x,y,z (n>=1 & $rss(2*n,x) &
   $rss(n,y) & $rst(n-1,z)) => ?msd_2 x=y+z":
eval eq3 "?msd_4 An,x,y,z ($rss(2*n+1,x) &
   $rss(n,y) & $rst(n,z)) => ?msd_2 x=y+z":
eval eq4 "?msd_4 An,x,y,z (n>=1 & $rst(2*n,x) &
   $rss(n,y) & $rst(n-1,z)) => ?msd_2 x+z=y":
eval eq5 "?msd_4 An,x,y,z ($rst(2*n+1,x) &
   $rss(n,y) & $rst(n,z)) => ?msd_2 x+z=y":
```

and Walnut returns TRUE for all of them.

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

## Theorem (Brillhart & Morton)

For $n \geq 1$ we have

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

- Trying to prove these results by directly translating the claims into `Walnut` leads to two difficulties:
- First, automata cannot compute squares or square roots.
- Second, our synchronized automata work with $n$ expressed in base $4$, but $s(n)$ and $t(n)$ are expressed in base $2$, and `Walnut` cannot directly compare arbitrary integers expressed in different bases.

- First, we define 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_upperbnd "?msd_4 An,y (n>=1 & $maps(n,y))
  => y<=3*n+1":
eval ms_lowerbnd "?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!

- To compare, Brillhart and Morton's proof of the lower bound for $s(n)$ spans three pages.
- They first introduce a new function $\omega(k)$ which gives the largest value of $n$ for which $s(n) = k$.
- They prove several lemmas about this new function by some non-trivial induction proofs.

TABLE 4

| $k$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $\omega(k)$ | 0 | 3 | 6 | 15 | 26 | 27 | 30 | 63 | 106 | 107 |

**Lemma 5.**

$$\omega(2n) = 4\omega(n) + 3, \qquad n \geq 1. \tag{26}$$

$$\omega(2n+1) = 4\omega(n+1) + 2, \quad n \geq 2,\ n+1 \neq 2^r,\ r \geq 2. \tag{27}$$

*Proof:* The proof of (26) is not hard. Note first that $s(n)$ and $n$ have opposite parity, so that $\omega(2n)$ must be odd. Hence we have either that $\omega(2n) = 4m+1$ or $\omega(2n) = 4m+3$, for some $m \geq 0$. The first case is impossible, because by (11), $s(4m+1) = s(4m+3) = 2n$, so $4m+1$ cannot be the largest argument of $s$ to give $2n$. Thus $\omega(2n) = 4m+3$. Then (11) implies that $2n = s(4m+3) = 2s(m)$, so that $s(m) = n$. If there were an $m_1 > m$ with $s(m_1) = n$, then $s(4m_1+3) = 2n$ and $4m_1 + 3 > 4m + 3 = \omega(2n)$ would contradict the definition of $\omega(2n)$. Thus $\omega(n) = m$, and $\omega(2n) = 4\omega(n) + 3$.

The proof of (27) is much trickier. To see how to approach the proof, let's first note one consequence of the formula. If (27) is true, then certainly

$$s(4\omega(n+1) + 2) = s(\omega(2n+1)) = 2n+1.$$

How might we prove just this much? Formula (12) gives

$$\begin{aligned} s(4\omega(n+1)+2) &= 2s(\omega(n+1)) + (-1)^{\omega(n+1)} a(\omega(n+1)) \\ &= 2(n+1) + (-1)^n a(\omega(n+1)), \end{aligned}$$

and so $s(4\omega(n+1)+2) = 2n+1$ if and only if $a(\omega(n+1)) = (-1)^{n+1}$ (when $n+1$ is not a power of 2). This shows that to prove (27) we must consider the formula

$$a(\omega(n)) = (-1)^n, \qquad n \geq 3,\ n \neq 2^r,\ r \geq 2. \tag{28}$$

Since induction has worked so often before, it is worth trying to prove (28) by induction as well. This is what we do now. Formula (28) holds for $n = 3$, since $a(\omega(3)) = a(6) = -1 = (-1)^3$. Assume that (28) has been proved for all the integers $m$ for which $3 \leq m < 2n+1$, for some $n \geq 2$. We proceed to prove it for $2n+1$ and $2n+2$. There are two cases to consider, because of the excluded values in (28).

Case 1: Suppose that $2n+2 \neq 2^r$, for any $r \geq 3$. Since we have already proved (26), we can use that formula and the defining formulas (1) for $a(n)$ to compute $a(\omega(2n+2))$:

$$\begin{aligned} a(\omega(2n+2)) &= a(4\omega(n+1)+3) = -a(2\omega(n+1)+1) \\ &= (-1)^{1+\omega(n+1)} a(\omega(n+1)) \\ &= (-1)^{1+n}(-1)^{n+1} = (-1)^{2n+2}; \end{aligned}$$

this computation uses the fact that the parities of $\omega(n+1)$ and $n+1$ are opposite, and the induction assumption ($n+1 < 2n+1$ and $n+1$ is not a power of 2). This proves (28) for $2n+2$. Before considering (28) for $2n+1$, we need a

formula for $\omega(2n+1)$. From what we have just shown it is easy to find a good candidate for $\omega(2n+1)$, since

$$s(\omega(2n+2)-1) = s(\omega(2n+2)) - a(\omega(2n+2)) = (2n+2)-1 = 2n+1.$$

Thus we might guess that $\omega(2n+1) = \omega(2n+2)-1$. If there were an $m > \omega(2n+2)-1$ for which $s(m) = 2n+1$, then because the sequence $\{s(m), m \geq 0\}$ goes to infinity by steps of $\pm 1$, there would have to be an integer $m' > m$ for which $s(m') = 2n+2$. But then $m' \geq m+1 > \omega(2n+2)$ would give a contradiction. Hence our guess was correct, and $\omega(2n+1) = \omega(2n+2)-1 = 4\omega(n+1)+2$. This proves (27), since in this case $n+1$ is not equal to a power of 2.

Now (28) follows for the value $2n+1$, since

$$\begin{aligned} a(\omega(2n+1)) &= a(4\omega(n+1)+2) = a(2\omega(n+1)+1) \\ &= (-1)^{\omega(n+1)} a(\omega(n+1)) \\ &= (-1)^n (-1)^{n+1} = (-1)^{2n+1}. \end{aligned}$$

Case 2: If $2n+2 = 2^r$, for some $r \geq 3$, then to complete the induction we have to prove (28) only for the value $2n+1 = 2^r - 1$. Here we need the fact that $\omega(2^r-1) = 2^{2r-1}-2$. To see this, first note that

$$s(2^{2r-1}-2) = s(2^{2r-1}-1) - a(2^{2r-1}-1) = 2^r - 1$$

by Lemma 4 and the fact that there are $2r-2$ pairs of consecutive 1's in the binary expansion of $2^{2r-1}-1$. Furthermore, it can be proved by induction that $s(m) \geq 2^r$ for $2^{2r-1} \leq m \leq 2^{2r}-1$ (with equality if and only if $m = 2^{2r}-1-\sum_{i=0}^{r-2}\varepsilon_i 2^{2i+1}$, where $\varepsilon_i = 0$ or 1). This, together with Theorem 1a (take $k \geq r$), shows that $s(m) \geq 2^r$ when $m > 2^{2r-1}-2$, and hence that $\omega(2^r-1) = 2^{2r-1}-2$, as claimed.

It follows that $a(\omega(2n+1)) = a(\omega(2^r-1)) = a(2^{2r-1}-2) = (-1)^{2r-3} = (-1)^{2n+1}$, and this completes the proof of (28). With (28) we have also completely proved (27) as well. ■

Looking back over this proof, we see that we were led to (28) by considering possible consequences of (27), but then proving (28) gave us a complete proof of (27) as a by-product: formula (27) is implied by (28) at the value $2n+2$. Actually, if we think of the induction proof as an argument that proceeds step-by-step through the positive integers, then the two formulas (27) and (28) are really *intertwined*, since (28) at $2n+2$ is used to establish (27), which is used in turn to prove (28) at $2n+1$. It is surprising that such intricate arguments are required to establish fairly simple recursion formulas.

After we had found the recursion formulas in Lemma 5, it seemed we were no closer to a proof of (25). However, we started to look for more patterns in the table by taking differences between consecutive values of $\omega(n)$, one of the standard ways of spotting possible formulas. Taking differences of the first 25 terms of the $\omega$ sequence gives:

| $n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\omega(n+1)-\omega(n)$ | 3 | 3 | 9 | 11 | 1 | 3 | 33 | 43 | 1 | 3 | 1 | 11 |

| $n$ | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\omega(n+1)-\omega(n)$ | 1 | 3 | 129 | 171 | 1 | 3 | 1 | 11 | 1 | 3 | 1 | 43 |

What strange numbers! For long stretches the difference is 1 at odd integers, and then it skyrockets. At even integers $n$, the difference takes on the values $3, 11, 3, 43$, except at powers of 2, where it also suddenly increases. Powers of 2! Suddenly we see that the difference depends only on the power of 2 dividing $n$, except when $n$ is 1 less than a power of 2, a wrinkle that fits with the recursion formulas in Lemma 5. We also see that the values $1, 3, 11, 43, 171$ satisfy a recursion: each value is 4 times the preceding value minus 1. When we solve the recursion for these values and investigate the wrinkle more closely, we find the following remarkable formulas.

**Lemma 6.** *a) If* $n = 2^\alpha(2m+1)$, *for* $m \geq 0$, $\alpha \geq 0$, *then* $\omega(n+1) - \omega(n) = (2^{2\alpha+1}+1)/3$, *unless* $\alpha = 0$ *and* $n = 2^s - 1$, $s \geq 1$. *b) If* $\alpha = 0$ *and* $n = 2^s - 1$, $s \geq 1$, *then* $\omega(n+1) - \omega(n) = 2^{2s-1} + 1$.

We omit the details of the proof, and note only that part a) can be proved by induction on $\alpha$, using (26), (27), and the special values $\omega(2^s - 1) = 2^{2s-1} - 2$ and $\omega(2^{s+2} - 2) = 2^{2s+3} - 5$.

Once we have a formula for the difference $\omega(n+1) - \omega(n)$, we are close to finding a formula for $\omega(k)$, since $\sum_{n=1}^{k-1}\{\omega(n+1) - \omega(n)\} = \omega(k) - \omega(1) = \omega(k)$. Summing up the expressions in Lemma 6 leads to the following explicit formula.

**Theorem 4.** *If* $2^r \leq k \leq 2^{r+1} - 1$, $r \geq 0$, *then*

$$\omega(k) = k - 1 + \frac{1}{3}(2^{2r+1} - 2) + 2\sum_{i=0}^{r-1}\left[\frac{k-1}{2^{i+1}}\right]2^{2i}.$$

We leave the somewhat technical details of the proof to the reader. This formula follows directly from Lemma 6, but may also be proved by a straightforward induction proof (on $k$) using only Lemma 5 and the fact that $\omega(2^r - 1) = 2^{2r-1} - 2$, $r \geq 0$. (See [**3**, Satz 1].)

Will this formula give us the lower bound we want?

**Theorem 5.** *For* $k \geq 1$, $k/\sqrt{\omega(k)} > \sqrt{3/5}$.

*Proof:* Assume that $2^r \leq k \leq 2^{r+1} - 1$, $r \geq 0$. By the formula for $\omega(k)$ we have

$$\begin{aligned}
3\omega(k) &= 3k - 3 + 2^{2r+1} - 2 + 6\sum_{i=0}^{r-1}\left[\frac{k-1}{2^{i+1}}\right]2^{2i} \\
&\leq 3k - 3 + 2^{2r+1} - 2 + 3(k-1)\sum_{i=0}^{r-1}2^i \\
&= 3k - 3 + 2^{2r+1} - 2 + 3(k-1)(2^r - 1) \\
&< 3k - 3 + 2k^2 - 2 + 3(k-1)k = 5k^2 - 5 < 5k^2
\end{aligned}$$

and the inequality of the theorem follows immediately. It worked!

**Corollary.** *For* $n \geq 1$, $s(n)/\sqrt{n} > \sqrt{3/5}$.

To summarize, we may combine Theorems 2, 3, and 5 in the following explicit result.

- 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 a little more technical (the details are in the paper), but still much easier than the complication induction of Brillhart and Morton.

One of the most fun properties of the Rudin-Shapiro summation function $s(n)$ is:

**Theorem (Brillhart & Morton 1978, Satz 22)**

There are exactly $n$ values of $k$ for which $s(k) = n$.

- `Walnut` can create base-$b$ linear representations for values of synchronized sequences.
- By a base-$b$ linear representation for a function $f(n)$ we mean vectors $v, w$, and a matrix-valued morphism $\gamma$ such that $f(n) = v\gamma(x)w$ for all strings $x$ representing $n$ in base $b$.
- (The dimension of $v$ is called the rank of the representation.)

`eval satz22 n "$rss(?msd_4 k,n)":`

gives us a base-$2$ linear representation of rank $7$ computing the function $f(n)$ that counts the number of times the function $s$ takes the value $n$.

`eval gfunc n "i<n":`

gives us a base-$2$ linear representation for the function $g(n) = n$:

- To conclude, we compute a base-$2$ linear representation for $f(n) - g(n)$, and minimize it using an algorithm of Schützenberger (Section 2.3 of Berstel & Reutenauer (2011)).
- When we do so, we get the representation for the $0$ function, so $f(n) = n$.

The same method allows us to prove a new result:

## Theorem

(a) For $n \in [0, 4^m/2)$, $0$ appears as a value of $t(n)$ exactly $2^{m-1}$ times, and $k$ appears exactly $2^m - k$ times for $1 \leq k < 2^m$.

(b) For $n \in [0, 4^m)$, $0$ appears as a value of $t(n)$ exactly $2^m - 1$ times, $2^m$ appears exactly once, and $k$ appears exactly $2(2^m - k)$ times for $1 \leq k < 2^m$.

- We have shown how to obtain much simpler proofs of Brillhart and Morton's many results on the partial sums of the Rudin-Shapiro sequence, and have obtained some new results.
- We can apply the same techniques to other sequences.
- For example, if we define a sequence $a'(n)$ that is $+1$ or $-1$ accordingly as the number of $00$'s in the binary expansion of $n$ is even or odd, we can prove the analogue of the Brillhart–Morton results for this sequence as well.

# The End