Lectures on Automatic Sequences

Jeffrey Shallit

April 25, 2020

Contents

Preface

These are lecture notes for a graduate course (or advanced undergraduate course) on the theory of automatic sequences and their generalizations. This course was given by Jeffrey Shallit at the University of Waterloo in the winter term of 2020. Two lectures were given each week for 9 weeks, with each lecture taking 80 minutes.

The main prerequisite for this course is familiarity with undergraduate combinatorics, algebra, and number theory. A course in formal languages and automata theory would also be very useful.

The four really big results of the course are

- Christol's theorem, which relates automatic sequences to algebraic elements of a formal power series field over a finite field (Lecture 8);
- The connection between automatic sequences and first-order logic (Lectures 10 and 11);
- Cobham's big theorem, which characterizes when a sequence can be simultaneously k- and k'-automatic (Lecture 15);
- the characterization of automatic real numbers as either rational or transcendental (Lecture 17).

The main text for the course is the book [6]. These notes provide additional material, improved results, and improved proofs.

Chapter 1

Introduction to automatic sequences

For us a *sequence* is (usually) a map from the natural numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$ to Σ, where Σ is a finite alphabet. (Sometimes we index starting at 1 instead of 0; sometimes we talk about "two-sided" or "bi-infinite" sequences; these are maps from $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ to Σ.) A sequence is also called an *infinite word* or *infinite string*. We write a sequence as $\mathbf{a} = (a_n)_{n\geq 0}$ or $\mathbf{a} = (a(n))_{n\geq 0}$. We (nearly) always use the **bold** font for infinite words.

The very simplest kinds of sequences are the ultimately periodic sequences. A sequence $(a_n)_{n\geq 0}$ is *ultimately periodic* if there exist integers $p \geq 1$, $N \geq 0$ such that $a_i = a_{i+p}$ for all $n \geq N$. Here p is called the *period* of the sequence and N is called the *preperiod*. Confusingly, sometimes the term *period* refers to the word $x = a_N a_{N+1} \cdots a_{N+p-1}$, and the term *preperiod* refers to the word $x = a_0 \cdots a_{N-1}$. An ultimately periodic sequence with period x and preperiod w in this latter sense is sometimes written as wx^ω, which means the infinite word $wxxx\cdots$.

Given a set $S \subseteq \mathbb{N}$, its *characteristic sequence* $\chi_S(n)$ is defined to be 1 if $n \in S$ and 0 otherwise.

Example 1. The characteristic sequence of the prime numbers $\mathbb{P} = \{2, 3, 5, \ldots\}$ is $\mathbf{p} = (p_n)_{n\geq 0}$:

n	0	1	2	3	4	5	6	7	8	9	10	11
p_n	0	0	1	1	0	1	0	1	0	0	0	1

Example 2. Let $s_k(n)$ be the sum of the digits of n when expressed in base k. The *Thue-Morse sequence* $\mathbf{t} = (t_n)_{n\geq 0}$ is defined as follows: $t_n = s_2(n) \bmod 2$. The first few terms are as follows:

n	0	1	2	3	4	5	6	7	8	9	10	11
$s_2(n)$	0	1	1	2	1	2	2	3	1	2	2	3
t_n	0	1	1	0	1	0	0	1	1	0	0	1

As we will see, these two sequences, $\mathbf{p}$ and $\mathbf{t}$, have some similarities and differences. Neither one, for example, is ultimately periodic. But $(t_n)_{n\geq 0}$ can be computed very quickly

using a recursion,

$$\begin{aligned} t_0 &= 0 \\ t_{2n} &= t_n \\ t_{2n+1} &= 1 - t_n \end{aligned} \tag{1.1}$$

for $n \geq 0$, while computing $\mathbf{p}$ efficiently needs a more complicated approach.

The sequence $\mathbf{t}$ is a prototypical example of an *automatic sequence*. *Very informally,* a sequence is k-automatic (for $k \geq 2$) if

(a) it has finite range; and

(b) it is completely defined by a system of equations analogous to those in (1.1), where the left- and right-hand sides involve only subsequences with indices of the form $k^i n + j$, $0 \leq j < k^i$.

The class of k-automatic sequences has many, many interesting properties that we will explore in this course. Some automatic sequences are "pseudo-random", lying in between the very simplest sequences (the ultimately periodic sequences) and the most complicated ones (random sequences).

Example 3. Let $e_{k;w}(n)$ be the number of (possibly overlapping) occurrences of the block w in the base-k expansion of n. The *Rudin-Shapiro sequence* $\mathbf{r} = (r_n)_{n\geq 0}$ is defined to be $e_{2;11}(n) \bmod 2$. (Warning: sometimes a variant of this sequence is also called the Rudin-Shapiro sequence; this is the sequence $(r'_n)_{n\geq 0}$ defined by $r'_n = (-1)^{r_n}$.) The first few terms are as follows:

n	0	1	2	3	4	5	6	7	8	9	10	11
$e_{2;11}(n)$	0	0	0	1	0	0	1	2	0	0	0	1
r_n	0	0	0	1	0	0	1	0	0	0	0	1

Then $\mathbf{r}$ is 2-automatic because we have

$$\begin{aligned} r_0 &= 0 \\ r_{2n} &= r_n \\ r_{4n+1} &= r_n \\ r_{4n+3} &= 1 - r_{2n+1} \end{aligned}$$

Now $\mathbf{r}$ has a truly *amazing* pseudorandomness property; namely, that for all integers $c \geq 1$ we have

$$\sum_{0\leq n<N} [r_n = r_{n+c}] = \frac{N}{2} + o(N),$$

where the implied error term in the $o(N)$ can depend on c. Here $[x = y]$ is the so-called *Iverson bracket*, defined to be 1 if $x = y$ and 0 otherwise. In other words, there is basically

no correlation between **r** and each of its shifts. In this respect, **r** behaves just like a truly random sequence! See [31].

In Figure 1.1 below we depict the difference $(\sum_{0\leq n<N}[r_n = r_{n+1}])-\frac{N}{2}$ for $N = 0, 1, \ldots, 2^{14}-1$.

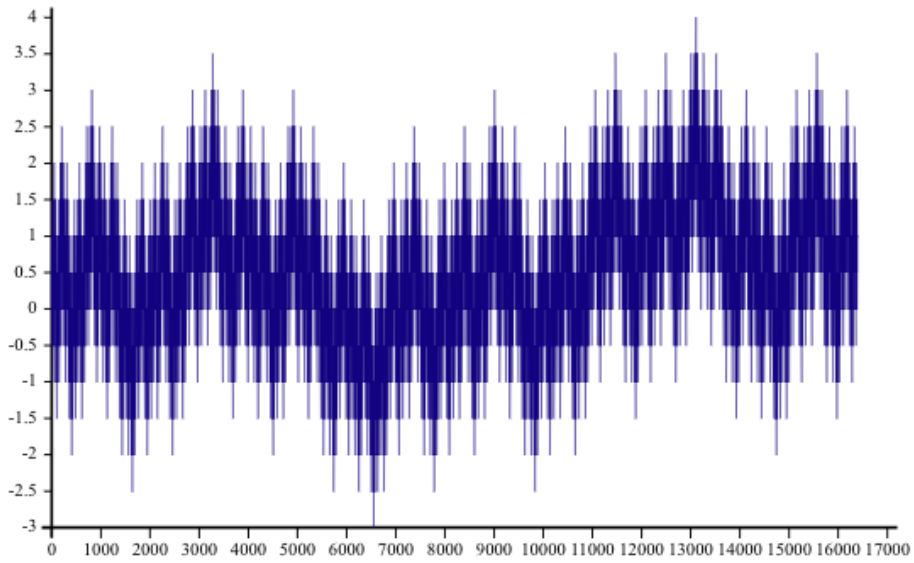


Figure 1.1: Pseudorandomness of the Rudin-Shapiro sequence

In this course we will study the computational, logical, algebraic, combinatorial, and number-theoretic properties of automatic sequences and their generalizations. Some generalizations include the *morphic sequences*, the *Sturmian sequences*, and the *k-regular sequences*.

We now give some examples of these properties.

1.1 Computational properties

Automatic sequences derive their name from another way to view such sequences: as computed by a *deterministic finite automaton with output* (also called DFAO). For example, here is a 2-DFAO computing the Rudin-Shapiro sequence:

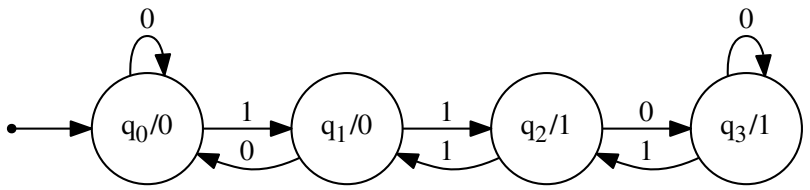


Figure 1.2: The Rudin-Shapiro DFAO

Here a state labeled q_i/a means that the state name is q_i and the output associated with the state is a. It is called a 2-DFAO because it expects its input to be the base-2 representation of some natural number n.

To use this machine, first express n in base 2. Feed the DFAO, starting with state q_0, with this representation. Follow the appropriate arrows for each digit. The output is the output associated with the last state reached. For example, if $n = 7$, then the base 2 representation is 111, and the sequence of states visited is q_0, q_1, q_2, q_1, and the output is 0.

The kinds of computational questions we are interested in include:

(a) Given a sequence represented in some way, is it k-automatic for some k?

(b) How many states are needed to compute a given automatic sequence?

(c) Can a sequence be simultaneously be k-automatic and k'-automatic for $k \neq k'$?

(d) What is the increase in the number of states when converting from one representation of a k-automatic sequence to another?

(e) Does it matter if numbers are read least-significant-digit first instead of most-significant-digit first? (This is sometimes called "lsd" and "msd".)

(f) If a sequence is not k-automatic, how can we prove this?

(g) What happens when we alter a k-automatic sequence in various ways? Does it remain k-automatic?

1.2 Logical properties

The kinds of logical questions we are interested in include:

(a) Is there a logical characterization of automatic sequences?

(b) What properties of automatic sequences are decidable?

(c) What logical theories involving automata are decidable/undecidable?

Example 4. For $n \geq 1$ define $V_k(n) = \max\{k^i : k^i \mid n\}$. Then the first-order logical theory $\mathrm{FO}(\mathbb{N}, +, V_2)$ is decidable, but $\mathrm{FO}(\mathbb{N}, +, V_2, V_3)$ is undecidable.

1.3 Algebraic properties

The kinds of algebraic questions we are interested in include the following:

(a) What are the relationships between automata, semigroups, and monoids?

(b) Let $(a_n)_{n\geq 0}$ be a sequence taking values in $\mathrm{GF}(q)$, the finite field with q elements. Under what conditions is the formal power series $\sum_{n\geq 0} a_n X^n$ algebraic?

Example 5. Recall that for $x, y \in \mathrm{GF}(q)$, for $q = p^n$, $n \geq 1$, and p prime, we have $(x + y)^p = x^p + y^p$. Suppose $q = 2$ and consider the formal power series T defined by $T(X) = \sum_{n\geq 0} t_n X^n$ where $(t_n)_{n\geq 0}$ is the Thue-Morse sequence. Then, using "decimation" on T we get

$$\begin{aligned} T(X) &= \sum_{n\geq 0} t_n X^n \\ &= \sum_{n\geq 0} t_{2n} X^{2n} + \sum_{n\geq 0} t_{2n+1} X^{2n+1} \\ &= \sum_{n\geq 0} t_n X^{2n} + X \sum_{n\geq 0} (1 - t_n) X^{2n} \\ &= \left(\sum_{n\geq 0} t_n X^n\right)^2 + X \sum_{n\geq 0} X^{2n} + X \left(\sum_{n\geq 0} t_n X^n\right)^2 \\ &= T(X)^2 + \frac{X}{1 - X^2} + X T(X)^2, \end{aligned}$$

which implies that

$$(1 + X)T(X)^2 + T(X) + \frac{X}{1 - X^2} = 0,$$

or

$$(1 + X)(1 + X^2)T(X)^2 + (1 + X^2)T(X) + X = 0.$$

So T the root of a quadratic equation with coefficients in $\mathrm{GF}(2)[X]$. Here we have used the fact that $1 = -1$ in $\mathrm{GF}(2)$. We will see that this is more generally true for all q-automatic sequences, where q is a prime power.

1.4 Combinatorial properties

The kinds of combinatorial questions we are interested in include:

(a) What kinds of repetitions can occur in a sequence? These might include *squares* (nonempty blocks of the form xx), *cubes* (nonempty blocks of the form xxx), etc.

(b) How many different blocks of size n occur in a sequence? This is sometimes called *subword complexity*.

(c) What is the frequency of occurrence of a given symbol (or block of symbols) in a sequence?

(d) What are the palindromes that occur in a sequence? The primitive words?

(e) What are the blocks shared in common between two sequences?

(f) Is a given sequence *recurrent*? That is, does every block that occurs, occur infinitely often?

1.5 Number-theoretic properties

The kinds of number-theoretic properties we are interested in include:

(a) Given a set of integers S, can every element of $\mathbb{N}$ be written as the sum of $\leq t$ elements of S? (additive number theory)

(b) How many different representations as the sum of $\leq t$ elements of S does n have?

(c) Let $b \geq 2$. Take an automatic sequence $(a_n)_{n\geq 0}$ and consider the real number $\sum_{n\geq 0} a_n b^{-n}$. Is it rational? Irrational? Algebraic? Transcendental? A Liouville number?

(d) Given an automatic set, does it contain a prime number? Infinitely many primes?

(e) Can one prove that π, $\log 2$, etc., do not have a base-b representation that is k-automatic?

(f) What is the asymptotic behavior of sums like $\sum_{0\leq n<N} a_n$, where $(a_n)_{n\geq 0}$ is a k-automatic sequence?

1.6 Notation

Here is the notation we use throughout the course:

- i, j, k, ℓ, m, n represent integers
- u, v, w, x, y, z represent finite words (strings)

- bold letters like $\mathbf{r}$, $\mathbf{t}$ represent infinite words (sequences)
- Σ_k is the alphabet $\{0, 1, \ldots, k-1\}$
- Σ^* is the set of all finite words over the alphabet Σ
- Σ^ω is the set of all 1-sided infinite words over Σ
- xy or $x \cdot y$ denote the concatenation of the words x and y
- x^i denotes the word $\overbrace{xx\cdots x}^{i}$.
- x^ω is the infinite word $xxx\cdots$
- $(n)_k$ is the canonical representation of n in base $k \geq 2$, using the digits in Σ_k only, and no leading zeroes. Note: the canonical representation of 0 is ϵ, the empty string
- $[w]_k$ is the number represented by w in base k, msd first. More precisely if $w = a_1 a_2 \cdots a_i$ then $[w]_k = \sum_{1 \leq j \leq i} a_j k^{i-j}$. Here there are no restrictions on the a_i; they can be $\geq k$ or ≤ 0. Notice that if $|a| = 1$ then $[wa]_k = k[w]_k + a$.
- $\nu_k(n)$ is the exponent of the highest power of k dividing n (for $n > 0$, $k \geq 2$)
- $V_k(n) = k^{\nu_k(n)}$
- w^R is the reversal of the word w
- $|x|$ is the length of the word x
- $|x|_a$ is the number of occurrences of the symbol a in x
- $x[i]$ is the i'th letter of x
- $x[i..j]$ is the subword beginning at position i and ending at position j of x

1.7 Notes

The main reference for the course is the book [6].

1.8 Exercises

1. Call a number *strange* if, in its base-2 representation, every maximal block of consecutive 1's is of odd length. Thus $n = 157$ is strange, because $(n)_2 = 10011101$, but $n = 158$ is not, because $(n)_2 = 10011110$. Find a 2-DFAO generating the characteristic sequence of the strange numbers.

2. Prove that the Thue-Morse sequence $\mathbf{t}$ is not ultimately periodic. (We will see a general way to solve problems like this in Lecture 11.)

3. Define $X_0 = 0$ and $X_{n+1} = X_n 0 \overline{X_n}$ for $n \geq 0$. Show that $\mathbf{X} = \lim_{n\to\infty} X_n$ is equal to $\overline{t_1 t_2 t_3 \cdots}$ where $\mathbf{t} = t_0 t_1 t_2 \cdots$ is the Thue-Morse sequence. Also find a 2-DFAO computing $\mathbf{X}$.

Chapter 2

Representation of integers

2.1 Representation of integers

Let $k \geq 2$. We say that a finite word w is a *canonical base-k representation* of some non-negative integer if

(a) $w[1] \neq 0$;

(b) $w \in \Sigma_k^*$.

Alternatively, the set of canonical representations can be written as $\Sigma_k^* - 0\Sigma_k^*$. The set of all canonical representations is written as C_k.

Theorem 6. *Every natural number n has a canonical base-k representation.*

Proof. We prove it by induction on n. The base cases are $n < k$. If $n = 0$, the canonical representation is ϵ, and if $1 \leq n < k$, the canonical representation is n. Now assume the result is true for $n' < n$; we prove it for n. Given $n \geq k$, define

$$a := n \bmod k;$$
$$n' := (n - a)/k.$$

It is easy to see that $0 \leq a < k$ and $1 \leq n' < n$. Hence n' has a canonical representation w'. Let $w = w'a$. Then

$$[w]_k = [w'a]_k = k[w'] + a = kn' + a = n,$$

and clearly $w \in C_k$. □

Theorem 7. *Canonical representations are unique.*

Proof. Suppose w, x are distinct canonical representations with $[w]_k = [x]_k$. Without loss of generality assume $|w| \leq |x|$.

Case 1: $m = |w| < |x| = n$. Then

$$[w]_k \leq (k-1)(1 + k + \cdots + k^{m-1}) = (k-1)\frac{k^m - 1}{k-1} = k^m - 1,$$

while

$$[x]_k \geq [1\overbrace{00\cdots0}^{n-1}]_k = k^{n-1},$$

which gives

$$[w]_k \leq k^m - 1 < k^m \leq k^{n-1} \leq [x]_k,$$

a contradiction.

Case 2: $|w| = |x| = n$. Without loss of generality assume n is as small as possible. If $w[1] = x[1]$ then write $w = aw'$, $x = ax'$ and observe that this implies that $w' \neq x'$ while $[w']_k = [x']_k$. So we have found an example with a smaller n, contradiction. So we must have $w[1] \neq x[1]$. Without loss of generality say $w[1] = a < b = x[1]$. Then

$$[w]_k \leq [a\overbrace{(k-1)\cdots(k-1)}^{n-1}]_k = a \cdot k^{n-1} + (k-1)\frac{k^{n-1} - 1}{k-1} < (a+1)k^{n-1},$$

while

$$[x]_k \geq [b\overbrace{0\cdots0}^{n-1}]_k = b \cdot k^{n-1},$$

a contradiction. □

2.2 Other kinds of base-k representations

2.2.1 Bijective representation

This kind of representation uses the digit set $\{1, 2, \ldots k\}$ instead of $\Sigma_k = \{0, 1, \ldots, k-1\}$. The lack of leading zeroes means that every natural number has exactly one representation, which means that it provides a natural bijection between $\{1, 2, \ldots, k\}^*$ and $\mathbb{N}$. As an example, here are the first few bijective base-2 representations:

n	representation
0	ϵ
1	1
2	2
3	11
4	12
5	21
6	22
7	111
8	112
9	121
10	122
11	211

2.2.2 The (k, ℓ)-numeration systems

These are a variation of base-$(k + \ell + 1)$ representation, using the digits

$$-k, 1-k, \ldots, -1, 0, 1, \ldots, \ell - 1, \ell.$$

The most famous example is $(1, 1)$-numeration, which is also called the *balanced ternary* system. Here every integer has a unique canonical representation, without leading zeroes:

n	$(1,1)$-representation
-7	$\overline{1}1\overline{1}$
-6	$\overline{1}10$
-5	$\overline{1}11$
-4	$\overline{11}$
-3	$\overline{1}0$
-2	$\overline{1}1$
-1	$\overline{1}$
0	ϵ
1	1
2	$1\overline{1}$
3	10
4	11
5	$1\overline{11}$
6	$1\overline{1}0$
7	$1\overline{1}1$

Here $\overline{1}$ is an abbreviation for the digit -1. One nice feature of balanced ternary is that one negates a number by simply changing the sign of each digit.

2.2.3 Redundant systems of numeration

As we will see in Lecture 15, sometimes it is useful to allow multiple representations for the same number. For example, we can take a finite set of digits S such that $\Sigma_k \subseteq S$, and consider base-k representation using the digits of S.

2.2.4 Base-$(-k)$ representation

Still another possibility is to keep the set of digits Σ_k, but change the base to $-k$. This system allows unique representation of all of $\mathbb{Z}$, not just $\mathbb{N}$. The table below illustrates representation in base-(-2):

n	base-(-2) representation
-6	1110
-5	1111
-4	1100
-3	1101
-2	10
-1	11
0	ϵ
1	1
2	110
3	111
4	100
5	101
6	11010
7	11011

2.2.5 Fibonacci representation

This system is based on the Fibonacci numbers, which are defined by $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$. The digits are $\Sigma_2 = \{0, 1\}$. If $w = a_1 a_2 \cdots a_t$, define $[w]_F = \sum_{1 \leq i \leq t} a_i F_{t+2-i}$.

There is a unique canonical representation for $\mathbb{N}$, which has no leading zeroes and obeys the rule that if $a_1 a_2 \cdots a_t$ is a canonical represenation, then $a_i a_{i+1} \neq 1$ for all i. In other words, a canonical representation does not have two consecutive 1's.

Here are the first few Fibonacci representations:

n	Fibonacci representation
0	ϵ
1	1
2	10
3	100
4	101
5	1000
6	1001
7	1010
8	10000

The canonical representation of n is written $(n)_F$.

More generally, we can consider the k-bonacci representation of integers, which use the generalized Fibonacci numbers defined as follows:

$$F_n^{(k)} = 0 \quad \text{for } 0 \le n \le k-2;$$
$$F_n^{(k)} = 1 \quad \text{for } n = k-1;$$
$$F_n^{(k)} = F_{n-1}^{(k)} + \cdots + F_{n-k}^{(k)} \quad \text{for } n \ge k.$$

The ordinary Fibonacci numbers are $F_n^{(2)}$. The numbers $F_n^{(3)}$ are sometimes called the "Tribonacci numbers" and are written T_n.

In this system, every integer $n \ge 0$ has a unique representation $n = \sum_{1 \le i \le t} a_i F_{k+t-i}^{(k)}$ where the $a_i \in \{0, 1\}$ and $a_i a_{i+1} \cdots a_{i+k-1} \ne 1$ for all i.

2.3 Greedy representations

Let $1 = u_0 < u_1 < u_2 \cdots$ be a strictly increasing sequence of integers. We can express $n = \sum_{0 \le i \le r} a_i u_i$ with $a_r \ne 0$, using the greedy algorithm:

Greedy(n)
$t := 0$;
while $(u_{t+1} \le n)$ do $t := t + 1$;
for $i := t$ downto 0 do
  $a_i := \lfloor n/u_i \rfloor$;
  $n := n - a_i u_i$;
return$(a_t \cdots a_0)$

Theorem 8. *Let $1 = u_0 < u_1 < \cdots$ be an increasing sequence of integers. Every non-negative integer n has exactly one representation of the form $n = \sum_{0 \le i \le s} a_i u_i$ where $a_s \ne 0$ and the a_i are non-negative integers satisfying the inequality*

$$a_0 u_0 + \cdots + a_i u_i < u_{i+1} \tag{2.1}$$

for all i.

Proof. By induction on n. It is clearly true for $n = 0$, for then the representation is ϵ. Otherwise, using the greedy algorithm above, write

$$\begin{aligned} n &= a_s u_s + r_s \quad (0 \le r_s < u_s) \\ r_s &= a_{s-1} u_{s-1} + r_{s-1} \quad (0 \le r_{s-1} < u_{s-1}) \\ &\vdots \\ r_2 &= a_1 u_1 + r_1 \quad (0 \le r_1 < u_1) \\ r_1 &= a_0 u_0 \end{aligned}$$

By induction $r_{i+1} = a_i u_i + \cdots + a_0 u_0$ for all i, and furthermore $n = r_{s+1}$. Also $r_{i+1} < u_{i+1}$, so the inequality (2.1) holds.

To see uniqueness, suppose n has two distinct representations

$$n = a_s u_s + \cdots + a_0 u_0 = b_s u_s + \cdots + b_0 u_0,$$

where we have padded the shorter representation, if needed, with leading 0's. At least one of a_s, b_s is nonzero. Let i be the largest index such that $a_{i+1} \neq b_{i+1}$. Without loss of generality, assume $a_{i+1} > b_{i+1}$. Then

$$\begin{aligned} u_{i+1} &\le (a_{i+1} - b_{i+1}) u_{i+1} \\ &= (b_i - a_i) u_i + \cdots + (b_0 - a_0) u_0 \\ &\le b_i u_i + \cdots + b_0 u_0, \end{aligned}$$

which contradicts (2.1). □

2.4 Using computational models to compute sequences

Roughly speaking, the sequences $(a_n)_{n \ge 0}$ we focus on in this course are determined as follows:

$$n \in \mathbb{N} \longrightarrow \text{representation of } n \longrightarrow \boxed{\text{computational device}} \longrightarrow a_n$$

Now we can formally define automata and k-automatic sequences. A deterministic finite automaton with output (DFAO) is a 6-tuple $M = (Q, \Sigma, \Delta, q_0, \delta, \tau)$, where

- Q is a finite nonempty set of states (typically written as $\{q_0, q_1, \ldots, q_{t-1}\}$);
- Σ is the finite input alphabet;
- Δ is the finite output alphabet;
- q_0 is the initial state;

- $\delta : Q \times \Sigma \to Q$ is the transition function, which is typically extended to $Q \times \Sigma^*$ in the obvious way;

- $\tau : Q \to \Delta$ is the output function.

On input the word w the output of M is said to be $\tau(\delta(q_0, w))$. If $\Sigma = \Sigma_k$, then we call M a k-DFAO.

When we use a DFAO to compute an automatic sequence, our convention is that the input is $(n)_k$, starting with the most significant digit. However, this can be relaxed or modified in a variety of different ways without substantially changing the definition. That is, the notion of k-automatic sequence is *robust:* small changes to the definition lead to the same class of sequences. Among the kinds of changes with this property are the following:

(a) We prove the input lsd-first;

(b) We allow inputs to have leading zeros;

(c) We use the bijective base-k representation instead of ordinary base-k representation.

Let us prove (a). Actually our proof is in more generality. Call $f : \Sigma^* \to \Delta$ a *finite-state function* if it is computed by some DFAO $M = (Q, \Sigma, \Delta, q_0, \delta, \tau)$.

Theorem 9. *Let f be a finite state function. Then the function f^R defined by $f^R(w) = f(w^R)$ is also a finite-state function.*

Proof. Let f be computed by $M = (Q, \Sigma, \Delta, q_0, \delta, \tau)$. Then we claim f^R is computed by $M^R = (S, \Sigma, \Delta, q_0', \delta', \tau')$ where

- $S = \Delta^Q$, the functions from Q to Δ;

- q_0' is the function sending q to $\tau(q)$;

- $\tau'(h) = h(q_0)$ for $h : Q \to \Delta$;

- $\delta'(g, a) = h$ where $h(q) := g(\delta(q, a))$.

It now suffices to prove $\delta'(q_0', w) = h$, where $h(q) = \tau(\delta(q, w^R))$. We can do this by induction on $|w|$.

The base case is $|w| = 0$, i.e., $w = \epsilon$. Then $\delta'(q_0', \epsilon) = q_0' =$ the map sending q to $\tau(q)$.

Now assume the claim is true for $|w| = n$. We prove it for $|w| = n + 1$. Write $w = xa$, $|x| = n$, and a a single letter. Then

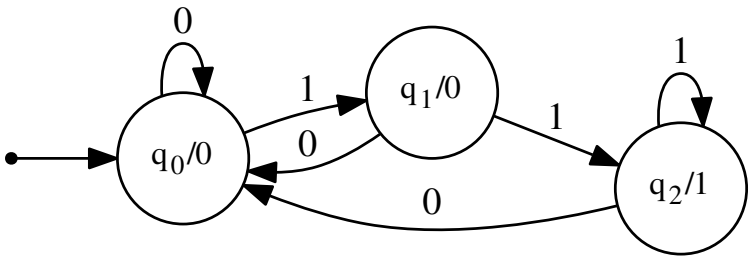
$$\begin{aligned} \delta'(q_0', xa) &= \delta'(\delta'(q_0', x), a) \\ &= \delta'(g, a) = h, \end{aligned}$$

where $g(q) = \tau(\delta(q, x^R))$ by induction. So

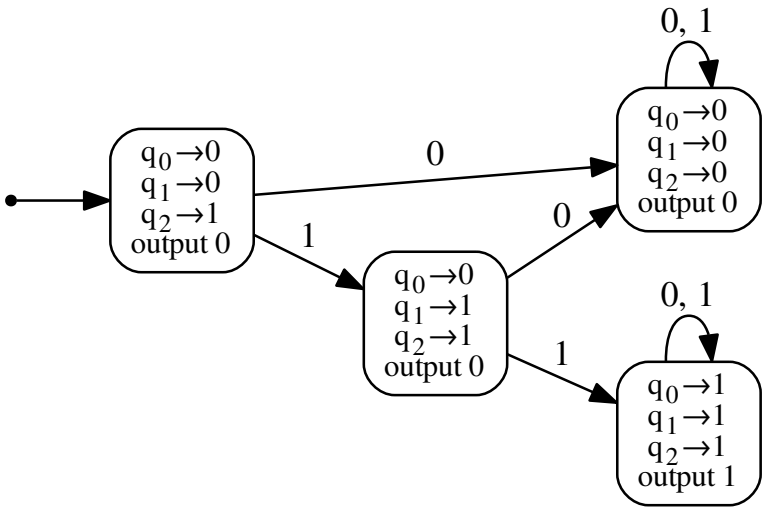
$$\begin{aligned} h(q) &= g(\delta(q, a)) \quad \text{(by definition of } \delta') \\ &= \tau(\delta(\delta(q, a), x^R)) \\ &= \tau(\delta(q, ax^R)) \\ &= \tau(\delta(q, (xa)^R)) \\ &= \tau(\delta(q, w^R)), \end{aligned}$$

as desired. □

Example 10. Consider the 2-DFAO below; it generates the characteristic sequence of those integers of the form $4i + 3$.



When we apply the construction of Theorem 9, we get the 2-DFAO below:



For (b), where inputs can have leading zeroes, we simply modify the automaton, introducing a new initial state q_0' that goes to itself on input 0, and in all other respects behaves like q_0.

From now on, we will assume that all DFAO's that work with the msd-first representation have an initial state that goes to itself on input 0.

The modification (c) is left as an exercise.

2.5 Notes

The textbook [6] gives some of the history of bijective base-k representation, but here are some additional references not cited there: [27], [46, pp. 34–36].

2.6 Exercises

1. (a) Let $[w]_F$ denote the value of the binary string w when interpreted in Fibonacci base, that is, if $w = a_1 a_2 \cdots a_t$, then $[w]_F = \sum_{1 \le i \le t} a_i F_{t+2-i}$, where $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for $n \ge 2$. Note that this interpretation means that w is written msd-first.

   Also note that here we do *not* assume that w is a canonical Fibonacci representation; in other words, w is allowed to have two consecutive 1's.

   Let w, x be binary strings. Prove that $[w]_F = [x]_F$ iff $[w0]_F = [x0]_F$.

(b) Show, by means of a counterexample, that the result is no longer true if w is over a larger alphabet such as $\{0, 1, 2\}$.

Chapter 3

Regular languages and morphisms

We now explore the relationship between regular languages, DFA's, and DFAO's.

Recall that a DFAO is a 6-tuple $(Q, \Sigma, \delta, q_0, \Delta, \tau)$, and the output on input w is $\tau(\delta(q_0, w))$.

A DFA is similar: it is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$ where $F \subseteq Q$ is the set of *final states* (also called *accepting states*).

We can think of a DFA as a DFAO by defining $\Delta = \{0, 1\}$ and

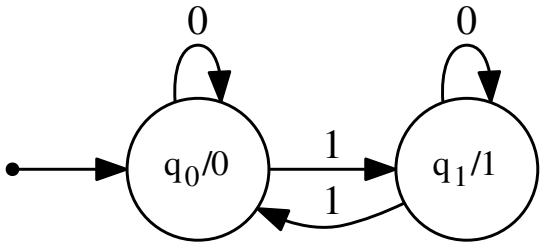
$$\tau(q) = \begin{cases} 1, & \text{if } q \in F; \\ 0, & \text{if } q \notin F. \end{cases}$$

If $\delta(q_0, w) \in F$, then we say M *accepts* w, and otherwise M *rejects* w. A *language* is a (finite or infinite) set of strings, a subset of Σ^*. The *language recognized* by a DFA M is defined to be

$$L(M) = \{x \in \Sigma^* \,:\, \delta(q_0, x) \in F\}.$$

A language is *regular* if $L = L(M)$ for some DFA M.

Example 11. A 2-DFAO for the Thue-Morse sequence **t** is depicted below:



The equivalent DFA is as follows:

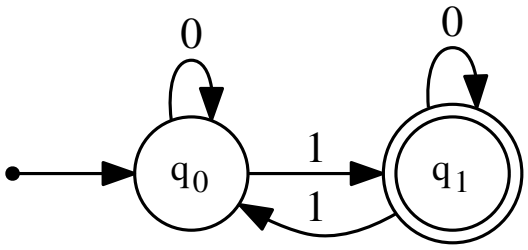


Figure 3.1: The Thue-Morse DFA

It recognizes the language $L = \{x \in \{0,1\}^* : |x|_1 \text{ is odd}\}$. Note that a double circle is used to denote the final states of a DFA.

3.1 Operations on regular languages

The principal operations on regular languages are union, concatenation, and Kleene star. Union is just the ordinary union of sets. The concatenation of languages is denoted $L_1 \cdot L_2$ or $L_1 L_2$ and is defined to be $\{xy : x \in L_1, y \in L_2\}$. We can raise a language to a natural number power by writing $L^n = \overbrace{LL\cdots L}^{n}$. Alternatively we can define L^n inductively by

$$L^n = \begin{cases} \{\epsilon\}, & \text{if } n = 0; \\ L \cdot L^{n-1}, & \text{if } n \geq 1. \end{cases}$$

Finally, we define Kleene star by $L^* = \bigcup_{i \geq 0} L^i$. Alternatively $L^* = \{x_1 x_2 \cdots x_i : i \geq 0 \text{ and each } x_i \in L\}$.

Theorem 12. *A language is regular iff it can be expressed by a finite combination of the operations of union, concatenation, and Kleene star, starting from ϵ and the elements of the alphabet Σ.*

Proof. (Sketch.) Make automata for ϵ and each element of Σ; join them using "ϵ-transitions" (which allow transitions from state to state without consuming any symbols of the input); remove the ϵ-transitions (which may result in a nondeterministic automaton or NFA); convert the NFA to a DFA using the "subset construction". For more details, see Section 4.1 of [6]. □

We use a notation for regular languages called *regular expressions*. In such an expression, let r_i be such an expression specifying L_i for $i = 1, 2$. Then

- $\emptyset$ denotes the empty set;
- ϵ denotes the language $\{\epsilon\}$;
- a denotes the language $\{a\}$;
- $(r_1)(r_2)$ denotes the language L_1L_2;
- $(r_1)^*$ denotes the language L_1^*;
- $r_1 \cup r_2$ denots the language $L_1 \cup L_2$.

Superfluous parens may be omitted. The precedence in regular expressions is as follows: star has highest precedence (is done first), then concatenation, then union.

Example 13. A regular expression for the language recognized by the DFA in Figure 11 is $0^*10^*(10^*10^*)^*$.

Six other useful operations on languages are complement, intersection, reversal, quotient, rlz (remove leading zeroes), and rtz (remove trailing zeroes). The reversal L^R of a language L is defined as follows: $L^R = \{x : x^R \in L\}$. The quotient of two language L_1/L_2 is defined to be

$$\{x \in \Sigma^* : \exists y \in L_2 \text{ such that } xy \in L_1\}.$$

For $x \in \Sigma_k^* - \Sigma_k^* 0$, define $\text{rtz}(x0^i) = x$ for all i. Similarly, for $x \in \Sigma_k^* - 0\Sigma_k^*$, define $\text{rtz}(0^i x) = x$ for all i.

Theorem 14.

(a) If L is regular, then so is $\overline{L}$.

(b) If L_1, L_2 are both regular, then so is $L_1 \cap L_2$.

(c) If L is regular, so is L^R.

(d) If L_1 is regular, then L_1/L_2 is regular for all languages L_2.

(e) If L is regular, then so is $\text{rtz}(L)$.

(f) If L is regular, then so is $\text{rlz}(L)$.

Proof.

(a) Take a DFA M for L, and change the "finality" of each state, making each final state non-final and vice versa. The resulting DFA recognizes $\overline{L}$.

(b) Take a DFA M_1 for L_1 and a DFA M_2 for L_2. Create a new DFA M for $L_1 \cap L_2$ by using the "direct product" construction, where states are ordered pairs. Then M simulates the computation of M_1 and M_2 simultaneously in parallel, and accepts iff final states are reached in both components.

(c) By induction on the number of operators in a regular expression for L. It suffices to show that $(L_1 L_2)^R = L_2^R L_1^R$ and $(L^*)^R = (L^R)^*$.

(d) Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA for L_1. Define $M' = (Q, \Sigma, \delta, q_0, F')$, where $F' = \{q \in Q : \exists y \in L_2 \text{ such that } \delta(q, y) \in F\}$.

(e) Check that $\text{rtz}(L) = (L/0^*) \cap (\Sigma_k^* - \Sigma_k^* 0)$.

(f) Check that $\text{rlz}(L) = (\text{rtz}(L^R))^R$.

□

We can now give an alternative characterization of automatic sequences. Let $\mathbf{a} = (a_n)_{n \geq 0}$ be a sequence taking values in Δ. For each $d \in \Delta$ define the *fiber* $I_d(\mathbf{a}) = \{(n)_k : a_n = d\}$.

Theorem 15. *The sequence* $\mathbf{a}$ *is* k*-automatic iff each of the languages* $I_d(\mathbf{a})$*, for* $d \in \Delta$*, is a regular language.*

Proof. $\Longrightarrow$: Let $M = (Q, \Sigma, \Delta, \delta, q_0, \tau)$ be a DFAO computing $\mathbf{a} = (a_n)_{n \geq 0}$. For each $d \in \Delta$, define $M_d = (Q, \Sigma, \delta, q_0, F_d)$, where $F_d = \{q \in Q : \tau(q) = d\}$. Then F_d recognizes the language $0^* I_d(\mathbf{a})$, so this is regular. Then $\text{rlz}(0^* I_d(\mathbf{a})) = I_d(\mathbf{a})$ is regular.

$\Longleftarrow$: Suppose each $I_d(\mathbf{a})$ is regular. Then $0^* I_d(\mathbf{a})$ is regular. So there exists a DFA $M_d = (Q_d, \Sigma, \delta_d, q_{0d}, F_d)$ recognizing $0^* I_d(\mathbf{a})$. Now combine each of these DFA's into a single DFAO M using the "direct product" construction, as follows: $M = (Q, \Sigma, \delta, q_0, \Delta, \tau)$ where

- $\Delta = \{d_1, d_2, \ldots, d_i\}$;
- $Q = Q_{d_1} \times \cdots \times Q_{d_i}$;
- $\delta([p_1, \ldots, p_i], a) = [\delta_{d_1}(p_1, a), \ldots, \delta_{d_i}(p_i, a)]$;
- $q_0 = [q_{01}, \ldots, q_{0i}]$;
- $\tau([p_1, p_2, \ldots, p_i]) = d_j$ if j is the (unique) index such that $p_j \in F_{d_j}$.

□

3.2 The pumping lemma

One of the classic results on regular languages, which can be found in every undergraduate textbook on automata theory, is the pumping lemma.

Lemma 16 (The "pumping lemma")**.** *Let L be regular. Then there exists a constant n, depending on L, such that for all $z \in L$ with $|z| \geq n$, there exists a decomposition $z = uvw$ with $|uv| \leq n$ and $|v| \geq 1$, such that $uv^i w \in L$ for all $i \geq 0$.*

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA recognizing L. Let $n = |Q|$. Then any path from q_0 to a final state of F of length $\geq n$ visits at least $n+1$ states, so by the pigeonhole principle some state is repeated in the list of the states visited on all prefixes of length $\leq n$. Thus there exist a state q and words u, v, w with $|uv| \leq n$ and $|v| \geq 1$ such that $z = uvw$ and $\delta(q_0, u) = q$, $\delta(q, v) = q$, and $\delta(q, w) \in F$. Hence $\delta(q_0, uv^i w) \in F$ for all $i \geq 0$. □

The idea behind this lemma is often useful to prove sequences not k-automatic.

Theorem 17. *Suppose $\mathbf{a} = (a(n))_{n\geq 0}$ is a k-automatic sequence. Then the subsequences $(a(k^n - 1))_{n\geq 0}$ and $(a(k^n))_{n\geq 0}$ are ultimately periodic.*

Proof. Let $M = (Q, \Sigma, \Delta, \delta, q_0, F)$ be a k-DFAO generating $\mathbf{a}$. We have $[k^n - 1]_k = \overbrace{(k-1)\cdots(k-1)}^{n}$. Then the sequence of states reached upon reading $k-1$ over and over must eventually repeat, and so the output of the k-DFAO on the subsequence $(a(k^n - 1))_{n\geq 0}$ is ultimately periodic. The same argument works for $(a(k^n))_{n\geq 0}$. □

Example 18. Consider sequence $\mathbf{u} = (u(n))_{n\geq 0}$ defined by $u(n) = t(\lfloor \log_2 n + 1 \rfloor)$, where $(t(n))_{n\geq 0}$ is the Thue-Morse sequence. Then $u(2^n - 1) = t(n)$, so $\mathbf{u}$ is not ultimately periodic and hence not 2-automatic.

3.3 Morphisms

Let Σ, Δ be alphabets. A *morphism* is a map from Σ^* to Δ^* that obeys the identity $h(xy) = h(x)h(y)$ for all $x, y \in \Sigma^*$. Note that $h(\epsilon)h(\epsilon) = h(\epsilon)$, and hence $h(\epsilon) = \epsilon$. Also the identity implies that it suffices to define a morphism on each element of Σ only.

Example 19. The Thue-Morse morphism μ is defined by $\mu(0) = 01$ and $\mu(1) = 10$.

A morphism is *k-uniform* if $|h(a)| = k$ for all $a \in \Sigma$. A 1-uniform morphism is called a *coding*.

If $\Delta \subseteq \Sigma$, then we can iterate h. We define

$$h^0(x) = x$$
$$h^1(x) = h(x)$$
$$h^2(x) = h(h(x)),$$

and in general $h^{i+1}(x) = h(h^i(x))$ for all $i \geq 0$.

Suppose $h(a) = ax$ for some $a \in \Sigma$ and $x \in \Sigma^*$, and furthermore suppose $h^i(x) \neq \epsilon$ for all $i \geq 0$. In this case we say h is *prolongable on* a, or just *prolongable*.

Theorem 20. *If h is prolongable on a, say $h(a) = ax$, then $\lim_{n\to\infty} h^n(a)$ exists and is equal to the infinite word*

$$h^\omega(a) := a\, x\, h(x)\, h^2(x) \cdots$$

which furthermore is a fixed point of h.

Proof. We prove by induction that $h^{n+1}(a) = a\, x\, h(x) \cdots h^n(x)$.The base case is $n = 0$, and then $h(a) = ax = ah^0(x)$.

Now assume the result is true for n; we prove it for $n + 1$. Then

$$\begin{aligned} h^{n+2}(a) &= h(h^{n+1}(a)) \\ &= h(a\, x\, h(x) \cdots h^n(x)) \\ &= h(a)\, h(x)\, h(h(x)) \cdots h(h^n(x)) \\ &= a\, x\, h(x)\, h^2(x) \cdots h^{n+1}(x), \end{aligned}$$

as desired. This is an infinite word because each $h^i(x)$ is nonempty by hypothesis.

Furthermore note that

$$\begin{aligned} h(h^\omega(a)) &= h(a\, x\, h(x)\, h^2(x) \cdots) \\ &= h(a)\, h(x)\, h^2(x)\, h^3(x) \cdots \\ &= a\, x\, h(x)\, h^2(x)\, h^3(x) \cdots \\ &= h^\omega(a), \end{aligned}$$

so $h^\omega(a)$ is a fixed point of h. □

A basic result about morphisms is the following.

Theorem 21. *Let $L \subseteq \Sigma^*$ be a regular language, and $h : \Sigma^* \to \Delta^*$ be a morphism. Then $h(L)$ is regular.*

Proof. Take a regular expression for L and replace every occurrence of the letter a by $(h(a))$. □

3.4 Cobham's little theorem

We are now ready for one of the fundamental (but easy!) results about automatic sequences, which relates fixed points of morphisms to automata.

Theorem 22. *Let $\mathbf{b} = (b_n)_{n\geq 0}$ be a sequence taking values in a finite alphabet Δ. Then $\mathbf{b}$ is k-automatic iff there exists a finite alphabet Γ, a k-uniform morphism $h : \Gamma^* \to \Gamma^*$ that is prolongable on some $a \in \Gamma$, and a coding $\tau : \Gamma^* \to \Delta^*$ such that $\mathbf{b} = \tau(h^\omega(a))$.*

Proof. $\Longrightarrow$: Suppose $\mathbf{b}$ is k-automatic. Then there exists a k-DFAO $M = (Q, \Sigma_k, \Delta, \delta, q_0, \tau)$ computing $\mathbf{b}$. Take $\Gamma = Q$. Define h as follows:

$$h(q) = \delta(q, 0)\delta(q, 1) \cdots \delta(q, k-1).$$

As we have seen in Lecture 2, we can assume without loss of generality that $\delta(q_0, 0) = q_0$. Take $a = q_0$.

Define $\mathbf{w} = h^\omega(a)$. First we will show, by induction on $|y|$, that

$$\delta(q_0, y) = \mathbf{w}[[y]_k]. \tag{3.1}$$

The base case is $|y| = 0$. Then the left-hand side of Eq. (3.1) is $\delta(q_0, \epsilon) = q_0 = a$, while the right-hand side is $\mathbf{w}[0] = a$.

Now assume that Eq. (3.1) is true for all y with $|y| < i$; we prove it for $|y| = i$. Write $y = xa$ for $a \in \Sigma_k$. Then

$$\begin{aligned}
\delta(q_0, y) &= \delta(q_0, xa) \\
&= \delta(\delta(q_0, x), a) \\
&= \delta(\mathbf{w}[[x]_k], a) \quad \text{(by induction)} \\
&= (h(\mathbf{w}[[x]_k]))[a] \quad \text{(by definition of } h\text{)} \\
&= (\mathbf{w}[k[x]_k..k[x]_k + k - 1])[a] \quad \text{(since } h(\mathbf{w}) = \mathbf{w} \text{ and } h \text{ is } k\text{-uniform)} \\
&= \mathbf{w}[k[x]_k + a] \\
&= \mathbf{w}[[xa]_k] \\
&= \mathbf{w}[[y]_k],
\end{aligned}$$

as desired.

It now follows that

$$\tau(\mathbf{w}[n]) = \tau(\mathbf{w}[(n)_k]) = \tau(\delta(q_0, (n)_k)) = b_n,$$

so $\tau(\mathbf{w}) = b_0 b_1 b_2 \cdots = \mathbf{b}$.

$\Longleftarrow$: Suppose $\mathbf{b} = \tau(\mathbf{w})$, where $\mathbf{w} = h^\omega(a)$ for some k-uniform morphism $h : \Gamma^* \to \Gamma^*$ prolongable on a. Define the k-DFAO $M = (\Gamma, \Sigma_k, \Delta, \delta, q_0, \tau)$, where $q_0 = a$ and

$$\delta(q, c) := (h(q))[c] \tag{3.2}$$

for all $q \in \Gamma$ and $c \in \Sigma_k$.

We now prove that

$$\mathbf{w}[n] = \delta(q_0, (n)_k) \tag{3.3}$$

for all $n \geq 0$, by induction on n. For $n = 0$ we have $\delta(q_0, (0)_k) = \delta(q_0, \epsilon) = q_0 = a = \mathbf{w}[0]$.

Now assume that Eq. (3.3) holds for $n' < n$; we prove it for n. Write $(n)_k = xa$, where $x \in \Sigma_k^*$ and $a \in \Sigma_k$. Then $n = kn' + a$ for $n' = [x]_k < n$.

$$\begin{aligned}
\delta(q_0, (n)_k) &= \delta(q_0, xa) \\
&= \delta(\delta(q_0, x), a) \\
&= \delta(\delta(q_0, (n')_k), a) \\
&= \delta(\mathbf{w}[n'], a) \quad \text{(by induction)} \\
&= (h(\mathbf{w}[n']))[a] \quad \text{(by Eq. (3.2))} \\
&= \mathbf{w}[kn' + a] \\
&= \mathbf{w}[n].
\end{aligned}$$

Hence $\tau(\delta(q_0, (n)_k)) = \tau(\mathbf{w}[n]) = \mathbf{b}[n] = b_n$. □

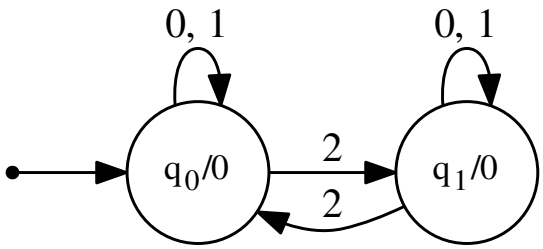
A consequence is that we can trivially convert from a k-DFAO computing $\mathbf{a}$ in the msd-first sense to the corresponding morphic representation $\tau(h^\omega(a))$, and vice versa, using the following correspondence:

- letters $\longleftrightarrow$ states;
- images of letters under morphism $h \longleftrightarrow$ mapping of states to other states on inputs $0, 1, \ldots, k-1$;
- image under $\tau \longleftrightarrow$ outputs of states.

Example 23. Refer to the Rudin-Shapiro DFAO from Lecture 1. We can convert this to its morphic representation as follows:

$$\begin{aligned}
h(q_0) &= q_0 q_1 & \tau(q_0) &= 0 \\
h(q_1) &= q_0 q_2 & \tau(q_1) &= 0 \\
h(q_2) &= q_3 q_1 & \tau(q_2) &= 1 \\
h(q_3) &= q_3 q_2 & \tau(q_3) &= 1.
\end{aligned}$$

Example 24. Consider the morphism defined by $g(0) = 001$ and $g(1) = 110$. Iterating g gives the infinite word $001001110001001110\cdots$ The corresponding 3-DFAO is then depicted below:



Remark 25. For each k-automatic sequence $\mathbf{b}$ there is a unique k-DFAO with the minimum number of states. (Uniqueness is up to renaming of the states.) Furthermore, if the states are numbered (say) $q_0, q_1, \ldots$, then there is a unique way to name the states, as follows: q_0 is the initial state, q_1 is the first state of the form $\delta(q_0, i)$ that is not equal to q_0, q_2 is either the first state of the form $\delta(q_0, j)$ unequal to q_0 or q_1 or (if there is no such state), the first state of the form $\delta(q_1, i)$ unequal to q_0, q_1, and so forth. This means that, given $\mathbf{b}$ and k, there is a canonical $\mathbf{w}$ and τ such that $\mathbf{b} = \tau(\mathbf{w})$; the sequence $\mathbf{w}$ is sometimes called the "interior sequence" of $\mathbf{w}$.

3.5 Notes

Cobham's little theorem is from [23], which is one of the most important papers in the area.

3.6 Exercises

1. Define a sequence of finite words as follows: $w(0) = \epsilon$, $w(1) = 1$, and

$$w(n) = w(n-1)\,2\,w(n-2)\,2\,w(n-1)$$

for $n \geq 1$. Thus, for example, $w(2) = 1221$ and $w(3) = 12212121221$.

Also define the morphism h by $h(1) = 122$ and $h(2) = 12$.

The goal is to show that the infinite words defined by $\lim_{n\to\infty} w(n)$ and $h^\omega(1)$ are the same. To do this, use induction to prove some identities linking $h^n(1)$ and $h^n(2)$ with $w(n)$ and/or $w(n+1)$. Then explain why these identities imply the desired result.

2. The *period-doubling* sequence $(d_n)_{n\geq 0}$ is defined by $d_n = |t_{n+1} - t_n|$, where $\mathbf{t} = t_0 t_1 t_2 \cdots$ is the Thue-Morse sequence.

Find a 2-automaton (in msd-format) generating the sequence $(d_n)_{n\geq 0}$.

3. The point of this exercise is to show that the fixed point of a morphism that is not uniform can also be the image of a fixed point of a morphism that is uniform, in a simple way.

   Consider the morphism h defined as follows:

$$\begin{aligned} h(2) &= 210 \\ h(1) &= 20 \\ h(0) &= 1, \end{aligned}$$

   and note that h is prolongable on 2. So $h^\omega(2)$ is well-defined, and is equal to the infinite word $210201\cdots$. Notice that h is not a uniform morphism.

   Now consider the 2-uniform morphism g and coding τ defined by

$$\begin{aligned} g(a) &= ab & \qquad \tau(a) &= 2 \\ g(b) &= ca & \qquad \tau(b) &= 1 \\ g(c) &= cd & \qquad \tau(c) &= 0 \\ g(d) &= ac & \qquad \tau(d) &= 1 \end{aligned}$$

   Note that g is prolongable on a, and so $\tau(g^\omega(a))$ is well-defined.

   Show that $\tau(g^\omega(a)) = h^\omega(2)$. Hint: using induction on n, prove some identities for $h^n(b)$ for $b = 0, 1, 2$ in terms of τ, powers of g, and some short words.

Chapter 4

The k-kernel

In Lecture 3 we saw a relationship between the msd-first k-DFAO and a representation as the image of a fixed point of a k-uniform morphism. This suggests trying to find a similar characterization for the lsd-first k-DFAO. Such a characterization involves something called the k-kernel. This characterization, where we focus on the least significant digits of a number as opposed to the most significant digits, is particularly useful for number-theoretic considerations about automatic sequences.

Given a sequence $\mathbf{u} = (u_n)_{n\geq 0}$ its k-kernel is defined to be the set of sequences

$$K_k(\mathbf{u}) = \{(u(k^i n + j))_{n\geq 0} \,:\, i \geq 0 \text{ and } 0 \leq j < k^i\}.$$

For example, for $k = 2$ this set is

$$\{(u(n))_{n\geq 0}, (u(2n))_{n\geq 0}, (u(2n{+}1))_{n\geq 0}, (u(4n))_{n\geq 0}, (u(4n{+}1))_{n\geq 0}, (u(4n{+}2))_{n\geq 0}, (u(4n{+}3))_{n\geq 0}, \ldots\}.$$

Notice that each sequence of the k-kernel arises from choosing a word of length t digits, and only indexing by those n having the specified trailing digits in their base-k representation.

Another way to think about this is that we take the original sequence and repeatedly apply the transformations

$$\begin{aligned} n &\to kn \\ n &\to kn + 1 \\ &\vdots \\ n &\to kn + k - 1 \end{aligned}$$

to the indices. This is a kind of repeated decimation of the sequence.

We are now ready to prove Eilenberg's theorem.

Theorem 26. *A sequence is k-automatic iff its k-kernel is of finite cardinality.*

Proof. $\Longrightarrow$: Suppose $(u_n)_{n\geq 0}$ is computed by the k-DFAO $M = (Q, \Sigma_k, \Delta, \delta, q_0, \tau)$ in msd-format. Without loss of generality, we can assume that $\delta(q_0, 0) = q_0$. Using Theorem 9, we

know that there is a DFA $M' = (Q', \Sigma_k, \Delta, \delta', q_0', \tau')$ such that $\tau(\delta(q_0, w)) = \tau'(\delta'(q_0', w^R))$. So M' computes $(u_n)_{n\geq 0}$ assuming the input is given lsd-first. That is, $u(n) = \tau'(\delta'(q_0', (n)^R 0^i))$ for all $i \geq 0$.

Let $(u(k^e n + j))_{n\geq 0}$ be an element of the k-kernel. Let $w \in \Sigma_k^e$ be such that $[w]_k = j$. Let $q = \delta'(q_0', w^R)$. We claim that $(u(k^e n + j))_{n\geq 0}$ is computed (in the lsd-first sense) by the k-DFAO where we change the initial state of M' to q, namely $(Q', \Sigma_k, \Delta, \delta', q, \tau')$. To see this note that if $n \neq 0$ then

$$\begin{aligned}
\delta'(q, (n)_k^R) &= \delta'(\delta'(q_0', w^R), (n)_k^R) \\
&= \delta'(q_0', w^R (n)_k^R) \\
&= \delta'(q_0', ((n)_k w)^R) \\
&= \delta'(q_0', (k^e n + j)_k^R),
\end{aligned}$$

and hence

$$\tau'(\delta'(q, (n)_k^R)) = \tau(\delta'(q_0', (k^e n + j)_k^R)) = u(k^e n + j).$$

On the other hand, if $n = 0$, then

$$\begin{aligned}
\delta'(q, (0)_k^R)) &= \delta'(q, \epsilon) \\
&= q \\
&= \delta'(q_0', w^R) \\
&= \delta'(q_0', (j)_k^R 0^i) \quad \text{for some } i \\
&= \delta'(q_0', (j)_k^R),
\end{aligned}$$

so

$$\tau'(\delta'(q, (0)_k^R)) = \tau(\delta'(q_0', (j)_k^R)) = u(j).$$

Thus we have identified each element of the k-kernel with some state of M'. It follows that the k-kernel is finite.

$\Longleftarrow$: Take the (finitely many) distinct elements of the k-kernel of $(u(n))_{n\geq 0}$, and make an automaton out of them. The elements of the k-kernel are the states, and transitions are defined as follows: If $p = (u(k^e n + j))_{n\geq 0}$, then $\delta(p, a) = q$, where $q = (u(k^e(kn + a) + j))_{n\geq 0}$. The initial state is $q_0 = (u(n))_{n\geq 0}$, and the output associated with $(u(k^e n + j))_{n\geq 0}$ is defined to be $u(j)$.

It is now a routine exercise to see that this is consistent (that is, if $(u(k^e n + j))_{n\geq 0}$ and $(u(k^f n + \ell))_{n\geq 0}$ are two elements of the k-kernel that coincide, then on each input these two states are mapped by δ in the same way). An easy induction now shows that on input w^R the automaton arrives at the state $(u(k^{|w|} n + [w]_k))_{n\geq 0}$ which has output $u([w]_k)$, as desired. □

Example 27. Let $(r(n))_{n\geq 0}$ be the Rudin-Shapiro sequence, as discussed in Lecture 1.

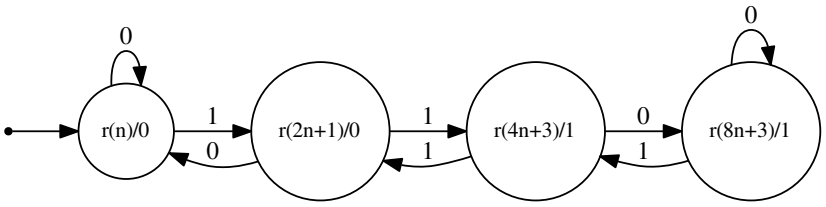
Check that it satisfies the recurrence relations

$$\begin{aligned} r(2n) &= r(n) \\ r(4n+1) &= r(n) \\ r(8n+7) &= r(2n+1) \\ r(16n+3) &= r(8n+3) \\ r(16n+11) &= r(4n+3), \end{aligned}$$

and hence its 2-kernel is

$$\{\, (r(n))_{n\geq 0},\ (r(2n+1))_{n\geq 0},\ (r(4n+3))_{n\geq 0},\ (r(8n+3))_{n\geq 0} \,\}.$$

We can now assemble the sequence of the 2-kernel into an automaton processing its input in lsd-first order:



4.1 Using the k-kernel to guess an automaton

Frequently we have a sequence known in some way (say, by a recursion, or as the fixed point of a non-uniform morphism), and we want to determine if it is k-automatic for some fixed k. Of course, in general, this is not a decidable problem.

Nevertheless, there are heuristic procedures that can guess the automaton if the sequence is indeed k-automatic. The easiest way is via the k-kernel. Start with the sequence $(u(n))_{n\geq 0}$, as many terms that are known. The main operation compares two sequences as follows: given prefixes of two sequences in the k-kernel (possibly of different sizes), we can definitively say the sequences are different if one is not a prefix of the other. However, if one is a prefix of the other, we assume (possibly wrongly) they are the same. If a prefix does not match any previously computed sequence, we split it ("decimate") by computing k new sequences via $n \to kn + a$ for $0 \leq a < k$. If a prefix is very short, it may match multiple previously-computed sequences; in this case, the data do not allow one to uniquely determine an automaton. Otherwise, one can form an automaton out of the elements of the k-kernel as above.

Example 28. Consider the *ordinary paperfolding sequence* $\mathbf{p} = (p_n)_{n\geq 1}$. (Notice that here we index starting at 1.) It is defined as the limit of the words P_n, $n \geq 1$, where

$$\begin{aligned} P_1 &= 0 \\ P_{n+1} &= P_n\, 0\, \overline{P_n}^R \end{aligned} \tag{4.1}$$

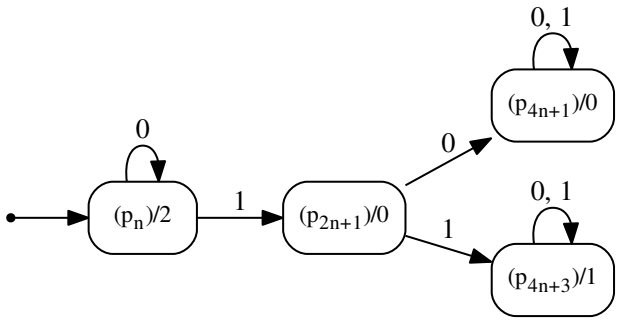
where the overline indicates the coding $0 \to 1$, $1 \to 0$. We find

$$\begin{aligned} P_2 &= 001 \\ P_3 &= 0010011 \\ P_4 &= 001001100011011 \\ &\vdots \end{aligned}$$

Add a new value $p_0 = 2$ so we can index starting at 0, and start decimating:

$$\begin{aligned} (p_n) &= 200100110001\cdots \\ (p_{2n}) &= 200100110001\cdots = (p_n) \\ (p_{2n+1}) &= 01010101\cdots \\ (p_{4n+1}) &= 000000\cdots \\ (p_{4n+3}) &= 111111\cdots \\ (p_{8n+1}) &= 000000\cdots = (p_{4n+1}) \\ (p_{8n+5}) &= 000000\cdots = (p_{4n+1}) \\ (p_{8n+3}) &= 111111\cdots = (p_{4n+3}) \\ (p_{8n+7}) &= 111111\cdots = (p_{4n+3}) \end{aligned}$$

which gives us this (conjectured) lsd-first automaton:



Of course, in general one cannot know for sure whether two elements of the k-kernel are truly equal, by examining only finitely many terms. This can lead to wrong conjectures.

Example 29. Start with the Thue-Morse sequence

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

amd define $\mathbf{s} = (s_n)_{n\geq 0}$ be the sequence of its run lengths (sizes of maximal blocks of consecutive identical elements), so

$$\mathbf{s} = 12112221121\cdots .$$

It can be shown that $\mathbf{s} = h^\omega(1)$, where $h(1) = 121$ and $h(2) = 12221$.

You might guess that $\mathbf{s}$ is 2-automatic—after all, it arises in a simple way from the 2-automatic sequence $\mathbf{t}$—but you would be wrong! Nevertheless, some elements of the 2-kernel can agree for many, many terms. For example,

$$s_{16n+1} = s_{64n+1}$$

agrees for the first million terms, which might easily lead you to suspect these two elements of the 2-kernel coincide. In fact, though, the equality holds for $0 \leq n \leq 1864134$, but fails at $n = 1864135$.

It is an open problem to understand exactly what is going on here, and find a tight bound on the number of terms two elements of the 2-kernel can agree on. An upper bound was given in [3].

4.2 Paperfolding sequences

The conjectured automaton for (p_n) obtained in the previous section immediately gives us the conjecture

$$p_n = \begin{cases} 0, & \text{if } j \equiv 0 \pmod 2; \\ 1, & \text{if } j \equiv 1 \pmod 2, \end{cases} \tag{4.2}$$

where $n = 2^i(2j+1)$ for integers $i, j \geq 0$.

We can now prove this conjecture by induction on n. It is true for $n = 1$ because then $i = 0$, $j = 0$, and $p_1 = 0$.

Now assume the claim is true for $n' < n$; we prove it for n. An easy induction shows that $|P_m| = 2^m - 1$. If $n = 2^i$ then p_n is the symbol immediately after P_n, which is 0. Since $j = 0$ for this n, the result holds.

Otherwise write $n = 2^i(2j+1)$ with $j > 0$. Write $2^t < n < 2^{t+1}$. Then by Eq. (4.1) we have $p_n = \overline{p_{2^{t+1}-n}}$. Now

$$\begin{aligned} 2^{t+1} - n &= 2^{t+1} - 2^i(2j+1) \\ &= 2^i(2^{t+1-i} - (2j+1)) \quad \text{(because } i < t\text{)} \\ &= 2^i(2r+1), \end{aligned}$$

where $r = 2^{t-i} - j - 1$. But r is the opposite parity of j, since $t - i > 0$. So

$$\begin{aligned} p_{2^{t+1}-n} &\equiv r \pmod 2 \\ &\equiv \overline{j} \pmod 2, \end{aligned}$$

which is what we needed to show.

The paperfolding sequence is so-named because it encodes the sequence of hills and valleys obtained by iterated folding of a piece of paper. (insert diagram here).

Eventually unfold all folds to 90°. We get a pattern of hills and valleys; writing 0 for a hill and 1 for a valley, we get $0010011\cdots$. To see that this is P_n, fold the paper once initially and then $n-1$ further times. At the last unfolding, we get $P_n = P_{n-1}0\overline{P_{n-1}}^R$.

More generally, you can fold at each step to insert either a hill (0) or a valley (1) at each step. This gives the *generalized paperfolding sequences:* starting fron a sequence of unfolding instructions $f_0, f_1, \ldots$, we define

$$\begin{aligned} P_0 &= \epsilon \\ P_{n+1} &= P_n\, f_n\, \overline{P_n}^R \quad \text{for } n \geq 0. \end{aligned}$$

Writing $P_\infty = \lim_{n\to\infty} = p_1 p_2 \cdots$, we have

$$p_n = \begin{cases} f_j, & \text{if } j \equiv 0 \pmod 2; \\ \overline{f_j}, & \text{if } j \equiv 1 \pmod 2, \end{cases}$$

where $n = 2^i(2j+1)$.

4.3 Another way to guess the automaton

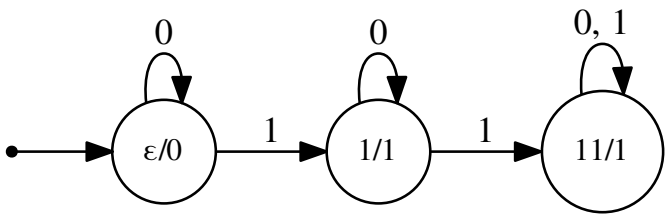
Above we saw how to guess the automaton for a k-regular sequence that was based on the 2-kernel; in other words, it was based on looking at the subsequences defined by fixing the last few bits of the base-k representation of n. This technique can (potentially) produce the automaton in lsd-first format.

There's a different way to handle this problem, which is based on fixing the *first* few bits instead. Here is one way to describe it. Given a sequence $\mathbf{a} = (a(n))_{n\geq 0}$, create a two-dimensional infinite array where the rows are labeled by the elements of C_k in radix order, and the columns are labeled by the elements of Σ_k^*. In row x and column y, put $a([xy]_k)$. It is now easy to show that this infinite array has only finitely many distinct rows iff $\mathbf{a}$ is k-automatic. Of course, we cannot compute the whole array, but we can assume two rows are identical iff they agree on the number of terms that are known. Assign a state for each distinct row, and create a transition function that, on input $a \in \Sigma_k$, maps the row labeled x to the row labeled xa. The output of the row labeled x is the value of $a([x]_k)$ corresponding to $y = \epsilon$. This creates a msd-first automaton.

Example 30. Let us apply this technique to the characteristic sequence of the powers of 2. Suppose we know the value of this sequence for $0 \le n \le 15$. Then we can construct the following table:

$x\backslash y$	ϵ	0	1	00	01	10	11	000	001	010	011	100	101	110	111
ϵ	0	0	1	0	1	1	0	0	1	1	0	1	0	0	0
1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0
10	1	1	0	1	0	0	0								
11	0	0	0	1	0	0	0								
000	0	0	1												
001	1	1	0												
010	1	1	0												
011	0	0	0												
100	1	1	0												
101	0	0	0												
110	0	0	0												
111	0	0	0												

From this we would guess that there are three distinct rows, corresponding to $x = \epsilon$, $x = 1$, and $x = 11$. This gives us the 3-state msd-first 2-DFAO given below:



4.4 Notes

For more about paperfolding, see [24].

4.5 Exercises

1. Use the experimental approach based on the 2-kernel to deduce an automaton for the characteristic sequence $\mathbf{c} = (c_n)_{n\ge 0}$ of those non-negative integers n that are the sum of three non-negative integer squares. The first few terms of $\mathbf{c}$ are

$$1111110111111101111111011110110111111101\cdots.$$

Step 1: Compute the first 5,000 terms (or so) of the characteristic sequence. The easiest way to do this is to loop over all possibilities i, j such that $i^2 + j^2 \leq n$, and then check if $n - i^2 - j^2$ is a square.

Step 2: Compute the 2-kernel by bifurcating the original sequence into odd and even terms, and then doing the same thing on the resulting subsequence, comparing each sequence with previously computed sequences, and assuming two subsequences are identical if they agree on the terms computed.

Step 3: Assemble these sequences into a least-significant-digit-first automaton.

Step 4: Can you prove the automaton is correct?

2. Let $(p_i)_{i\geq 1}$ be the regular paperfolding sequence.

   (a) show that $p_{2i} = p_i$ and $p_{2i+1} = (-1)^i$ for all $i \geq 1$.

   (b) Prove that $\sum_{n\geq 1} \frac{p_n}{n} = \frac{\pi}{2}$. Hint: split the sum into odd and even indexed terms.

3. The *period-doubling* sequence $(d_n)_{n\geq 0}$ is defined by $d_n = |t_{n+1} - t_n|$, where $\mathbf{t} = t_0 t_1 t_2 \cdots$ is the Thue-Morse sequence.

   (a) Find a 2-automaton (in msd-format) generating the sequence $(d_n)_{n\geq 0}$.

   (b) Prove that the period-doubling sequence contains no fourth powers (no blocks of the form $xxxx$, where x is nonempty).

Chapter 5

Continued fractions

A (simple) continued fraction is an expression of the form

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \ddots}}} \tag{5.1}$$

which we abbreviate as $[a_0, a_1, a_2, a_3, \ldots]$. It may be finite or infinite.

Example 31.

$$\begin{aligned} \frac{157}{68} &= [2, 3, 4, 5] \\ e &= [2, 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, \ldots] \\ \frac{1+\sqrt{5}}{2} &= [1, 1, 1, 1, \ldots] \\ \pi &= [3, 7, 15, 1, 292, \ldots] \end{aligned}$$

The following lemma lets us compute the values of truncated continued fractions $[a_0, a_1, \ldots, a_n]$ in a left-to-right manner.

Lemma 32. *Define*

$$\begin{aligned} p_{-2} &= 0 & q_{-2} &= 1 \\ p_{-1} &= 1 & q_{-1} &= 0 \\ p_k &= a_k p_{k-1} + p_{k-2} & q_k &= a_k q_{k-1} + q_{k-2}, \end{aligned}$$

for $k \geq 0$. Then

$$\frac{p_n}{q_n} = [a_0, a_1, \ldots, a_n].$$

Proof. By induction on n. The base cases $n = 0, 1$ are easily checked. Otherwise we have

$$\begin{aligned}
[a_0, a_1, \ldots, a_{n-1}, a_n, a_{n+1}] &= [a_0, a_1, \ldots, a_{n-1}, a_n + 1/a_{n+1}] \\
&= \frac{(a_n + \frac{1}{a_{n+1}})p_{n-1} + p_{n-2}}{(a_n + \frac{1}{a_{n+1}})q_{n-1} + q_{n-2}} \quad \text{(by induction)} \\
&= \frac{(a_{n+1}a_n + 1)p_{n-1} + a_{n+1}p_{n-2}}{(a_{n+1}a_n + 1)q_{n-1} + a_{n+1}q_{n-2}} \\
&= \frac{a_{n+1}(a_n p_{n-1} + p_{n-2}) + p_{n-1}}{a_{n+1}(a_n q_{n-1} + q_{n-2}) + q_{n-1}} \\
&= \frac{a_{n+1}p_n + p_{n-1}}{a_{n+1}q_n + q_{n-1}}.
\end{aligned}$$

□

We can now reinterpret this in terms of products of 2×2 matrices:

Corollary 33. *We have*

$$\begin{bmatrix} a_0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_1 & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} a_n & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} p_n & p_{n-1} \\ q_n & q_{n-1} \end{bmatrix}, \tag{5.2}$$

This is a really useful and fundamental result. From it we can obtain many other standard results in continued fractions:

Corollary 34.

(a) $(-1)^{n+1} = p_n q_{n-1} - p_{n-1} q_n$.

(b)

$$\begin{bmatrix} a_n & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_{n-1} & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} a_0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} p_n & q_n \\ p_{n-1} & q_{n-1} \end{bmatrix}, \tag{5.3}$$

(c) $[a_n, a_{n-1}, \ldots, a_1, a_0] = \frac{p_n}{p_{n-1}}$.

(d) $[a_n, a_{n-1}, \ldots, a_1] = \frac{q_n}{q_{n-1}}$.

(e)

$$\begin{bmatrix} a_0 & 1 \\ 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} a_1 & 1 \\ 1 & 0 \end{bmatrix}^{-1} \cdots \begin{bmatrix} a_n & 1 \\ 1 & 0 \end{bmatrix}^{-1} = (-1)^{n+1} \begin{bmatrix} q_{n-1} & -q_n \\ -p_{n-1} & p_n \end{bmatrix}.$$

(f)

$$\begin{bmatrix} -a_0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -a_1 & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} -a_n & 1 \\ 1 & 0 \end{bmatrix} = (-1)^{n+1} \begin{bmatrix} p_n & -p_{n-1} \\ -q_n & q_{n-1} \end{bmatrix}. \tag{5.4}$$

Proof.

(a) Follows by taking the determinant of both sides of Eq. (5.2).

(b) Follows by taking the transpose of both sides of Eq. (5.2).

(c) Follows immediately from (b).

(d) Follows immediately from (b).

(e) Note that

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = (ad-bc)^{-1} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix},$$

from which it follows (by taking the inverse of both sides of Eq. (5.3)) that

$$\begin{bmatrix} a_0 & 1 \\ 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} a_1 & 1 \\ 1 & 0 \end{bmatrix}^{-1} \cdots \begin{bmatrix} a_n & 1 \\ 1 & 0 \end{bmatrix}^{-1} = \left(\begin{bmatrix} a_n & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_{n-1} & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} a_0 & 1 \\ 1 & 0 \end{bmatrix} \right)^{-1} \tag{5.5}$$

$$= \begin{bmatrix} p_n & q_n \\ p_{n-1} & q_{n-1} \end{bmatrix}^{-1}$$

$$= (-1)^{n+1} \begin{bmatrix} q_{n-1} & -q_n \\ -p_{n-1} & p_n \end{bmatrix}.$$

(f) Observe that

$$A \begin{bmatrix} -c & 1 \\ 1 & 0 \end{bmatrix} A = \begin{bmatrix} c & 1 \\ 1 & 0 \end{bmatrix}^{-1} \tag{5.6}$$

where $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ is the antidiagonal matrix with determinant -1, satisfying $A^2 = I$.

Thus we have

$$\begin{bmatrix} -a_0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -a_1 & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} -a_n & 1 \\ 1 & 0 \end{bmatrix} = A^2 \begin{bmatrix} -a_0 & 1 \\ 1 & 0 \end{bmatrix} A^2 \begin{bmatrix} -a_1 & 1 \\ 1 & 0 \end{bmatrix} A^2 \cdots A^2 \begin{bmatrix} -a_n & 1 \\ 1 & 0 \end{bmatrix} A^2$$

$$= A \left(A \begin{bmatrix} -a_0 & 1 \\ 1 & 0 \end{bmatrix} A \right) \left(A \begin{bmatrix} -a_1 & 1 \\ 1 & 0 \end{bmatrix} A \right) \cdots \left(A \begin{bmatrix} -a_n & 1 \\ 1 & 0 \end{bmatrix} A \right) A$$

$$= A \begin{bmatrix} a_0 & 1 \\ 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} a_1 & 1 \\ 1 & 0 \end{bmatrix}^{-1} \cdots \begin{bmatrix} a_n & 1 \\ 1 & 0 \end{bmatrix}^{-1} A$$

$$= A(-1)^{n+1} \begin{bmatrix} q_{n-1} & -q_n \\ -p_{n-1} & p_n \end{bmatrix} A$$

$$= (-1)^{n+1} \begin{bmatrix} p_n & -p_{n-1} \\ -q_n & q_{n-1} \end{bmatrix},$$

as desired.

□

5.1 The continued fraction algorithm

Given a real number x, we can expand it as a continued fraction as follows:

CFE(x) { returns continued fraction of x}
while ($x \neq \infty$) do
  output($\lfloor x \rfloor$)
  $x := 1/(x - \lfloor x \rfloor)$

For example, on input $x = \pi$ we successively get an output of $\lfloor \pi \rfloor = 3$. Then x is set to $x_1 = 1/(\pi - 3) \doteq 7.06251\cdots$ and the output is 7. Then x is set to $1/(x_1 - 7) \doteq 15.99659\cdots$ and the output is 15, and so forth. This gives

$$\pi = [3, 7, 15, 1, 292, \ldots].$$

There are two entirely different meanings for the symbol $[a_0, a_1, \ldots, a_n]$ in the number theory literature. The first (and the one we typically) meaning is a rational function in the $n + 1$ variables $a_0, a_1, \ldots, a_n$. In this interpretation, the a_i could lie in any field at all and do not need to be integers. For example, if X is an indeterminate, then $[X, -X, X] = (X^3 - 2X)/(X^2 - 1)$.

The other meaning of $x = [a_0, a_1, \ldots]$ is that we apply the algorithm CFE to x and the algorithm outputs $a_0, a_1, \ldots$. In this case, typically (but not always) we assume x is a real number.

In the number theory literature these two meanings are often conflated, so be careful.

We now consider another domain where continued fractions have meaning: formal power series. Let X be an indeterminate. A *formal Laurent series* in X^{-1} over a field is an expression of the form $A(X) = \sum_{i \geq -c} a_i X^{-i}$ for some integer $c < \infty$. (In other words, there are only finitely many terms with positive exponents.) The formal Laurent series form a field, with the usual operations of addition and multiplication of series. This field has a theory of continued fractions (first expounded in Artin's thesis) exactly analogous to the theory for real numbers, even using the same algorithm. The only difference is that we need to define the notion of "floor" of a power series. We do that as follows:

$$\lfloor A(X) \rfloor = \sum_{-c \leq i \leq 0} a_i X^{-i}.$$

For example, let $A(X) = \sum_{i \geq 0} X^{-2^i} = X^{-1} + X^{-2} + X^{-4} + X^{-8} + \cdots$. Then we find

$$\begin{aligned}
\lfloor A \rfloor &= 0 \\
A_1 &:= \frac{1}{A} = X - 1 + X^{-1} - 2X^{-2} + 3X^{-3} - 4X^{-4} + 6X^{-5} - \cdots \\
A_2 &= \frac{1}{A_1 - (X - 1)} = X + 2 + X^{-1} - X^{-3} + 2X^{-5} + 2X^{-6} - \cdots \\
A_3 &= \frac{1}{A_2 - (X + 2)} = X + X^{-1} - X^{-3} - 2X^{-4} - 2X^{-5} + \cdots
\end{aligned}$$

and so $A = [0, X-1, X+2, X, \ldots] = [0, X-1, \mathbf{W}]$, where $\mathbf{W}$ is an infinite word constructed by "perturbed symmetry", i.e.,

$$W_0 = (X+2), X$$
$$W_{n+1} = W_n, X, (X-2), W_n^R$$

and $\mathbf{W} = \lim_{n\to\infty} W_n$. We don't prove this here; instead we prove something slightly easier to prove.

5.2 Paperfolding continued fractions

Lemma 35 (The "Folding Lemma"). *Suppose*

$$\frac{p_n}{q_n} = [c_0, c_1, \ldots, c_n]$$

and let w be the word $c_1, c_2, \ldots, c_n$. Then

$$[c_0, w, t, -w^R] = \frac{p_n}{q_n} + \frac{(-1)^n}{tq_n^2}.$$

Remark 36. The claim holds both if the c_i are integers, or polynomials in an indeterminate X.

Proof. By Eq. (5.2) we have

$$\begin{bmatrix} c_0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} c_1 & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} c_n & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} p_n & p_{n-1} \\ q_n & q_{n-1} \end{bmatrix}. \tag{5.7}$$

By Eq. (5.4) we have

$$\begin{bmatrix} -c_0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -c_1 & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} -c_n & 1 \\ 1 & 0 \end{bmatrix} = (-1)^{n+1} \begin{bmatrix} p_n & -p_{n-1} \\ -q_n & q_{n-1} \end{bmatrix}. \tag{5.8}$$

Take the transpose of both sides of the previous equation to get

$$\begin{bmatrix} -c_n & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -c_{n-1} & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} -c_0 & 1 \\ 1 & 0 \end{bmatrix} = (-1)^{n+1} \begin{bmatrix} p_n & -q_n \\ -p_{n-1} & q_{n-1} \end{bmatrix}. \tag{5.9}$$

Now multiply the previous equation on the right by

$$\begin{bmatrix} -c_0 & 1 \\ 1 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & c_0 \end{bmatrix}$$

to get

$$\begin{bmatrix} -c_n & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -c_{n-1} & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} -c_1 & 1 \\ 1 & 0 \end{bmatrix} = (-1)^n \begin{bmatrix} q_n & * \\ -q_{n-1} & * \end{bmatrix}, \tag{5.10}$$

where the asterisks denote entries that we don't need to know. Putting together Eqs. (5.7) and (5.10), we get

$$\begin{aligned} &\begin{bmatrix} c_0 & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} c_n & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} t & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -c_n & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} -c_1 & 1 \\ 1 & 0 \end{bmatrix} \\ &\quad = \begin{bmatrix} p_n & p_{n-1} \\ q_n & q_{n-1} \end{bmatrix} \begin{bmatrix} t & 1 \\ 1 & 0 \end{bmatrix} (-1)^n \begin{bmatrix} q_n & * \\ -q_{n-1} & * \end{bmatrix} \\ &\quad = \begin{bmatrix} (tp_n + p_{n-1})q_n - p_n q_{n-1} & * \\ (tq_n + q_{n-1})q_n - q_n q_{n-1} & * \end{bmatrix} (-1)^n \\ &\quad = \begin{bmatrix} tp_n q_n + (-1)^n & * \\ tq_n^2 & * \end{bmatrix} (-1)^n. \end{aligned}$$

It now follows from the correspondence between 2×2 matrices and continued fractions that

$$\begin{aligned} [c_0, w, t, -w^R] &= \frac{tp_n q_n + (-1)^n}{tq_n^2} \\ &= \frac{p_n}{q_n} + \frac{(-1)^n}{tq_n^2}, \end{aligned}$$

as desired. □

We now apply Lemma 35 to determine the continued fraction for a set of formal power series. First, recall the folding map $F_a : \Sigma^* \to \Sigma^*$, where $\Sigma = \{1, -1\}$. It is defined by $F_a(w) = w, a, -w^R$. We define

$$\mathrm{Fold}(a_n, a_{n-1}, \ldots, a_1) = F_{a_1}(F_{a_2}(\cdots F_{a_n}(\epsilon) \cdots)).$$

The ordinary paperfolding sequence is then given by $\mathrm{Fold}(1, 1, 1, \ldots)$, up to renaming of the symbols.

Theorem 37. *Let $e_0 = 1$ and $e_i = \pm 1$ for $i \geq 1$. Define $E(X) = X \sum_{i \geq 0} e_i X^{-2^i}$. Then $E(X) = [1, \mathrm{Fold}(e_1 X, -e_2 X, -e_3 X, \ldots)]$.*

Proof. Define the partial sum $E_m(X) = X \sum_{0 \leq i \leq m} e_i X^{-2^i}$. We prove by induction on m that the continued fraction for $E_m(X)$ is $[1, \mathrm{Fold}(e_1 X, -e_2 X, -e_3 X, \ldots, -e_m X)]$.

It is easy to verify that

$$\begin{aligned} E_0(X) &= X(X^{-1}) = 1 = [1] \\ E_1(X) &= X(X^{-1} + e_1 X^{-2}) = 1 + e_1 X^{-1} = [1, e_1 X]; \end{aligned}$$

this gives the base cases of the induction.

Now assume the claim is true for $m-1 \geq 1$; we prove it for m. Then

$$E_{m-1}(X) = [1, \text{Fold}(e_1X, -e_2X, -e_3X, \ldots, -e_{m-1}X)] = p_n/q_n.$$

By clearing the denominator, it is easy to see that $q_n = X^{2^{m-1}-1}$. Apply the folding lemma with $t = -e_mX$, letting $c_0 = 1$ and $w = \text{Fold}(e_1X, -e_2X, -e_3X, \ldots, -e_{m-1}X)$. Let $n = |w|$. An easy induction shows that $n = 2^{m-1} - 1$, which is odd because $m - 1 \geq 1$. The the definition of the folding map, together with the folding lemma, gives us

$$\begin{aligned}
[1, \text{Fold}(e_1X, -e_2X, -e_3X, \ldots, -e_mX)] &= [c_0, w, t, -w^R] \\
&= \frac{p_n}{q_n} + \frac{(-1)^n}{tq_n^2} \\
&= E_{m-1}(X) + \frac{1}{e_mXq_n^2} \\
&= E_{m-1}(X) + \frac{e_m}{X \cdot (X^{2^{m-1}-1})^2} \\
&= E_{m-1}(X) + \frac{e_m}{X^{2^{m-1}}} \\
&= E_m(X).
\end{aligned}$$

The desired result now follows by letting $m \to \infty$. □

Example 38. The simplest case is where $e_i = 1$ for all $i \geq 0$. We then get

$$\begin{aligned}
X \sum_{i \geq 0} X^{-2^i} &= [1, \text{Fold}(X, -X, -X, -X, \ldots)] \\
&= [1, X, -X, -X, -X, X, X, -X, -X, X, -X, -X, X, X, X, -X, \ldots].
\end{aligned}$$

We'd now like to "specialize" the continued fraction for the formal power series, by setting (for example) $X = 2$. This allows us to write the real number $E(2)$ as a continued fraction, but unfortunately this continued fraction has some negative partial quotients, which means it is not a legitimate simple continued fraction.

All is not lost, however, because there are some simple rules for removing negative and 0 terms for a continued fraction. Namely,

$$\begin{aligned}
[, \ldots, a, 0, b, \ldots] &= [, \ldots, a + b, \ldots] \\
[, \ldots, a, -b, c, \ldots] &= [, \ldots, a - 1, 1, b - 2, 1, c - 1, \ldots].
\end{aligned}$$

These rules can be verified by multiplying the corresponding 2×2 matrices out and observing that (up to ± 1 factors) they are the same.

Theorem 39. *The partial quotients of the simple continued fraction for $E(2)$ are all 1's and 2's.*

Proof. We claim that a general paperfolding continued fraction can be written as $[1, b_1, b_2, \ldots]$ where the b_i are blocks of partial quotients contained in the following finite set:

$$\{(2), (2,-2,2), (2,-2,-2,2), (2,-2,-2,-2,2), (2,-2,-2,2,-2,-2,2),$$
$$(2,-2,-2,2,-2,-2,-2,2), (2,-2,-2,-2,2,-2,-2,2)\}.$$

To see this, consider the consecutive occurrences of $(2,2)$ in a paperfolding sequence and look for the possible blocks of partial quotients that can appear between two such. (This can be done either by induction, or with `Walnut`: consider those n such that $(2,2)$ appears at position i, $(2,2)$ appears at position $i+n$, and there is no $(2,2)$ occuring in intermediate positions.)

Each of these blocks can then be rewritten as follows:

$$B_1 = (2) \to C_1 = (2)$$
$$B_2 = (2,-2,2) \to C_2 = (1,2,1)$$
$$B_3 = (2,-2,-2,2) \to C_3 = (1,1,1,1,1,1)$$
$$B_4 = (2,-2,-2,-2,2) \to C_4 = (1,1,1,2,1,1,1)$$
$$B_5 = (2,-2,-2,2,-2,-2,2) \to C_5 = (1,1,1,1,2,1,1,1,1)$$
$$B_6 = (2,-2,-2,2,-2,-2,-2,2) \to C_6 = (1,1,1,1,2,1,2,1,1,1)$$
$$B_7 = (2,-2,-2,-2,2,-2,-2,2) \to C_7 = (1,1,1,2,1,2,1,1,1,1)$$

To verify these, it suffices to compute the 2×2 matrix products associated with both sides. For example,

$$(2,-2,2) \sim \begin{bmatrix} -4 & -3 \\ -3 & -2 \end{bmatrix}$$

while

$$(1,2,1) \sim \begin{bmatrix} 4 & 3 \\ 3 & 2 \end{bmatrix}$$

Thus, for example, from the $2,-2,-2,-2,\ldots$ folding we get

$$[1, B_4, B_5, B_1, B_6, B_2, B_3, B_1, B_6, B_5, B_1, B_2, B_4, B_2, B_3, \ldots]$$

and this gets mapped to

$$[1, C_4, C_5, C_1, C_6, C_2, C_3, C_1, C_6, C_5, C_1, C_2, C_4, C_2, C_3, \ldots].$$

□

5.3 Notes

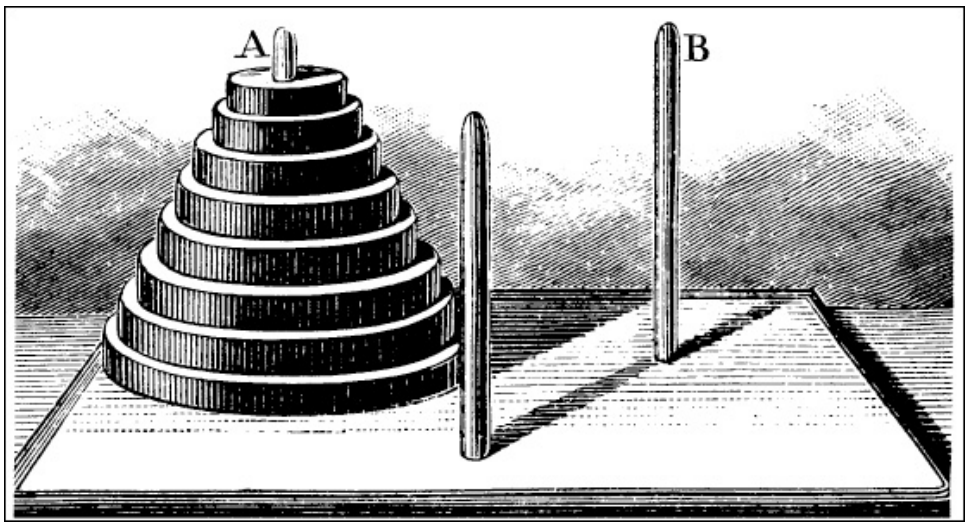
For more about these kinds of continued fractions, see [43, 44, 45, 48]. The proof of Theorem 39 was omitted from [48] and appears here for the first time.

Chapter 6

The Tower of Hanoi and closure properties of automatic sequences

6.1 The tower of Hanoi

The tower of Hanoi is a classic puzzle that you probably know. It consists of 3 pegs, labeled $1, 2, 3$, and N disks of strictly increasing sizes. Disks can be moved from peg to peg, but one can never put a larger disk on top of a smaller one. Initially all N disks are on peg 1, with the largest at the bottom and the smallest at the top, and we want to move all N of them to another peg.



Question 40. What is the most efficient way to do this (using the smallest number of moves)?

It is easy to see the answer is $2^N - 1$. For in order to move all N disks, the N'th must move at some point. Letting $t(N)$ denote the minimum number of moves needed, we see that in order to be able to move the N'th disk, at least one peg must be empty (otherwise

we'd be forced to put the N'th—the largest—on top of a smaller one). So all $N-1$ disks must be on some peg, and the N'th must be alone on its peg, and the remaining peg must be empty. Move the $N-1$ in the most efficient way possible. Then move the N'th. Finally, it remains to relocate the $N-1$ other disks, which we do in the most efficient way possible. This gives $t(N) = t(N-1) + 1 + t(N-1)$, and since $t(0) = 0$, the claim $t(N-1) = 2^N - 1$ follows by induction.

Question 41. In moving N disks, how can we efficiently determine what the n'th move is?

Our goal is to answer this question by providing a 2-DFAO that can compute the n'th move. We begin by encoding the moves of disks as follows:

$$\begin{aligned} \mathtt{a} &: 1 \to 2 & \qquad \overline{\mathtt{a}} &: 2 \to 1 \\ \mathtt{b} &: 2 \to 3 & \qquad \overline{\mathtt{b}} &: 3 \to 2 \\ \mathtt{c} &: 3 \to 1 & \qquad \overline{\mathtt{c}} &: 1 \to 3 \end{aligned}$$

So, in order to move 3 disks from peg 1 to peg 3 we write

$$\mathtt{a}\,\overline{\mathtt{c}}\,\mathtt{b}\,\mathtt{a}\,\mathtt{c}\,\overline{\mathtt{b}}\,\mathtt{a}.$$

Next, we introduce a coding that does a cyclic shift of the moves:

$$\begin{aligned} \sigma(\mathtt{a}) &= \mathtt{b} & \qquad \sigma(\overline{\mathtt{a}}) &= \overline{\mathtt{b}} \\ \sigma(\mathtt{b}) &= \mathtt{c} & \qquad \sigma(\overline{\mathtt{b}}) &= \overline{\mathtt{c}} \\ \sigma(\mathtt{c}) &= \mathtt{a} & \qquad \sigma(\overline{\mathtt{c}}) &= \overline{\mathtt{a}} \end{aligned}$$

We now define H_i to be the word encoding the optimal solution of the tower of Hanoi problem that moves i disks

from peg 1 to peg 2 if i is odd;

from peg 1 to peg 3 if i is even.

We do it this somewhat odd way so that H_i is a prefix of H_{i+1} for all i. Let $H = \lim_{n\to\infty} H_n$, the unique infinite word of which $H_0, H_1, H_2, \ldots$ are all prefixes. For example,

$$\begin{aligned} H_0 &= \epsilon \\ H_1 &= \mathtt{a} \\ H_2 &= \mathtt{a}\,\overline{\mathtt{c}}\,\mathtt{b} \\ H_3 &= \mathtt{a}\,\overline{\mathtt{c}}\,\mathtt{b}\,\mathtt{a}\,\mathtt{c}\,\overline{\mathtt{b}}\,\mathtt{a}. \end{aligned}$$

Lemma 42.

$$H_{2i+1} = H_{2i}\,\mathtt{a}\,\sigma^2(H_{2i}), \quad i \geq 0; \tag{6.1}$$

$$H_{2i} = H_{2i-1}\,\overline{\mathtt{c}}\,\sigma(H_{2i-1}), \quad i \geq 1. \tag{6.2}$$

Proof. Let H_{2i+1} be the optimal solution for $2i+1$ disks, moving disks from peg 1 to peg 2. First we move $2i$ disks from peg 1 to peg 3 via H_{2i}. Then we move disk number $2i+1$ from peg 1 to peg 2 via $\mathtt{a}$. Finally, we move $2i$ disks from peg 3 to peg 2 via $\sigma^2(H_{2i})$.

A similar argument works for H_{2i}. □

We now introduce a useful 2-uniform morphism φ on the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \overline{\mathtt{a}}, \overline{\mathtt{b}}, \overline{\mathtt{c}}\}$.

$$\begin{aligned} \varphi(\mathtt{a}) &= \mathtt{a}\,\overline{\mathtt{c}} & \varphi(\overline{\mathtt{a}}) &= \mathtt{a}\,\mathtt{c} \\ \varphi(\mathtt{b}) &= \mathtt{c}\,\overline{\mathtt{b}} & \varphi(\overline{\mathtt{b}}) &= \mathtt{c}\,\mathtt{b} \\ \varphi(\mathtt{c}) &= \mathtt{b}\,\overline{\mathtt{a}} & \varphi(\overline{\mathtt{c}}) &= \mathtt{b}\,\mathtt{a} \end{aligned}$$

This morphism φ interacts nicely with σ:

Lemma 43. *Let $w \in \Sigma^*$. Then*

$$\varphi(\sigma(w)) = \sigma^2(\varphi(w)) \tag{6.3}$$
$$\varphi(\sigma^2(w)) = \sigma(\varphi(w)). \tag{6.4}$$

Proof. It suffices to check this for each letter in Σ. □

Lemma 44. *For $i \geq 0$ we have*

$$H_{2i+1} = \varphi(H_{2i})\,\mathtt{a} \tag{6.5}$$
$$H_{2i+2} = \varphi(H_{2i+1})\,\mathtt{b}\ . \tag{6.6}$$

Proof. By induction on i. The case $i = 0$ is easy to check. Then

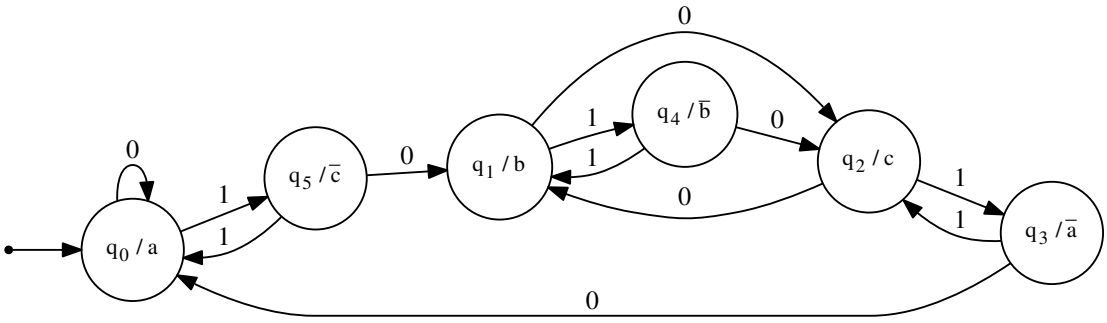
$$\begin{aligned} H_{2i+1} &= H_{2i}\,\mathtt{a}\,\sigma^2(H_{2i}) \quad \text{(by Eq. (6.1))} \\ &= \varphi(H_{2i-1})\,\mathtt{b}\,\mathtt{a}\,\sigma^2(\varphi(H_{2i-1})\,\mathtt{b}) \quad \text{(by induction and Eq. (6.6))} \\ &= \varphi(H_{2i-1})\,\varphi(\overline{\mathtt{c}})\,\varphi(\sigma(H_{2i-1}))\,\mathtt{a} \quad \text{(by Eq. 6.3)} \\ &= \varphi(H_{2i-1}\,\overline{\mathtt{c}}\,\sigma(H_{2i-1}))\,\mathtt{a} \\ &= \varphi(H_{2i})\,\mathtt{a} \quad \text{(by Eq. (6.2)).} \end{aligned}$$

The case of H_{2i+2} follows similarly and is omitted. □

Putting this all together, we get

Theorem 45. *We have $\mathbf{H} = \varphi(\mathbf{H})$, so $\mathbf{H}$ is 2-automatic.*

The infinite word $\mathbf{H}$ is generated by the automaton depicted below.



Using the techniques of Lecture 11, we can prove that **H** is squarefree; it does not contain two consecutive identical blocks.

6.2 Closure properties of automatic sequences

What operations can we do to a k-automatic sequence and have it remain k-automatic.

Let's start with extracting a linearly-indexed subsequence.

Theorem 46. *Suppose* $\mathbf{u} = (u(n))_{n\geq 0}$ *is* k*-automatic, and suppose* a, b *are non-negative integers. Then* $(u(an+b))_{n\geq 0}$ *is* k*-automatic.*

We actually give two different proofs of this fact, each using a different approach.

Proof #1, using the k*-kernel.* Suppose the k-kernel of $\mathbf{u}$ is

$$K_k(\mathbf{u}) = \{(u_1(n))_{n\geq 0}, \ldots, (u_r(n))_{n\geq 0}\}.$$

Define

$$S = \{(u_i(an+c))_{n\geq 0} : 1 \leq i \leq r,\ 0 \leq c < a+b\}.$$

Define $v(n) = u(an+b)$ and $\mathbf{v} = (v(n))_{n\geq 0}$. We claim that the k-kernel of $\mathbf{v}$ is a subset of S. Consider $(v(k^e \cdot n + j))_{n\geq 0}$ for $0 \leq j < k^e$, $e \geq 0$. By dividing by k^e, we can find d, f such that

$$ja + b = d \cdot k^e + f$$

for $0 \leq f < k^e$, $0 \leq d < a+b$. Then

$$\begin{aligned} v(k^e \cdot n + j) &= u(a(k^e \cdot n + j) + b) \\ &= u(k^e(an+d) + f). \end{aligned}$$

Now $(u(k^e \cdot m + f))_{m\geq 0}$ is an element of $K_k(\mathbf{u})$, say $(u_i(m))_{m\geq 0}$ for some i. So, substituting $m = an + d$, we get

$$v(k^e \cdot n + j) = u(k^e(an+d) + f) = u_i(an+d)$$

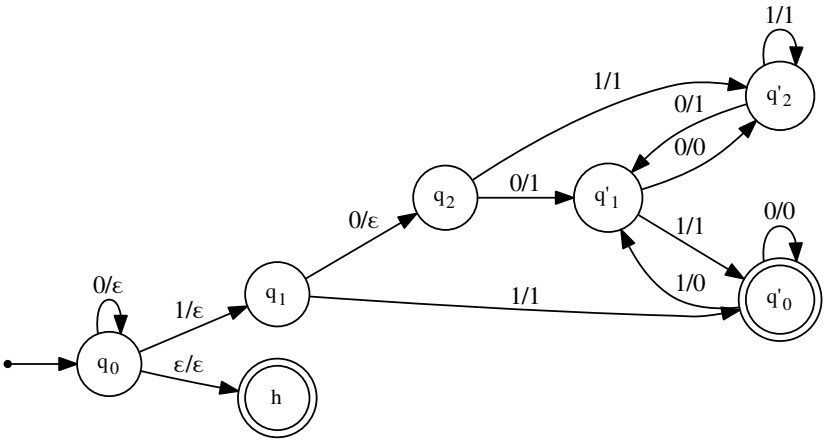
for $n \geq 0$. It follows that

$$(v(k^e \cdot n + j))_{n \geq 0} = (u_i(an + d))_{n \geq 0} \in S.$$

Since S is of finite cardinality, it must be that $K_k(\mathbf{v})$ is also of finite cardinality. So $\mathbf{v}$ is k-automatic. □

Our second proof is based on tranducers. A *finite-state transducer* is a generalized nondeterministic automaton with inputs and outputs on every transition; these inputs and outputs can be arbitrary words. The output of a transducer on an input x is the set of words $T(x)$ that can be formed by the concatenation of outputs associated with some factorization of x However, only computations that end in a final state give outputs. For a language L we define $T(L) = \bigcup_{x \in L} T(x)$.

Example 47. Let's look at a finite-state transducer that maps $(3n)_2$ to $(n)_2$, but $(3n+1)_2$ and $(3n+2)_2$ to $\emptyset$. We need to keep track of the "carries" unaccounted for in long division by 3, as well as whether or not we have output a 1 yet (to avoid outputting non-canonical representations, that is, representations of n with leading zeroes). A state is of the form q_i or q_i', where i is the unaccounted carry, and the prime indicates we have already output a 1. An input of ϵ is treated differently.



Theorem 48. *If L is a regular language, and T is a finite-state transducer, then $T(L)$ is regular.*

We omit the proof. It can be found (more or less) in the textbook, or in the literature under the name "Nivat's theorem".

We can now give the second proof of Theorem 46.

Proof #2: transducer-based. Since $\mathbf{u}$ is k-automatic, each of its fibers

$$I_d = \{(n)_k \ : \ u(n) = d\}$$

is regular. For each I_d, apply a finite-state transducer T that maps an input of the form $an + b$ to n, and every other input to $\emptyset$. (Such a transducer is a simple generalization of the one depicted in Example 47.) We get

$$T(I_d) = \{(n)_2 : u(an + b) = d\}.$$

Putting the fibers back together as a DFAO (as in the proof of Theorem 15), we get a DFAO generating $(u(an + b))_{n\geq 0}$. □

We can now prove a kind of converse of Theorem 46.

Theorem 49. *Let $a \geq 1$ be an integer, and let $(u(n))_{n\geq 0}$ be a sequence such that $(u(an + i))_{n\geq 0}$ is k-automatic for $0 \leq i < a$. Then $(u(n))_{n\geq 0}$ is k-automatic.*

Proof. Define $t_i(n) = u(an + i)$ for $n \geq 0$ and $0 \leq i < a$. By hypothesis each $(t_i(n))_{n\geq 0}$ is k-automatic. So each of the fibers $\{(n)_k \ : \ t_i(n) = d\}$ for $d \in \Delta$ and $0 \leq i < a$ is a regular language.

Construct a (nondeterministic) finite-state transducer mapping $(n)_k$ to $(an + i)_k^R$; this can be done by implementing the ordinary multiplication algorithm, maintaining the carries we expect in the states. It follows that each of the languages

$$X_{i,d} = \{(an + i)_k \ : \ t_i(n) = d\} = \{(an + i)_k \ : \ u(an + i) = d\}$$

is regular.

Now let $Y_d = \bigcup_{0\leq i<a} X_{i,d}$. Then Y_d is regular and $Y_d = \{(n)_k \ : \ u(n) = d\}$. So by Theorem 15 we see that $(u(n))_{k\geq 0}$ is k-automatic. □

Finally, one of the deepest results about automatic sequences is that you can transduce *the sequence itself* and still get an automatic sequence, provided the transducer obeys certain properties.

Theorem 50. *Let T be a deterministic finite-state transducer where each letter gets mapped to a word of length t, for some fixed $t \geq 1$. If $\mathbf{u} = (u(n))_{n\geq 0}$ is k-automatic, so is $T(\mathbf{u})$.*

For the proof, see Section 6.9 of the course text [6].

Corollary 51. *If $(u(n))_{n\geq 0}$ is a k-automatic sequence over Σ_d for some $d \geq 1$, so is*

$$\left((\sum_{0\leq i<n} u_i) \bmod d \right)_{n\geq 0}.$$

.

The requirement that the transducer cannot be relaxed, in general, as the following example shows. We can even use a non-uniform morphism.

Example 52. Consider the 2-automatic sequence $\mathbf{a} = (a_n)_{n\geq 0}$ that is the characteristic sequence of the powers of 2. Consider the morphism h defined by $h(0) = 0$ and $h(1) = 10$. If we apply h to $\mathbf{a}$, we get the infinite word

$$\mathbf{b} = 0\,1\,0\,1\,0^2\,1\,0^4\,1\,0^8\,1\cdots.$$

Assume $\mathbf{b}$ is 2-automatic. Then the fiber

$$I_1(\mathbf{b}) = \{(2^r + r)_2 \,:\, r \geq 0\}$$

is a regular language. Now

$$(2^r + r)_2 = 1 \quad \overbrace{00\cdots 0}^{r-\lfloor \log_2 r\rfloor - 1} \quad (r)_2$$

because $|(r)_2| = \lfloor \log_2 r \rfloor + 1$. If $I_1(\mathbf{b})$ is regular, so is $A = \{0^{r-\lfloor \log_2 r\rfloor -1}\ (r)_2 \,:\, r \geq 1\}$, which we get from removing the first 1 from each element of $I_1(\mathbf{b})$. Apply the pumping lemma (Lemma 16) to $z = 0^{r-\lfloor \log_2 r\rfloor -1}\ (r)_2$ for r sufficiently large. Then $u = 0^k$, $v = 0^\ell$, $w = 0^m(r)_2$ for some k, ℓ, m with $\ell \geq 1$. Pumping with $i = 2$ gives a word with a larger number of 0's at the front, but does not change the $(r)_2$ at the end, so $uv^2w \notin A$, a contradiction. So A is not regular, and hence $\mathbf{b}$ is not 2-automatic.

6.3 Change of base

Theorem 53. *Let $i, j \geq 1$ be integers and $k \geq 2$. A sequence is (k^i)-automatic iff it is (k^j)-automatic.*

Proof. It suffices to prove this for $j \geq 1$. If $\mathbf{a} = (a_n)_{n\geq 0}$ is (k^i)-automatic, then each of fibers

$$I_d = \{(n)_{k^i} \,:\, a_n = d\}$$

is regular. Let the morphism φ map c, for $0 \leq c < k^i$, to w, where $|w| = i$ and $[w]_k = c$. Apply φ to I_d. It is now easy to see that

$$\mathrm{rlz}(\varphi(I_d)) = \{(n)_k \,:\, a_n = d\}.$$

Since the regular languages are closed under the operation of a morphism and rlz (see Theorem 3), the fiber $\{(n)_k \,:\, a_n = d\}$ is regular. So $\mathbf{a}$ is k-automatic.

For the other direction, assume $\mathbf{a}$ is k-automatic. Then by Cobham's little theorem (Lecture 3), there exists a k-uniform morphism h prolongable on a letter c and a coding τ such that $\mathbf{a} = \tau(h^\omega(c))$. Then $h^\omega(c) = g^\omega(c)$, where $g = h^i$. So $\mathbf{a} = \tau(g^\omega(c))$, which again by Cobham's little theorem shows that $\mathbf{a}$ is k^i-automatic. □

6.4 Notes

The material about the Tower of Hanoi can be found in [4].

Chapter 7

Formal power series

Let K be a field, such as $\mathbb{Q}, \mathbb{R}, \mathbb{C}$, or a finite field. Then

- $K[X]$ denotes the ring of *polynomials* in one indeterminate X, with coefficients taken from K. A polynomial is *monic* if its leading coefficient (that is, the coefficient of the term of highest degree in X) is 1.
- $K(X)$ denotes the field of *rational functions* (quotients of polynomials) in X, with coefficients in K.
- $K[[X]]$ denotes the ring of *formal power series* in X, with coefficients in K: formal expressions of the form $\{\sum_{n\geq 0} a_n X^n : a_n \in K\}$, where convergence is not considered. This is a ring, and not (in general) a field, because (for example), X does not have an inverse.
- $K((X))$ denotes the field of *formal Laurent series* in X, with coefficients in K: formal expressions of the form $\{\sum_{n\geq -n_0} a_n X^n : a_n \in K\}$, where $n_0 < \infty$ is an integer. In other words, we only allow finitely many negative exponents.

The operations of addition, subtraction, and multiplication are as usual. For example, addition in $K[[X]]$ is defined by

$$\left(\sum_{n\geq 0} a_n X^n\right) + \left(\sum_{n\geq 0} b_n X^n\right) = \sum_{n\geq 0} (a_n + b_n) X^n,$$

and multiplication is defined by

$$\left(\sum_{n\geq 0} a_n X^n\right)\left(\sum_{n\geq 0} b_n X^n\right) = \sum_{n\geq 0}\left(\sum_{i+j=n} a_i b_j\right) X^n.$$

Note that both $\mathrm{GF}(q)[[X]]$ and $\mathrm{GF}(q)((X))$ can be viewed as vector spaces over the field $\mathrm{GF}(q)(X)$. Exercise: check that all the axioms for a vector space are satisfied. We will need this later in our proof of Christol's theorem.

Sometimes, we study $K[[X^{-1}]]$ and $K((X^{-1}))$ instead of $K[[X]]$ and $K((X))$. This is not significantly different, and so if we state theorems in one case, they generally apply to the other case. Note that the same rational function will, in general, have two completely different expansions depending on whether it is in $K((X))$ or $K((X^{-1}))$. For example,

$$\frac{1}{1-X} = 1 + X + X^2 + \cdots \quad \text{in } \mathbb{Q}((X))$$
$$\frac{1}{1-X} = -X^{-1} - X^{-2} - X^{-3} - \cdots \quad \text{in } \mathbb{Q}((X^{-1})).$$

Remark 54. We have $K(X) \subset K((X))$. In other words, a rational function always has a formal Laurent series expansion. If you wish to play with this in `Maple`, you can use the command `series`. To get a formal Laurent series expansion of p/q in $K((X))$, use the command

```
series(p/q, X=0, n);
```

to get n terms of the expansion.

To get a formal Laurent series expansion of p/q in $K((X^{-1}))$, use the command

```
series(p/q, X=infinity, n);
```

to get n terms of the expansion.

7.1 Algebraic numbers

We say that a real number x is *algebraic* over $\mathbb{Q}$ if there exist rational numbers $a_0, a_1, \ldots, a_n$, with $a_n \neq 0$, such that $a_n x^n + \cdots + a_x + a_0 = 0$. Of course, by multiplying through by the lcm of the denominators of the rational a_i, we can actually assume that the a_i are integers. The unique monic polynomial p of minimal degree such that $p(x) = 0$ is called the *minimal polynomial* for x.

The real algebraic numbers form a field, often written as $\mathbb{A}$: the sum, difference, product, and quotient of two algebraic numbers is also algebraic.

Example 55. The following are some examples of algebraic numbers and their corresponding minimal polynomials:

$2/3$	$3X - 2$
$\sqrt[3]{2}$	$X^3 - 2$
$\frac{1+\sqrt{5}}{2}$	$X^2 - X - 1$

If $x \in \mathbb{R}$ is not algebraic, it is *transcendental*. Examples of transcendental numbers include

$$\begin{array}{cc} e & \text{(Hermite, 1873)} \\ \pi & \text{(Lindemann, 1882)} \\ \ln 2 & \text{(Weierstrass, 1885)} \end{array}$$

In general, it is very difficult to tell, given a real number defined in some way, whether it is algebraic or transcendental. For example, the status of Euler's constant $\gamma \doteq 0.57721$ is not currently known.

7.2 Algebraic formal Laurent series

The power series analogue of a rational real number is a rational function. Similarly, there is a power series analogue of an algebraic real number. We say that a formal Laurent series $F(X)$ is *algebraic* over $K(X)$ if there exist rational functions $A_0, A_1, \ldots, A_n \in K(X)$, with $A_n \neq 0$, such that $A_0 + A_1 F + \cdots + A_n F^n = 0$. Of course, by multiplying through by all the denominators of the A_i, we may assume that the A_i are actually polynomials.

Like the algebraic real numbers, the algebraic formal Laurent series also form a field.

Example 56. Let

$$\begin{aligned} f(X) &= \frac{1 - \sqrt{1 - 4X}}{2X} \\ &= 1 + X + 2X^2 + 5X^3 + 14X^4 + 42X^5 + \cdots \\ &= \sum_{n \geq 0} C_n X^n \quad \text{where } C_n = \frac{\binom{2n}{n}}{n+1}, \text{ the } n\text{'th Catalan number.} \end{aligned}$$

Then f satisfies the equation $Xf^2 - f + 1 = 0$, so f is algebraic. The "other root" of the equation $Xf^2 - f + 1 = 0$ is $X^{-1} - f$, and of course is also algebraic.

7.3 Finite fields

These fields exist precisely when the number of elements is p^n, for $n \geq 1$ and p is a prime number. Up to isomorphism, the field $\mathrm{GF}(p^n)$ is unique. (Sometimes it is written as $\mathcal{F}_{p^n}$.) The number p is called the *characteristic* of the field $\mathrm{GF}(p^n)$.

The case of $n = 1$ is particularly simple: here $\mathrm{GF}(p)$ just corresponds to doing arithmetic modulo p. For $n \geq 2$, however, $\mathrm{GF}(p^n)$ does *not* correspond to arithmetic modulo p^n (rookie mistake). To obtain a finite field with p^n elements, find an irreducible polynomial $q(X)$ of degree n, and do arithmetic modulo $q(X)$ and modulo p.

There are several facts about the finite field $\mathrm{GF}(p^n)$ that are crucial to know. First is that if $z \in \mathrm{GF}(p^n)$ then

$$\overbrace{z + z + \cdots + z}^{p \text{ summands}} = pz = 0.$$

Of course, this also applies for $z \in \mathrm{GF}(p^n)[X]$, or $z \in \mathrm{GF}(p^n)(X)$ or $z \in \mathrm{GF}(p^n)[[X]]$ or $z \in \mathrm{GF}(p^n)((X))$.

Second is that the multiplicative group of $\mathrm{GF}(q)$ has $q-1$ elements, so that if $z \in \mathrm{GF}(q)^*$, then $z^{q-1} = 1$ and so

$$z^q = z \quad \text{for all } z \in \mathrm{GF}(q) \ . \tag{7.1}$$

Be careful: this is not true (for example) for $z \in \mathrm{GF}(q)[X]$: the polynomial z^q is not the same as the polynomial z — for one thing, their degrees are different.

Third is that for $y, z \in \mathrm{GF}(p^n)$ we have

$$(y+z)^p = y^p + z^p. \tag{7.2}$$

To see this, use the binomial theorem:

$$(y+z)^p = \sum_{0 \le n \le p} \binom{p}{n} y^n z^{p-n} = y^p + z^p,$$

because each binomial coefficient $\binom{p}{n}$ is divisible by p, except $\binom{p}{0} = \binom{p}{p} = 1$. Naturally, the identity (7.2) also extends to elements of $\mathrm{GF}(p^n)[X]$, $\mathrm{GF}(p^n)(X)$, $\mathrm{GF}(p^n)[[X]]$, and $\mathrm{GF}(p^n)((X))$.

Eq. (7.2) has the following very important implication for polynomials $s(X) \in \mathrm{GF}(q)[X]$ and power series in $\mathrm{GF}(q)[[X]]$ and $\mathrm{GF}(q)((X))$:

$$s(X)^q = s(X^q). \tag{7.3}$$

To see this for polynomials, observe that

$$\begin{aligned} (aX+bY)^q &= (aX)^q + (bY)^q \quad \text{(by Eq. (7.2))} \\ &= a^q X^q + b^q Y^q \\ &= aX^q + bY^q, \quad \text{(by Eq. (7.1))} \end{aligned}$$

and use induction on the degree of the polynomial. With a bit more work, by taking better and better approximations, one can show that Eq. (7.3) also holds for $s(X)$ a formal power series or a formal Laurent series.

Example 57. For $p = 2$ and $n = 2$ the unique irreducible polynomial of degree n over $\mathrm{GF}(p)$ is $r(X) = X^2 + X + 1$. Let α be a zero of r. Here is a multiplication table in $\mathrm{GF}(p^n)$:

	0	1	α	$\alpha+1$
0	0	0	0	0
1	0	1	α	$\alpha+1$
α	0	α	$\alpha+1$	1
$\alpha+1$	0	$\alpha+1$	1	α

If you wish to experiment with finite fields in Maple, you can find an irreducible polynomial r and then use the command `Rem(f, r, X) mod p;` to compute f mod r and mod p.

7.4 Algebraic formal Laurent series

A classical result, whose proof we leave as an exercise, is the following:

Theorem 58. *Let $f \in \mathrm{GF}(q)((X))$ be a formal Laurent series. Then f is a rational function iff $f = \sum_{n \geq -n_0} a_n X^n$ and the sequence $(a_i)_{i \geq 0}$ is ultimately periodic.*

Some formal Laurent series are algebraic over $\mathrm{GF}(q)(X)$. We give two examples.

Example 59. Let $q = p = 2$ and $f(X) = X + X^2 + X^4 + X^8 + \cdots$.

We then find

$$\begin{aligned} f^2 = & \quad X^2 + X^4 + X^8 + \cdots \\ f = X + & X^2 + X^4 + X^8 + \cdots, \end{aligned}$$

so adding term-by-term gives $f + f^2 = X$, or $f^2 + f + X = 0$. Thus f is a quadratic element of $\mathrm{GF}(2)[[X]]$. Since the coefficients of f are not ultimately periodic, by Theorem 58 we see that it is not a rational function, and hence $f^2 + f + X = 0$ is actually the minimal polynomial of f.

Example 60. Recall the Rudin-Shapiro sequence $(r_n)_{n \geq 0}$ from Lecture 1. We saw previously that

$$\begin{aligned} r_0 &= 0 \\ r_{2n} &= r_n \\ r_{4n+1} &= r_n \\ r_{4n+3} &= 1 - r_{2n+1} \end{aligned}$$

Let $R(X) = \sum_{n \geq 0} r_n X^n \in \mathrm{GF}(2)[[X]]$ be the Rudin-Shapiro power series. We will show that $R(X)$ is algebraic.

A general way to do this is to use "decimation": break the power series up into pieces based on the residue class (mod q) of the indices. In this case $q = 2$, so we use "bifurcation":

$$\begin{aligned} R(X) &= \sum_{n \geq 0} r_n X^n \\ &= \sum_{n \geq 0} r_{2n} X^{2n} + \sum_{n \geq 0} r_{2n+1} X^{2n+1} \\ &= \sum_{n \geq 0} r_n X^{2n} + X \sum_{n \geq 0} r_{2n+1} X^{2n} \\ &= R(X^2) + X S(X^2) \\ &= R(X)^2 + X S(X)^2, \end{aligned}$$

where we have introduced $S(X) = \sum_{n\geq 0} r_{2n+1}X^n$. Now do the same thing to $S(X)$:

$$\begin{aligned}
S(X) &= \sum_{n\geq 0} r_{2n+1}X^n \\
&= \sum_{n\geq 0} r_{4n+1}X^{2n} + \sum_{n\geq 0} r_{4n+3}X^{2n+1} \\
&= \sum_{n\geq 0} r_n X^{2n} + X\sum_{n\geq 0}(1 - r_{2n+1})X^{2n} \\
&= R(X^2) + X\sum_{n\geq 0} X^{2n} - X\sum_{n\geq 0} r_{2n+1}X^{2n} \\
&= R(X^2) + \frac{X}{1-X^2} - XS(X^2) \\
&= R(X)^2 + \frac{X}{(1+X)^2} + XS(X)^2.
\end{aligned}$$

(Recall that $1 = -1$ in GF(2).) We have now found the following system of equations:

$$R = R^2 + XS^2 \tag{7.4}$$

$$S = R^2 + \frac{X}{(1+X)^2} + XS^2 \tag{7.5}$$

At this point, there are two ways we can proceed, in order to find an algebraic equation for R. Either we can use some clever substitution to solve this system of algebraic equations, or we can use a general computational tool (Gröbner bases) to solve the system.

We give both approaches. From Eq. (7.4) we get $XS^2 = R + R^2$, and replace the XS^2 in Eq. (7.5) with $R + R^2$. This gives

$$S = R^2 + \frac{X}{(1+X)^2} + R + R^2 = R + \frac{X}{(1+X)^2}.$$

Now substitute this expression for S into Eq. (7.4) to get

$$\begin{aligned}
R &= R^2 + X\left(\frac{X}{(1+X)^2}\right)^2 \\
&= R^2 + XR^2 + \frac{X^3}{(1+X)^4},
\end{aligned}$$

which gives us the equation $(1+X)^5R^2 + (1+X)^4R + X^3 = 0$ for R.

For the other approach, we can use the `Groebner` package of `Maple`, as follows:

```
sys := [R^2+X*S^2+R, (1+X)^2*S+(1+X)^2*R^2+(1+X)^2*X*S^2+X];
A := Groebner[Basis](sys, lexdeg([S],[R]), characteristic=2);
A[1];
```

which outputs $(X^5+X^4+X+1)R^2+(X^4+1)R+X^3$. After factoring the coefficients, which can be done in `Maple` with

```
Factor(coeff(A[1],R,2)) mod 2;
Factor(coeff(A[1],R,1)) mod 2;
```

we see that this equation is the same as the one we found above.

Let us now give an example for characteristic 3.

Example 61. The *Cantor sequence* $(c_n)_{n\geq 0}$ is defined by

$$c_n = \begin{cases} 1, & \text{if } (n)_3 \in \{0,2\}^*; \\ 0, & \text{otherwise.} \end{cases}$$

Let $C(X) = \sum_{n\geq 0} c_n X^n = 1 + X^2 + X^6 + X^8 + X^{18} + \cdots$. We easily check that

$$\begin{aligned} c_{3n} &= c_n \\ c_{3n+1} &= 0 \\ c_{3n+2} &= c_n, \end{aligned}$$

so that $(c_n)_{n\geq 0}$ is a 3-automatic sequence.

We find

$$\begin{aligned} C(X) &= \sum_{n\geq 0} c_n X^n \\ &= \sum_{n\geq 0} c_{3n} X^{3n} + \sum_{n\geq 0} c_{3n+1} X^{3n+1} + \sum_{n\geq 0} c_{3n+2} X^{3n+2} \\ &= \sum_{n\geq 0} c_n X^{3n} + \sum_{n\geq 0} c_n X^{3n+2} \\ &= C(X^3) + X^2 C(X^3) \\ &= C(X)^3 + X^2 C(X)^3 \\ &= C(X)^3 (1+X^2). \end{aligned}$$

Hence we have shown that $(1+X^2)C^3 - C = 0$. Dividing out by $C \neq 0$, we get $(1+X^2)C^2 = 1$, or $C = (1+X^2)^{-1/2}$.

7.5 Exercises

1. The ordinary paperfolding sequence on $\{0,1\}$ is the sequence $(p_n)_{n\geq 1}$ defined by $p_{2n} = p_n$ for $n \geq 1$ and $p_{4n+1} = 0$ for $n \geq 0$ and $p_{4n+3} = 1$ for $n \geq 0$.

   Consider the formal power series $P(X) = \sum_{n\geq 1} p_n X^n$ over $GF(2)[[X]]$. Show that P is algebraic over $GF(2)[X]$ by explicitly deriving an algebraic equation, with *polynomial* coefficients, of which P is a root.

Chapter 8

Christol's theorem

We are now ready for the statement and proof of Christol's theorem, which is one of the four big results of the course.

Theorem 62 (Christol, 1979)**.** *Let Δ be a nonempty finite set. Let $\mathbf{a} = (a_i)_{i\geq 0}$ be a sequence over Δ. Let $p \geq 2$ be a prime number. Then $\mathbf{a}$ is p-automatic iff there exist an integer $n \geq 1$ and an injective map $\beta : \Delta \to \mathrm{GF}(p^n)$ such that $\sum_{i\geq 0} \beta(a_i)X^i$ is algebraic over $\mathrm{GF}(p^n)(X)$.*

There are some minor errors in the textbook presentation of the proof, which we have corrected here. We prove the theorem in two parts, one for each direction.

8.1 Preliminaries

Before we can prove Christol's theorem, we need some useful lemmas.

We first introduce the Cartier operators on Laurent series over $\mathrm{GF}(q)$.

Definition 63. Let $A(X) = \sum_{i\geq -n_0} a_i X^i$. From now on we will write this as $\sum_i a_i X^i$, with the understanding that only finitely many a_i, for $i < 0$, are nonzero. Then

$$\Lambda_r(A) := \sum_i a_{qi+r} X^i.$$

(If we wished to be completely explicit, we could write instead

$$\Lambda_r(A) := \sum_{i\geq -(r+n_0)/q} a_{qi+r} X^i,$$

specifying the range of i in detail. But this becomes more of an annoyance as we proceed, so we don't do it.)

Note that the Cartier operator Λ_r is a *linear operator*: $\Lambda_r(A + B) = \Lambda_r(A) + \Lambda(B)$.

Lemma 64. *We have $A(X) = \sum_{0\leq r<q} X^r(\Lambda_r(A))^q$.*

Proof. We have

$$\begin{aligned} A(X) &= \sum_i a_i X^i \\ &= \sum_{0 \le r < q} \sum_i a_{qi+r} X^{qi+r} \\ &= \sum_{0 \le r < q} X^r \sum_i a_{qi+r} X^{qi} \\ &= \sum_{0 \le r < q} X^r \left(\sum_i a_{qi+r} X^i \right)^q \qquad \text{(by Eq. (7.3))} \\ &= \sum_{0 \le r < q} X^r \cdot (\Lambda_r(A))^q. \end{aligned}$$

□

Corollary 65. *If $A \in \mathrm{GF}(q)[X]$ is a polynomial of degree d, then $\Lambda_r(A)$ is a polynomial of degree $\le d/q$.*

8.2 One direction

We are now ready to prove the $\Longrightarrow$ direction of Christol's theorem.

Proof. Choose n sufficiently large so that $|\Delta| \le p^n$ and injective map $\beta : \Delta \to \mathrm{GF}(p^n)$. Thus, without loss of generality, we can assume $\Delta \subseteq \mathrm{GF}(p^n)$. We want to show that $\sum_{i \ge 0} a_i X^i$ is algebraic over $\mathrm{GF}(p^n)(X)$. Since $(a_i)_{i \ge 0}$ is p-automatic, it is also q-automatic for $q = p^n$. So the q-kernel of $\mathbf{a}$ is finite, say

$$K_k(\mathbf{a}) = \{\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(d)}\}$$

with $\mathbf{a} = \mathbf{a}^{(1)}$. Write $\mathbf{a}^{(i)} = (a_n^{(i)})_{n \ge 0}$. Define

$$A_j(X) = \sum_{n \ge 0} a_n^{(j)} X^n$$

for $1 \le j \le d$. Now

$$\begin{aligned} A_j(X) &= \sum_{0 \le r < q} \sum_{m \ge 0} a_{qm+r}^{(j)} X^{qm+r} \\ &= \sum_{0 \le r < q} X^r \sum_{m \ge 0} a_{qm+r}^{(j)} (X^q)^m. \end{aligned}$$

But each $(a_{qm+r}^{(j)})_{m \ge 0}$ is one of the $\mathbf{a}^{(i)}$. So $A_j(X)$ is a $\mathrm{GF}(q)(X)$-linear combination of the $A_i(X^q)$, that is,

$$A_j(X) \in \langle A_1(X^q), \ldots, A_d(X^q) \rangle, \quad 1 \le j \le d, \tag{8.1}$$

where $\langle\cdot\rangle$ denotes the vector space generated by the items inside the brackets. Now substitute X^q for X in Eq. (8.1). We get

$$A_j(X^q) \in \langle A_1(X^{q^2}), \ldots, A_d(X^{q^2})\rangle, \quad 1 \le j \le d,$$

and hence by transitivity, we get

$$A_j(X) \in \langle A_1(X^{q^2}), \ldots, A_d(X^{q^2})\rangle, \quad 1 \le j \le d.$$

Repeating this reasoning d times, we get

$$A_j(X^{q^k}) \in \langle A_1(X^{q^{d+1}}), \ldots, A_d(X^{q^{d+1}})\rangle, \quad 1 \le j \le d,\ 0 \le k \le d.$$

But the dimension of

$$\langle A_1(X^{q^{d+1}}), \ldots, A_d(X^{q^{d+1}})\rangle$$

as a vector space over $\mathrm{GF}(q)(X)$ is at most d (the number of generators), so the $d+1$ formal series

$$A_j(X), A_j(X^q), \ldots, A_j(X^{q^d})$$

are linearly related, for each j. In particular, for $j = 1$, this gives a linear relation for $A = A_1$, namely

$$\sum_{0\le i\le d} p_i(X)A(X^{q^i}) = 0$$

for some polynomials $p_i \in \mathrm{GF}(q)[X]$, not all 0. But $A(X^{q^i}) = A(X)^{q^i}$ by Eq. (7.3). So

$$\sum_{0\le i\le d} p_i(X)A(X)^{q^i} = 0,$$

showing that A algebraic of degree at most q^d, where d is the size of the q-kernel.

This completes the proof of one direction of Christol's theorem. □

8.3 The other direction

For the other direction, we need some more lemmas.

Lemma 66. *Let G, H be formal Laurent series over* $\mathrm{GF}(q)$. *Then*

$$\Lambda_r(G^q \cdot H) = G \cdot \Lambda_r(H).$$

Proof. Write

$$G(X) = \sum_k g_k X^k$$

and

$$H(X) = \sum_j h_j X^j.$$

Then

$$\begin{aligned}
\Lambda_r(G^q \cdot H) &= \Lambda_r\left(\left(\sum_k g_k X^k\right)^q \left(\sum_j h_j X^j\right)\right) \\
&= \Lambda_r\left(\left(\sum_k g_k X^{qk}\right)\left(\sum_j h_j X^j\right)\right) \quad \text{(by Eq. (7.3))} \\
&= \Lambda_r\left(\sum_i X^i \sum_{\substack{j,k \\ i=qk+j}} g_k h_j\right) \\
&= \sum_i X^i \sum_{\substack{j,k \\ qi+r=qk+j}} g_k h_j.
\end{aligned}$$

Now observe that $qi + r = qk + j$ iff $j = qt + r$, where $i = k + t$. Thus

$$\begin{aligned}
\sum_i X^i \sum_{\substack{j,k \\ qi+r=qk+j}} g_k h_j &= \sum_k X^{k+t} \sum_t g_k h_{qt+r} \\
&= \sum_k g_k X^k \sum_t h_{qt+r} X^t \\
&= \left(\sum_k g_k X^k\right)\left(\sum_t h_{qt+r} X^t\right) \\
&= G \cdot \Lambda_r(H).
\end{aligned}$$

□

The next result is sometimes called Ore's lemma.

Lemma 67. *Suppose $A(X) = \sum_{i \geq 0} a_i X^i$ is algebraic over $\mathrm{GF}(q)(X)$. Then there exist an integer t and $t+1$ polynomials $B_0(X), \ldots, B_t(X)$, not all 0, such that*

$$B_0 A + B_1 A^q + \cdots + B_t A^{q^t} = 0.$$

Furthermore we can choose the B_i such that $B_0 \neq 0$.

Proof. If A is algebraic, then the series

$$A^q, A^{q^2}, \ldots$$

cannot all be linearly independent. So there exists a nontrivial relation

$$B_0 A + B_1 A^q + \cdots + B_t A^{q^t} = 0. \tag{8.2}$$

with the $B_i \in \mathrm{GF}(q)[X]$, not all 0.

It remains to show such a relation exists with $B_0 \neq 0$. Assume that we have a relation like Eq. (8.2) with t minimal, and let $j = \min_{\{} i : B_i \neq 0\}$.

Assume, to get a contradiction, that $j > 0$. By the definition of j we have

$$\sum_{j \le i \le t} B_i A^{q^i} = 0. \tag{8.3}$$

Using Lemma 64, we write

$$B_j = \sum_{0 \le r < q} X^r (\Lambda_r(B_j))^q.$$

Since $B_j \neq 0$, it follows that $\Lambda_r(B_j) \neq 0$ for some r. From Eq. (8.3) we have

$$\Lambda_r \left(\sum_{j \le i \le t} B_i A^{q^i} \right) = 0,$$

so if $j \neq 0$ then by Lemma 66 we get

$$\sum_{j \le i \le t} \Lambda_r(B_i) A^{q^{i-1}} = 0.$$

But this is a new relation for A, of smaller degree, with the coefficient $\Lambda_r(B_j) \neq 0$, a contradiction.

So our assumption $j > 0$ must be false, and so $j = 0$. Thus $B_0 \neq 0$. □

Finally, we need one more result.

Lemma 68. *Let $\mathbf{a} = (a_n)_{n \ge 0}$ be a sequence over $\mathrm{GF}(q)$, and set $A(X) = \sum_{n \ge 0} a_n X^n$. Then $\mathbf{a}$ is q-automatic iff there exists a finite collection of Laurent power series $\mathcal{F}$ such that*

(i) $A(X) \in \mathcal{F}$; and

(ii) $\Lambda_r(C) \in \mathcal{F}$ for all $C \in \mathcal{F}$, $0 \le r < q$.

Proof. $\Longrightarrow$: Let the q-kernel $K_q(\mathbf{a})$ be $\{\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(r)}\}$ with $\mathbf{a}^{(1)} = \mathbf{a}$ and write $\mathbf{a}^{(i)} = (a_n^{(i)})_{n \ge 0}$ for $1 \le i \le r$.

Take $\mathcal{F}$ to be the collection of power series $\{\sum_{n \ge 0} a_n^{(i)} X^n : 1 \le i \le r\}$. Then $A(X) \in \mathcal{F}$. We have

$$\begin{aligned} \Lambda_r \left(\sum_{n \ge 0} a_n^{(i)} X^n \right) &= \sum_{n \ge 0} a_{qn+r}^{(i)} X^n \\ &= a_n^{(j)} X^n \in \mathcal{F}, \end{aligned}$$

for some j depending on i, r.

$\Longleftarrow$: Suppose such an $\mathcal{F}$ exists. Choose i, t such that $i \geq 0$ and $0 \leq t < q^i$. For $n \geq 0$ define $b_n = a_{q^i n+t}$, and set $\mathbf{a} = (a_n)_{n\geq 0}$ and $\mathbf{b} = (b_n)_{n\geq 0}$. Thus $\mathbf{b}$ is an element of the q-kernel of $\mathbf{a}$. Furthermore set $A(X) = \sum_{n\geq 0} a_n X^n$ and $B(X) = \sum_{n\geq 0} b_n X^n$.

Let $z = c_0 c_1 \cdots c_{i-1} \in \Sigma_q^*$ be a word such that $[z]_q = t$. Define

$$B'(X) = \Lambda_{r_0}(\Lambda_{r_1}(\cdots(\Lambda_{r_{i-1}}(A))\cdots)).$$

Then $B' \in \mathcal{F}$ by hypotheses (i) and (ii).

However, it is easy to see that $B' = B$. So $B \in \mathcal{F}$. Since $\mathcal{F}$ is finite, and since the map that sends a sequence to its corresponding formal power series is injective, it follows that the q-kernel of $\mathbf{a}$ is also finite. So $\mathbf{a}$ is q-automatic. □

We now have everything we need to prove the $\Longleftarrow$ direction of Christol's theorem.

Proof. Suppose $A(X) = \sum_{i\geq 0} a_i X^i$ is algebraic. From Ore's lemma, there is a relation

$$\sum_{0\leq i\leq t} B_i(X) A(X)^{q^i} = 0 \tag{8.4}$$

with the $B_i \in \mathrm{GF}(q)[X]$ and $B_0 \neq 0$.

Define $G(X) = A(X)/B_0(X) \in \mathrm{GF}(q)((X))$. This is well-defined since $B_0 \neq 0$. Then $A = GB_0$. Substituting GB_0 for A in Eq. (8.4), we see that

$$\sum_{0\leq i\leq t} B_i (GB_0)^{q^i} = 0,$$

which implies that

$$B_0^2 G + \sum_{1\leq i\leq t} B_i (GB_0)^{q^i} = 0,$$

and hence

$$\begin{aligned} G &= -\left(\sum_{1\leq i\leq t} B_i (GB_0)^{q^i}\right) B_0^{-2} \\ &= -\sum_{1\leq i\leq t} B_i G^{q^i} B_0^{q^i-2} \\ &= \sum_{1\leq i\leq t} C_i G^{q_i}, \end{aligned}$$

where $C_i = B_i B_0^{q^i-2}$. Note that C_i is a polynomial because $q^i \geq q \geq 2$.

Here is the plan. We will create a finite (but rather large) set S of Laurent series, containing $A = \sum_{i\geq 0} a_i X^i$ as a member, and show that S is mapped into itself by Λ_r. By Lemma 68, this implies that the q-kernel of A is finite, and hence that A is q-automatic.

Define

$$N = \max(\deg B_0, \deg C_1, \deg C_2, \ldots, \deg C_t).$$

The set S is defined as follows:

$$S = \{H \in \mathrm{GF}(q)((X)) : \text{ there exist polynomials } D_i \in \mathrm{GF}(q)[X], \deg D_i \le N,\ 0 \le i \le t, \text{ such that } H = \sum_{0 \le i \le t} D_i G^{q^i}\}.$$

Note that S is finite, and furthermore that $A = B_0 G \in S$.

We now show $\Lambda_r(S) \subseteq S$. Let $H \in S$. Then

$$\begin{aligned}
\Lambda_r(H) &= \Lambda_r\left(D_0 G + \sum_{1 \le i \le t} D_i G^{q^i}\right) \\
&= \Lambda_r\left(\sum_{1 \le i \le t} (D_0 C_i + D_i) G^{q^i}\right) \\
&= \sum_{1 \le i \le t} \Lambda_r((D_0 C_i + D_i) G^{q^i}) \quad \text{(by linearity of } \Lambda_r) \\
&= \sum_{1 \le i \le t} \Lambda_r(D_0 C_i + D_i) G^{q^{i-1}} \quad \text{(by Lemma 66)}.
\end{aligned}$$

Now $\deg D_0, \deg C_i, \deg D_i \le N$. So $\deg(D_0 C_i + D_i) \le 2N$. So from Corollary 65 we get $\deg \Lambda_r(D_0 C_i + D_i) \le 2N/q \le N$, because $q \ge 2$. It now follows from Lemma 68 that $\mathbf{a}$ is q-automatic. □

8.4 An application of Christol's theorem

Let $g(X) = \sum_{n \ge 0} g_n X^n$ and $h(X) = \sum_{n \ge 0} h_n X^n$ be two formal power series. The *Hadamard product* $g \odot h$ is defined as follows:

$$(g \odot h)(X) = \sum_{n \ge 0} g_n h_n X^n.$$

Theorem 69. *Let g, h be two algebraic formal power series in* $\mathrm{GF}(q)[[X]]$. *Then $g \odot h$ is algebraic.*

Proof. If g, h are algebraic, then by Christol's theorem the sequences $(g_n)_{n \ge 0}$ and $(h_n)_{n \ge 0}$ are both q-automatic So $(g_n h_n)_{n \ge 0}$ is q-automatic, by the product construction for automata. So $\sum_{n \ge 0} g_n h_n X^n$ is algebraic by Christol's theorem. □

Remark 70. This theorem does not hold, in general, for fields of characteristic 0. As an example, consider

$$F(X) = \sum_{n \ge 0} \binom{2n}{n} X^n = (1 - 4X)^{-1/2},$$

which is algebraic over any field K. But $F \odot F = \sum_{n \ge 0} = \binom{2n}{n}^2 X^n$ is transcendental over $\mathbb{Q}(X)$.

8.5 Notes

Christol's theorem is from [18]. Also see [19].

8.6 Exercises

1. Recall the definition of strange numbers $(s_n)_{n\geq 0}$ from Lecture 1. Note: 0 is strange by the definition, so $s_0 = 1$. Define the formal power series $S(X) = \sum_{n\geq 0} s_n X^n$. Find an algebraic equation, with polynomial coefficients, for S over $GF(2)(X)$.

2. Recall the ordinary paperfolding sequence $(p_n)_{n\geq 1}$ defined by $p_0 = 0$, $p_{2n} = p_n$ for $n \geq 1$ and $p_{4n+1} = 0$ for $n \geq 0$ and $p_{4n+3} = 1$ for $n \geq 0$.

   Consider the formal power series $P(X) = \sum_{n\geq 1} p_n X^n$ over $GF(2)[[X]]$. Show that P is algebraic over $GF(2)(X)$ by explicitly deriving an algebraic equation, with *polynomial* coefficients, of which P is a root.

Chapter 9

Transcendence in finite characteristic

One of the nicest applications of Christol's theorem is to a topic from number theory: namely, transcendence in finite characteristic.

Let us recall some basic facts about transcendence in the real numbers. If $\alpha \in \mathbb{R}$ is not algebraic over $\mathbb{Q}$, then it is *transcendental*. Many of the classic numbers of mathematics such as π and e are known to be transcendental, but the proofs are hard and are considered a triumph of 19th century mathematics.

Instead, one might study transcendence in finite characteristic: which power series in $\mathrm{GF}(q)[[X^{-1}]]$ (or $\mathrm{GF}(q)((X^{-1}))$ are transcendental? It turns out that this question is often much easier in $\mathrm{GF}(q)$ than over $\mathbb{Q}$.

Going back to $\mathbb{R}$ for a moment, recall that a very famous function is the Riemann zeta function defined by

$$\zeta(s) = \sum_{n \geq 1} \frac{1}{n^s},$$

for which the special values $\zeta(2) = \pi^2/6$, $\zeta(4) = \pi^4/90$, are known. More generally, it is known that for all positive even s there exists a rational number r such that $\zeta(s) = \pi^s \cdot r$.

This suggests trying to find an analogue, in $\mathrm{GF}(q)[[X^{-1}]]$ of the Riemann zeta function in finite characteristic. For example, instead of integers, we could use polynomials over $\mathrm{GF}(q)$. Instead of positive integers, we could use *monic* polynomials over $\mathrm{GF}(q)$ (that is, those whose leading coefficient is 1). This suggests defining

$$\zeta_q(s) = \sum_{\substack{P \in \mathrm{GF}(q)[X] \\ P \text{ monic}}} \frac{1}{P^s}.$$

Thus, for example,

$$\begin{aligned} \zeta_2(1) &= \frac{1}{1} + \frac{1}{X} + \frac{1}{X+1} + \frac{1}{X^2} + \frac{1}{X^2+1} + \frac{1}{X^2+X} + \frac{1}{X^2+X+1} + \frac{1}{X^3} + \cdots \\ &= 1 + X^{-2} + X^{-3} + X^{-4} + X^{-5} + X^{-9} + X^{-10} + \cdots \in \mathrm{GF}(2)[[X^{-1}]]. \end{aligned}$$

This $\zeta_q(s)$ is called the *Carlitz zeta function*.

To continue the analogy, recall that the Riemann zeta function $\zeta(s)$ admits a famous infinite product formula

$$\zeta(s) = \prod_{p \text{ prime}} (1 - p^{-s})^{-1},$$

known as the Euler product. To see this, write

$$(1 - p^{-s})^{-1} = 1 + p^{-s} + p^{-2s} + \cdots,$$

and use the unique factorization of natural numbers.

The Carlitz zeta function has an analogous product formula. What is the analogue of a prime number? It is an *irreducible polynomial*, one that cannot be factored as the product of polynomials of lower degree. It is not hard to see that

$$\zeta_q(s) = \prod_{\substack{P \in \mathrm{GF}(q)[X] \\ P \text{ monic and irreducible}}} (1 - P^{-s})^{-1}.$$

Carlitz proved that if $q-1 \mid s$, then then there is a rational function R such that $\zeta_q(s) = \Pi_q^s \cdot R$, where

$$\Pi_q = \prod_{k \geq 1} \left(1 - \frac{X^{q^k} - X}{X^{q^{k+1}} - X}\right).$$

As another point of analogy, note that $q - 1 = |\mathrm{GF}(q)^*|$ and $2 = |\mathbb{Z}^*|$, where $*$ denotes the set of invertible elements. For more of these amazing analogies, see [47].

Wade proved that Π_q is transcendental over $\mathrm{GF}(q)(X)$. We can prove this using Christol's theorem.

We'll need the notation of *formal derivative* of a power series. This is a linear map from $K[[X^{-1}]]$ to $K[[X^{-1}]]$ that is defined just like the derivative (with respect to X in calculus): If $A(X) = \sum_{n \geq 0} X^{-n}$, then

$$A'(X) := \sum_{n \geq 1} -n a_n X^{-n-1} = \sum_{m \geq 2} (1 - m) a_{m-1} X^{-m}.$$

Proposition 71. *If $A(X)$ is algebraic over $\mathrm{GF}(q)(X)$, then so are $A'(X)$ and $A'(X)/A(X)$.*

Proof. For $A'(X)$, we can appeal directly to Christol's theorem. Write $A(X) = \sum_{n \geq 0} a_n X^n$; since A is algebraic, we know that $(a_n)_{n \geq 0}$ is q-automatic. Hence $((1 - m)a_{m-1})_{m \geq 2}$ is q-automatic, because automatic sequences are closed under shifts and $((1 - m) \bmod p)_{m \geq 2}$ is a periodic sequence. So again by Christol's theorem, we see that $A'(X)$ is algebraic.

The claim for $A'(X)/A(X)$ follows because the algebraic elements of $\mathrm{GF}(q)((X^{-1}))$ form a field. □

Remark 72. The quantity A'/A is sometimes called the logarithmic derivative, because it is the (formal) derivative of $\log A$. The nice thing about the logarithmic derivative is that it can turn products into sums: we have

$$\frac{(AB)'}{AB} = \frac{AB' + BA'}{AB} = \frac{A'}{A} + \frac{B'}{B}.$$

We can now prove Wade's theorem.

Theorem 73. *The formal series* $\Pi_q(X)$ *is transcendental over* $\mathrm{GF}(q)(X)$.

Proof. Write $q = p^t$ for some t. We have

$$\begin{aligned}\left(1 - \frac{X^{q^k} - X}{X^{q^{k+1}} - X}\right)' &= -\left(\frac{X^{q^k} - X}{X^{q^{k+1}} - X}\right)' \\ &= -\frac{(X^{q^{k+1}} - X)(X^{q^k} - X)' - (X^{q^k} - X)(X^{q^{k+1}} - X)'}{(X^{q^{k+1}} - X)^2} \\ &= -\frac{(X^{q^{k+1}} - X)(-1) - (X^{q^k} - X)(-1)}{(X^{q^{k+1}} - X)^2} \\ &= -\frac{(X - X^{q^{k+1}}) + (X - X^{q^k})}{(X^{q^{k+1}} - X)^2} \\ &= \frac{X^{q^{k+1}} - X^{q^k}}{(X^{q^{k+1}} - X)^2}.\end{aligned}$$

Hence

$$\begin{aligned}\frac{\left(1 - \frac{X^{q^k} - X}{X^{q^{k+1}} - X}\right)'}{\left(1 - \frac{X^{q^k} - X}{X^{q^{k+1}} - X}\right)} &= \frac{\frac{X^{q^{k+1}} - X^{q^k}}{(X^{q^{k+1}} - X)^2}}{\frac{X^{q^{k+1}} - X^{q^k}}{(X^{q^{k+1}} - X)^2}} \\ &= \frac{1}{X^{q^{k+1}} - X^{q^k}}.\end{aligned}$$

Hence

$$\begin{aligned}\frac{\Pi_q'}{\Pi_q} &= \sum_{k \geq 1} \frac{1}{X^{q^{k+1}} - X} \\ &= \left(\sum_{k \geq 1} \frac{1}{X^{q^{k+1}} - X}\right) - \frac{1}{X^{q^k} - X}.\end{aligned}$$

Assume, to get a contradiction, that Π_q is algebraic. Then Π_q' is algebraic, and hence Π_q'/Π_q is algebraic. Then

$$B := \sum_{k \geq 1} \frac{1}{X^{q^k} - X}$$

is algebraic. Now

$$\begin{aligned}
B &= \sum_{k\geq 1} \frac{1}{X^{q^k} - X} \\
&= \sum_{k\geq 1} \frac{1}{X^{q^k}(1 - (\frac{1}{X})^{q^k-1})} \\
&= \sum_{k\geq 1} \frac{1}{X^{q^k}} \sum_{n\geq 0} (\frac{1}{X})^{n(q^k-1)} \\
&= \frac{1}{X} \sum_{k\geq 1} \frac{1}{X^{q^k-1}} \sum_{n\geq 0} (\frac{1}{X})^{n(q^k-1)} \\
&= \frac{1}{X} \sum_{\substack{k\geq 1 \\ n\geq 0}} (\frac{1}{X})^{(n+1)(q^k-1)} \\
&= \frac{1}{X} \sum_{\substack{k\geq 1 \\ m\geq 1}} (\frac{1}{X})^{m(q^k-1)} \\
&= \frac{1}{X} \sum_{r\geq 1} X^{-r} \sum_{\substack{k,m\geq 1 \\ m(q^k-1)=r}} 1 \\
&= \frac{1}{X} \sum_{r\geq 1} X^{-r} \sum_{\substack{k\geq 1 \\ (q^k-1)|r}} 1 \\
&= \frac{1}{X} \sum_{r\geq 1} c(r) X^{-r},
\end{aligned}$$

where

$$c(r) = \sum_{\substack{r\geq 1 \\ (q^k-1)|r}} 1.$$

By Christol's theorem, it must be that $(c(r) \bmod p)_{r\geq 1}$ is q-automatic. Then by Theorem 17, we have $(c(q^n - 1) \bmod p)_{n\geq 0}$ is ultimately periodic. But

$$\begin{aligned}
c(q^n - 1) &= \sum_{\substack{k\geq 1 \\ (q^k-1)|q^n-1}} 1 \\
&= \sum_{\substack{k\geq 1 \\ k|n}} 1 \\
&= d(n),
\end{aligned}$$

the number of divisors of n. Here we have used the fact that $(q^k - 1) \mid (q^n - 1)$ iff $k \mid n$.

It follows that $(d(n) \bmod p)_{n\geq 1}$ is ultimately periodic. So there exist $t \geq 1$, $n_0 \geq 1$ such that $d(n+it) \equiv d(n) \pmod{p}$ for all $n \geq n_0$ and $i \geq 1$. Choose $i = ni'$. Then

$$d(n(1+i't)) \equiv d(n) \pmod{p}$$

for all $i' \geq 1$. By Dirichlet's theorem on primes in arithmetic progressions, we can find $i' \geq 1$ such that $p' = 1 + i't$ is a prime, and $p' \geq n_0$. Choose $n = p'$. Then $d(n(1+i't)) = d(p'^2) \equiv d(p') \pmod{p}$. But the square of a prime has 3 divisors, while a prime has 2 divisors. So $3 \equiv 2 \pmod{p}$, a contradiction. So Π_q is not algebraic.

Thus we have proven that Π_q is transcendental. □

9.1 Transcendence over $\mathbb{Q}(X)$

Although Christol's theorem is about transcendence over $\mathrm{GF}(q)(X)$, it can also be applied to other settings. The main tool is the following result.

Theorem 74. *Let $F(X) \in \sum_{i\geq 0} f_i X^i \in \mathbb{Z}[[X]]$ be a formal power series with integer coefficients, and let p be a prime number. Let $F_p(X) = \sum_{i\geq 0}(f_i \bmod p)X^i$ be the reduction of F mod p. If F is algebraic over $\mathbb{Q}(X)$, then F_p is algebraic over $\mathrm{GF}(p)(X)$.*

Proof. Suppose F is algebraic over $\mathbb{Q}(X)$. Then there exist an integer d and $d+1$ polynomials $B_0, B_1, \ldots, B_d$, not all 0, in $\mathbb{Q}[X]$ such that

$$B_0 + B_1 F + \cdots + F_d F^d = 0 \tag{9.1}$$

By clearing denominators of the rational number coefficients of the polynomials, we can assume that $B_i \in \mathbb{Z}[X]$ for $0 \leq i \leq d$. We can assume that the gcd, e, of all of the coefficients of the resulting B_i is 1; if not divide Eq. (9.1) by e. Now consider Eq. 9.1 mod p. Since not all of the coefficients are divisible by p, this gives a nontrivial relation for F_p (of possibly lower degree). Hence F_p is algebraic over $\mathrm{GF}(p)(X)$. □

Example 75. Define $\theta_3(X) = \sum_{-\infty<n<\infty} X^{n^2}$, the classical theta-series. Assume, to get a contradiction, that $\theta_3(X)$ is algebraic over $\mathbb{Q}(X)$. Since $\theta_3(X) = 1 + 2\sum_{n\geq 1} X^{n^2}$, it follows that $\sum_{n\geq 1} X^{n^2}$ is algebraic. Then by Theorem 74, it follows that $\sum_{n\geq 1} X^{n^2}$ is algebraic over $\mathrm{GF}(2)(X)$. By Christol's theorem, then, the characteristic sequence of the squares is 2-automatic. But by a result we prove in the next section, this is not so. Hence $\theta_3(X)$ is transcendental over $\mathbb{Q}(X)$.

9.2 Gaps and automatic sequences

We prove a very useful and general theorem, due to Cobham, about gaps in automatic sequences.

Theorem 76. *Let $\mathbf{x} = (x(n))_{n\geq 0}$ be a k-automatic sequence over Δ. Let $d \in \Delta$. Define α_j to be the position of the j'th occurrence of d in $\mathbf{x}$. (More formally, if $|\mathbf{x}[0..t-1]|_d = j-1$ and $\mathbf{x}[t] = d$, then $\alpha_j = t$.) Then either*

$$\limsup_{n\to\infty} \frac{|\mathbf{x}[0..n-1]|_d}{\log n} < \infty$$

or

$$\liminf_{j\to\infty} \alpha_{j+1} - \alpha_j < \infty$$

(or both).

Remark 77. It is possible for both alternatives to hold. For example, consider the characteristic sequence of the set

$$\{2^n : n \geq 1\} \cup \{2^n - 1 : n \geq 1\},$$

and $d = 1$.

Proof. Since $\mathbf{x}$ is k-automatic, there is a k-DFAO $M = (Q, \Sigma, \Delta, \delta, q_0, \tau)$ generating it. By introducing a new initial state, if necessary, we can assume that $\delta(q_0, 0) = q_0$, and no transitions other than this one enter q_0.

Call a state q of M *recurrent* if there are infinitely many n such that $\delta(q_0, (n)_k) = q$, and *transient* otherwise. Evidently at least one state must be recurrent. Fix a letter $d \in \Delta$. Call a state q *fecund* with respect to d if there are two distinct strings of the same length, w and w', such that $\tau(\delta(q, w)) = \tau(\delta(q, w')) = d$, and *barren* otherwise.

Case (a): There exists a recurrent state that is fecund. Then there are infinitely many n such that $\delta(q_0, (n)_k) = q$, and w and w' such that $\tau(\delta(q, w)) = \tau(\delta(q, w')) = d$. Let $|w| = |w'| = t$ and let $[w]_k = a$ and $[w']_k = b$. Without loss of generality, say $a < b$. Then $\tau(\delta(q_0, (n)_k w)) = \tau(\delta(q_0, (n)_k w') = d$ for infinitely many n, and so $x(2^t n + a) = x(2^t n + b)$ for infinitely many n. But then $\liminf_{j\to\infty} \alpha_{j+1} - \alpha_j \leq b - a < \infty$.

Case (b): Every recurrent state is barren. Then since there are only finitely many states in M, there must exist an i such that every state of the form $\delta(q_0, (n)_k)$ for $n \geq k^i$ is recurrent. Let us find an upper bound on the number of possible occurrences of d in $\mathbf{x}[0..k^j - 1]$. If $k^i < n < k^j$, write $(n)_k = wx$, where $|w| = i$ and $|x| \leq j - i$. Since q is barren, at most one x of each length has $\tau(\delta(q_0, wx)) = d$. Thus there are at most $k^i(j - i + 1)$ possible n for which $\mathbf{x}[n] = d$. The total number of possible n up to k^j for which $\mathbf{x}[n] = d$ is therefore $k^i + k^i(j - i + 1)$. This is $O(j)$, and the result now follows. □

Corollary 78. *Let p be a polynomial with rational coefficients such that $p(\mathbb{N}) \subseteq \mathbb{N}$. Then the characteristic sequence $\mathbf{c}$ of the set $\{p(i) : i \geq 0\}$ is k-automatic iff $\deg p < 2$.*

Proof. If $\deg p < 2$, then this characteristic sequence is ultimately periodic, and hence k-automatic.

Otherwise assume $\deg p \geq 2$, and $\mathbf{c}$ is k-automatic. Take $d = 1$ in Theorem 76. We have $\alpha_{j+1} - \alpha_j = p(j'+1) - p(j')$ for $j' = j + c$, and j sufficiently large and c a constant. But this difference is a polynomial of degree $(\deg p) - 1$ and hence goes to ∞ as j gets large, so the theorem tells us that

$$\limsup_{n \to \infty} \frac{|\mathbf{x}[0..n-1]|_d}{\log n} < \infty.$$

But $|\mathbf{x}[0..n-1]|_d = \Theta(n^{1/s})$, where $s = \deg p$, a contradiction. Hence $\mathbf{c}$ cannot be k-automatic. □

9.3 Notes

For the transcendence of the formal analogue of π, see [2].

Theorem 76 is from [23].

Chapter 10

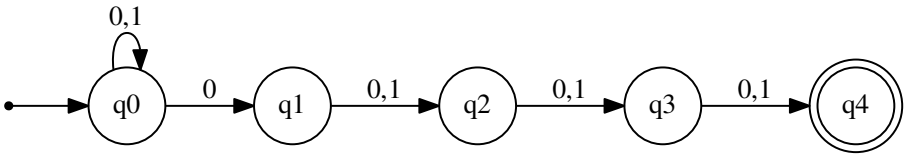
The logical approach to automatic sequences

In this lecture we will explore the work of Presburger, Büchi, Bruyère, Point, Villemaire, and Hodgson relating automata to logic.

10.1 Nondeterministic automata

We will need another model of automata: nondeterministic finite automata (NFA). These are like deterministic automata, except that from each state there can be 0, 1, or more transitions on a single symbol. Acceptance is defined by the *existence* of *some* path from the initial state to the final state, labeled by the input. The transition function δ now has domain $Q \times \Sigma$ and range 2^Q. We can convert an NFA to an equivalent DFA with a construction called the "subset construction"; its states are subsets of states of the original automaton. In the worst case, an NFA with n states can require as many as 2^n states in an equivalent DFA.

Example 79. A classic example: a nondeterministic machine accepting the set of all binary strings having a 0 symbol in the fourth position from the end:



NFA's are often more expressive than DFA's, but have the following drawback: it is

not easy to take the complement of a nondeterministic automaton. In order to do so, one must first convert the NFA to a DFA which, as we've observed above, could result in an exponential blow-up in the number of states.

10.2 First-order logic

The great mathematician Hilbert had two dreams:

- To show that every true statement is provable (killed by Gödel)
- To provide an algorithm to decide if an input statement is provable (killed by Turing)

Nevertheless, some subclasses of problems are complete and decidable — i.e., an algorithm exists that is guaranteed to prove or disprove any well-formed assertion.

By *first-order logic*, we mean the set of all formulas formed from

- any finite number of variables that can take values in some domain;
- equality defined on variables;
- possibly other comparison operators that can be applied to variables, such as less than, greater than, etc., depending on domain;
- possibly other functions applied to the variables, such as addition or multiplication;
- logical operations such as `and` ($\wedge$), `or` ($\vee$), `logical implication` ($\implies$), `iff` ($\iff$), and `not` ($\neg$);
- quantifiers, such as `for all` ($\forall$) and `there exists` ($\exists$).

In a formula φ in first-order logic,

- Variables can be either *bound* (associated with a quantifier) or *unbound.*
- If all variables are bound, then we can assign a truth value to the formula φ: it is either true or false.
- If some variables are unbound, then we can consider the set $S(\varphi)$ of all values of the variables for which φ is true.

A first-order logical theory is *decidable* if there is an algorithm that, given a well-formed formula with all variables bound, will decide its truth.

In the case of unbound variables, we'd also like it if we could algorithmically construct the representations of all integers for which the formula is true.

10.3 Presburger arithmetic

Presburger arithmetic is $\mathrm{FO}(\mathbb{N}, +)$, the first-order theory of the natural numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$ with addition. Sometimes Presburger arithmetic is written to include $<$, the "less-than" operator. But it is not really needed, since the assertion $x < y$ is equivalent to $\exists z\ (z \neq 0)\ \wedge\ y = x + z$.

Example 80 (The Chicken McNuggets Problem)**.** This is a famous problem in elementary arithmetic books in the US:
At McDonald's, Chicken McNuggets are available in packs of either 6, 9, or 20 nuggets. What is the largest number of McNuggets that one cannot purchase?

In Presburger arithmetic we can express the "Chicken McNuggets theorem" that 43 is the largest integer that cannot be represented as a non-negative integer linear combination of $6, 9$, and 20, as follows:

$$(\forall n > 43\ \exists x, y, z \geq 0 \text{ such that } n = 6x + 9y + 20z)\ \wedge \\ \neg(\exists x, y, z \geq 0 \text{ such that } 43 = 6x + 9y + 20z). \quad (10.1)$$

Here, of course, "$6x$" is shorthand for the expression "$x + x + x + x + x + x$", and similarly for $9y$ and $20z$.

Example 81. A few more examples: x is even is represented by

$$\exists y\ x = y + y.$$

definition of the number $x = 1$:

$$(x \neq 0) \ \wedge (\forall y \ (y \neq 0) \implies y \geq x)$$

commutativity of addition: asserted by

$$\forall x \ \forall y \ (x + y = y + x).$$

Presburger proved that $\mathrm{FO}(\mathbb{N}, +, 0, 1)$ is *decidable*: that is, there exists an algorithm that, given a well-formed formula in the theory, will decide its truth. He used quantifier elimination.

J. Richard Büchi found a much simpler proof of Presburger's result, based on automata. It gives us automata for the unbound variable case, too!

The main ideas are as follows:

- represent integers in an integer base $k \geq 2$ using the alphabet $\Sigma_k = \{0, 1, \ldots, k-1\}$.
- represent n-tuples of integers as words over the alphabet Σ_k^n, padding with leading zeroes, if necessary. This corresponds to reading the base-k representations of the n-tuples *in parallel*.
- For example, the pair $(21, 7)$ can be represented in base 2 by the word

$$[1, 0][0, 0][1, 1][0, 1][1, 1].$$

Büchi's proof:

- Automata will accept words over the alphabet Σ_k^n representing n-tuples of integers
- The language accepted is the set of all n-tuples of integers for which the formula (or subformula) is true
- Parsing the formula corresponds to performing operations on automata
- For example, if automaton M corresponds to some formula φ, then $\neg\varphi$ can be obtained by changing the "finality" of M's states: a final state becomes non-final and vice-versa
- Care is needed to handle the "leading zeroes" problem

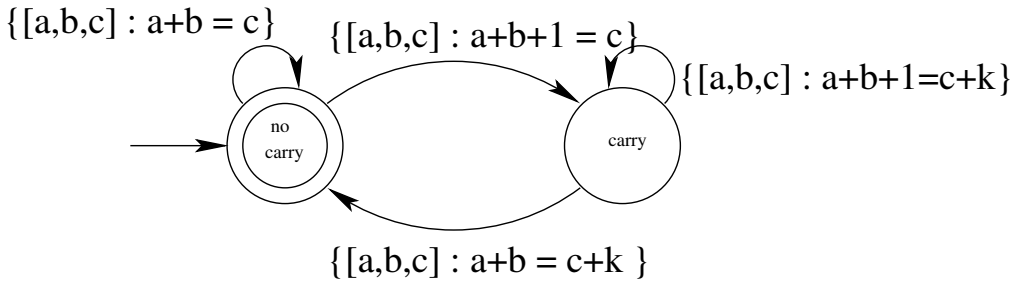
We need a way to represent pairs, triples, and in general r-tuples of integers in base k. We describe this for pairs; the extension to triples and larger r-tuples is easy. The canonical representation of (m, n) in base k is determined as follows: we write both m and n in base k, and the pad the shorter (if necessary) with leading zeros so that both are of the same length. Then the letters of the canonical representation are chosen from the alphabet $\Sigma_k \times \Sigma_k$.

For example, consider $(17, 7)_2$. The representation of 17 is 10001, and the representation of 7 is 111. We pad the shorter with zeros to get 00111. Then

$$(17, 7)_2 = [1, 0][0, 0][0, 1][0, 1][1, 1].$$

Decidability of Presburger arithmetic:

- The relation $x + y = z$ can be checked by a simple 2-state automaton depicted below, where transitions not depicted lead to a nonaccepting "dead state".



- Relations like $x = y$ and $x < y$ can be checked similarly. (exercise)

- Given a formula with free variables $x_1, x_2, \ldots, x_n$, we construct an automaton accepting the base-k expansion of those n-tuples $(x_1, \ldots, x_n)$ for which the proposition holds.

- If a formula is of the form $\exists x_1, x_2, \ldots x_n\ p(x_1, \ldots, x_n)$, then we use nondeterminism to "guess" the x_i and check them.

- If the formula is of the form $\forall p$, we use the equivalence $\forall p \equiv \neg\exists\neg p$; this may require using the subset construction to convert an NFA to a DFA and then flipping the "finality" of states.

- Ultimately, if all variables are bound, we are left with a single state machine that either accepts (formula is true) or rejects (formula is false)

The bad news:

- The worst-case running time of the algorithm above is bounded above by

$$2^{2^{\cdot^{\cdot^{\cdot^{2^{p(N)}}}}}},$$

  where the number of 2's in the exponent is equal to the number of quantifier alternations, p is a polynomial, and N is the number of states needed to describe the underlying automatic sequence.

- The bound for Presburger arithmetic can be improved to double-exponential.

A couple of additional tricks: if the last quantifiers are $\exists$, all we need to do is check to see if the resulting automaton accepts some word.

In this case, we do not need to convert an NFA to a DFA.

We can check acceptance with depth-first search, by seeing if there is a path in the automaton from the initial state q_0 to a state of F. This can be done in time linear in the size of the automaton.

Similarly, if we want to know if there are infinitely many integers for which some formula holds (which is sometimes written $\exists_\infty$) we just need to check for which states q there is a nonempty cycle beginning and ending at q (which can be done using depth-first search), and then check to see if there is a path from q_0 to q and q to a final state. Again, linear time.

Some subtleties:

Every integer has infinitely many representations!

For example, 5 in base 2 can be written as 101, 0101, 00101, and so forth.

It is best to allow all possible representations in our automata.

(If we do not, then we can run into problems working with k-tuples of integers where one integer has a larger representation than other.)

10.4 Augmenting Presburger arithmetic

As described, Presburger arithmetic isn't so interesting (although it is used, e.g., in system verification). But if we add DFAO's to the mix, using the same decision procedure, we suddenly can prove theorems people actually want to prove.

For example, we can start with a 2-DFAO M for the Thue-Morse sequence $\mathbf{t}$, write a formula for $\mathbf{t}$ having an overlap, and use the decision procedure to decide it — thus reproving Thue's 1912 result by machine. But what is the logical theory corresponding to starting with a DFAO?

Julius Richard Büchi (1924–1984) was apparently the first to consider this question. He thought one should add, to Presburger arithmetic, the function $\nu_k(n)$, which is the function computing the *exponent* of the highest power of k dividing n. For example, $\nu_2(24) = 3$.

This was a mistake.

The correct function to add is $V_k(n)$, the function computing the highest power of k, say k^e, dividing n. For example, $V_2(24) = 8$.

Exercise: show that for $k \geq 2$ the theory $\mathrm{FO}(\mathbb{N}, +, V_k)$ coincides with $\mathrm{FO}(\mathbb{N}, +, V_{k^2})$.

Theorem 82. *A set of integers is definable in* $\mathrm{FO}(\mathbb{N}, +, V_k)$ *if and only if its characteristic sequence is k-automatic.*

Proof. First we show how to construct a finite automaton M_φ corresponding to any formula φ of $\mathrm{FO}(\mathbb{N}, +, V_k)$.

The idea again is that M_φ will accept the base-k representations of all n-tuples $(x_1, x_2, \ldots, x_n)$ of natural numbers making $\varphi(x_1, x_2, \ldots, x_n)$ true.

We use the *least-significant-digit first* representation for numbers.

We observe that $\mathrm{FO}(\mathbb{N}, R_+, R_{V_k})$ is equivalent to $\mathrm{FO}(\mathbb{N}, +, V_k)$, where $R_+(x, y, z)$ is the relation $x + y = z$ and $R_{V_k}(x, y)$ is the relation $V_k(x) = y$.

We already saw automata for addition, so it suffices to give an automaton for $V_k(x) = y$. This is left to the reader as an exercise. □

Corollary 83. *The theory* $\mathrm{FO}(\mathbb{N}, +, V_k)$ *is decidable.*

Proof. We can decide if a formula in $\mathrm{FO}(\mathbb{N}, +, V_k)$ is true, just as with Presburger arithmetic, by creating the automaton associated with the formula and checking if it accepts.

Next we show how to encode a binary automatic sequence $(s(n))_{n\geq 0}$ in $\mathrm{FO}(\mathbb{N}, +, V_k)$. Actually we encode $\{n \ : s(n) = 1\}$ and we use the equivalent theory $\mathrm{FO}(\mathbb{N}, R_+, R_{V_k})$.

The basic idea, given an integer x for which $s(x) = 1$, is to encode another integer y that gives the sequence of states x encounters as it is processed by the automaton.

To do so we need new relations

$$e_{j,k}(x, y)$$

for $0 \leq j < k$. The meaning of this relation is that y is some power of k, say $y = k^e$, and the coefficient of k^e in the base-k representation of x is equal to j.

We also need $\lambda_k(x)$, which is the greatest power of k occurring with a nonzero coefficient in the base-k representation of x. By definition we set $\lambda_k(0) = 1$.

We also need $P_k(x)$, which is true if x is a power of k and false otherwise.

Now we show how to express $e_{j,k}(x, y)$ and $\lambda_k(x)$ and $P_k(x)$ in $\mathrm{FO}(\mathbb{N}, +, V_k)$.

$P_k(x)$ is the easiest. We have $P_k(x)$ is the same as $V_k(x) = x$.

$\lambda_k(x) = y$ is the next easiest. The basic idea is to observe that if we trap x between two powers of k, say $k^e \leq x < k^{e+1}$, then $\lambda_k(x) = k^e$.

So $\lambda_k(x) = y$ is the same as

$$(P_k(y) \ \wedge \ (y \leq x) \ \wedge \ x < ky) \ \vee \ ((x = 0) \ \wedge \ (y = 1)).$$

Finally, we can express $e_{j,k}(x, y)$ as follows: we group the powers of k appearing in x as follows: those appearing in y, those of exponent less than the one occurring in y, and those of exponent greater. So $e_{j,k}(x, y)$ is equivalent to

$$P_k(y) \ \wedge \ (\exists \ell \ \exists g \ (x = \ell + jy + g) \ \wedge \ (\ell < y) \wedge \ ((y < V_k(g)) \vee (g = 0))).$$

Now that we have these relations, we can encode the computation of a DFAO with a large formula (similar to the way we encode a Turing machine with a SAT formula):

To simplify things, we assume the DFAO has at most k states. If it has more, another trick is needed.

The idea is to create a base-k integer y that encodes the series of states encountered as we process the base-k digits of the input integer x.

If $x = \sum_{0\leq i\leq l} a_i k^i$, the input is $a_0 a_1 \cdots a_l$ and the series of states encountered is $p_0, p_1, \ldots, p_{l+1}$. Our formula should say that

(i) $p_0 = 0$;

(ii) $\delta(p_i, a_i) = p_{i+1}$ for $0 \leq i \leq l$;

(iii) $p_{l+1} \in F$.

We can encode these as follows:

(i) $e_{1,k}(y, 1)$;

(ii) $\forall t\ P_k(t)\ \wedge\ (t < z) \wedge \bigwedge_{\delta(q,b)=q'} (e_{q,k}(y, t)\ \wedge\ e_{b,k}(x, t)\ \Longrightarrow\ e_{q',k}(y, kt))$;

(iii) $\bigvee_{q\in F} e_{q,k}(y, z)$.

Finally, the formula is

$$\exists y\ \exists z\ P_k(z)\ \wedge\ (z > y)\ (z > x)\ \wedge\ (i)\ \wedge\ (ii)\ \wedge\ (iii).$$

□

Not all morphic sequences have decidable theories. Consider the morphism $a \to abcc$, $b \to bcc$, $c \to c$. The fixed point of this morphism is

$$\mathbf{s} = abccbccccbccccccbccccccccb\cdots$$

It encodes, in the positions of the b's, the characteristic sequence of the squares. So the first-order theory $\mathrm{FO}(\mathbb{N}, +, 0, 1, n \to \mathbf{s}[n])$ is powerful enough to express the assertion that "n is a square" With that, one can express multiplication, and so it is undecidable (Church, 1936).

10.5 Open Problems

1. Is the logical theory $(\mathbb{N}, +, P_2, P_3)$ decidable? Here P_k is the predicate "is a power of k". Recently this was resolved by work of Christian Schulz.

2. Is the logical theory $(\mathbb{N}, +, n \to p(n))$ decidable? Here $p(n)$ is the primality predicate, which is true if n is prime and false otherwise.

3. Is the logical theory $(\mathbb{N}, +, n \to \varphi(n))$ decidable? Here $\varphi(n)$ is Euler's phi function, counting the number of integers $\leq n$ and relatively prime to it.

4. Is the following problem decidable? Given two k-automatic sequences $(a(n))_{n\geq 0}$ and $(b(n))_{n\geq 0}$, are there integers $c \geq 1$ and $d \geq 0$ such that $a(n) = b(cn + d)$ for all n?

10.6 Notes

Hands down, the best article on the subject of this lecture is [10].

10.7 Exercises

1. Give a logical formula $\psi(x, y)$ in $\mathrm{FO}(N, +, V_2)$ that is true iff y is the second largest power of 2 appearing in the binary representation of x (or 0 if there is no second largest power).

Chapter 11

Deciding properties of automatic sequences

As we will see, the logical theory we discussed in the previous lecture is powerful enough to express many assertions about automatic sequences. Luckily, Hamoon Mousavi has created a Java prover `Walnut` that implements the decision procedure discussed yesterday, and it is publicly available. There is also a software manual available that describes its use. In this lecture, we'll look at a variety of properties of automatic sequences and prove them using `Walnut`.

11.1 Ultimate periodicity

We can write a formula for ultimate periodicity of a sequence S as follows:

$$\exists n \geq 0 \ \exists p \geq 1 \ \forall j \geq n \ S[j] = S[j+p].$$

When we translate this to `Walnut`, we need not specify $n \geq 0$ explicitly, as this is implicit in the domain $\mathbb{N}$. We translate $\exists p \geq 1 \cdots$ to

```
Ep (p >= 1) & ...
```

We translate $\forall j \geq n \cdots$ to

```
Aj (j >= n) => ...
```

Let us now prove that the Thue-Morse sequence is not ultimately periodic.

```
% cd Walnut/bin
% java Main.prover
eval tmup "En Ep (p>=1) & Aj (j >= n) => T[j] = T[j+p]":
```

Now go and check the file `tmup.txt` in the directory `Walnut/Result`, and it says "false". So the Thue-Morse sequence is not ultimately periodic.

We can use this technique to obtain bounds on how far we have to go to check a property.

Theorem 84. *If a k-DFAO of n states generates an ultimately periodic sequence S, then the preperiod and period are bounded by $k^{3 \cdot 2^{4n^2}}$.*

Proof. We can make a DFA accepting those $(j,l)_k$ such that $S[j] = S[l]$ in n^2 states. We can enforce $(j \geq n) \wedge (l = j + p)$ using a total of $4n^2$ states. Checking $\forall j$ requires some nondeterminism and another negation, giving 2^{4n^2}. Finally, checking $p \geq 1$ takes 3 states, so $3 \cdot 2^{4n^2}$ states. Such an automaton, if it accepts anything at all, must accept p and n having at most $3 \cdot 2^{4n^2}$ symbols. □

Note: there are better results due to Honkala, Sakarovitch, etc.

11.2 Squares

We can detect if a sequence has squares (that is, blocks of the form xx, where x is nonempty) using this method. As an example, let us consider the Hanoi sequence **H** discussed in Lecture 6. Let us recode the values one more time, as follows: $\mathtt{a}, \mathtt{b}, \mathtt{c}, \overline{\mathtt{a}}, \overline{\mathtt{b}}, \overline{\mathtt{c}} \to 0, 1, 2, 3, 4, 5$. If this is coded as a file `H.txt`, then we can use the predicate that asserts the existence of squares:

$$\exists i, p \ (p \geq 1) \ \wedge \ \forall t \ (t < p) \implies \mathbf{H}[i+t] = \mathbf{H}[i+p+t].$$

When we write this in `Walnut` as

```
Ei,p (p>=1) & At (t<p) => H[i+t]=H[i+p+t]
```

we get the answer `false`, so **H** is squarefree.

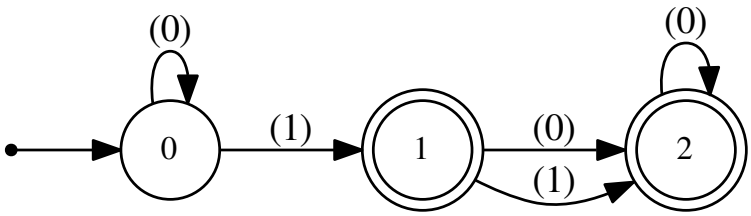
Similarly, we can write a formula for the orders of squares in a sequence S as follows:

$$(n > 0) \ \wedge \exists i \ \forall j \ (j < n) \implies S[i+j] = S[i+j+n]$$

In `Walnut`, for the Thue-Morse sequence, this is done with the command

```
eval tmsq "(n>0) & Ei Aj (j<n) => T[i+j] = T[i+j+n]":
```

Then we go and look in the `Result` directory for `tmsq.gv`.



Thus there are squares of order 2^n and $3 \cdot 2^n$ for all $n \geq 0$ in the Thue-Morse sequence. Where are they?

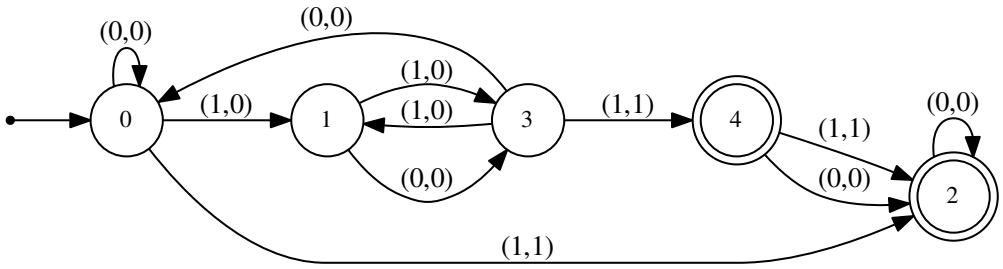
We can write a formula for the positions and orders of squares in a sequence S as follows:

$$(n > 0) \ \wedge \forall j \ (j < n) \implies S[i + j] = S[i + j + n]$$

In `Walnut`, for the Thue-Morse sequence, this is

```
eval tmsqp "(n>0) & Aj (j<n) => T[i+j] = T[i+j+n]":
```

Then we go and look in the `Result` directory for `tmsqp.gv`.



11.3 Overlaps

We can write a formula for the orders and positions of overlaps in a sequence S as follows:

$$(n \geq 1) \ \wedge \ \forall j \ (j \leq n) \implies S[i + j] = S[i + j + n]$$

When we do this in `Walnut` for the Thue-Morse sequence we type

```
eval tmover "(n>=1) & Aj (j<=n) => T[i+j] = T[i+j+n]":
```

which gives an automaton that accepts nothing.

11.4 Arbitrary fractional powers

Fractional powers are generalizations of integer powers. We say a string x is an (ℓ/p)-power if it is of length ℓ and has period p. For example, `ionization` is a $(10/7)$-power.

We say a word w avoids α powers, for $\alpha > 1$ a real number, if w has no factor that is a (ℓ/p)-power for $(\ell/p) \geq \alpha$. We say a word w avoid α^+ powers if w has no factor that is a (ℓ/p)-power for $(\ell/p) > \alpha$.

Thus, avoiding squares is avoiding 2-powers, and avoiding overlaps is avoiding 2^+-powers.

We can write a formula for a word S avoiding α-powers:

$$\neg(\exists i \ \exists n \ (n \geq 1) \ \wedge \ \forall j \ (j + n < \alpha n) \implies S[i + j] = S[i + j + n])$$

or avoiding α^+-powers:

$$\neg(\exists i\ \exists n\ (n \geq 1)\ \wedge\ \forall j\ (j+n \leq \alpha n)\implies S[i+j] = S[i+j+n])$$

In order for this to be expressible in our logical theory, we must have $\alpha = \ell/p$ for some integers ℓ, p. Then we rewrite

$$j+n < \alpha n \quad \text{as} \quad \ell j < (p-\ell)n$$

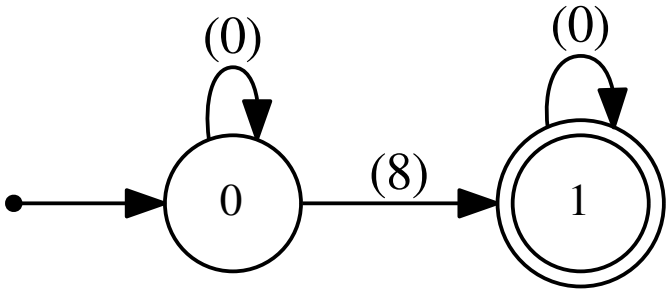
and

$$j+n \leq \alpha n \quad \text{as} \quad \ell j \leq (p-\ell)n.$$

Example: the Leech sequence:

$$0 \to 0121021201210; \quad 1 \to 1202102012021; \quad 2 \to 2010210120102$$

This sequence avoids $(15/8)^+$ powers and has infinitely many $(15/8)$-powers. We can create a file named `LE.txt` in the `Word Automata` directory that implements this morphism. Then we say
`eval le158 "?msd_13 Ei (n>=1) & Aj (8*j < 7*n) => LE[i+j] = LE[i+j+n]":`
After a reasonable delay we get the automaton



which says that there are powers $x^{15/8}$ for $|x| = 8 \cdot 13^i$ and $i \geq 0$.

11.5 Antisquares

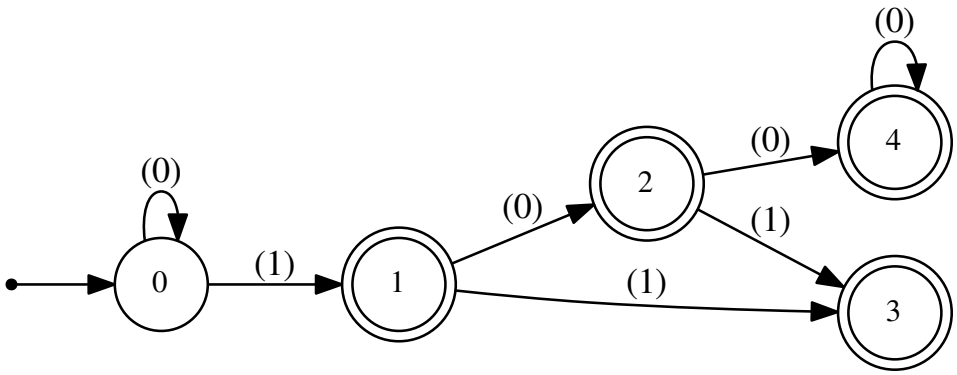
Antisquares are binary words of the form $x\overline{x}$, where $\overline{x}$ means change 0 to 1 and 1 to 0.

A formula for lengths of antisquares:

$$\exists i\ (n \geq 1)\ \wedge\ \forall j\ (j < n)\implies S[i+j] \neq S[i+j+n].$$

Let's compute antisquare orders for the Rudin-Shapiro sequence in `Walnut`:

```
Ei (n>=1) & Aj (j<n) => RS[i+j]!=RS[i+j+n]
```



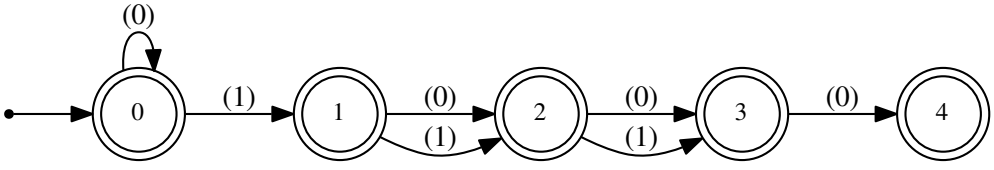
This gives the following orders of antisquares: 2^i for $i \geq 0$ and 3 and 5.

11.6 Palindromes

We can write a formula for the lengths of palindromes occurring in a sequence S as follows:

$$\exists i\ \forall j\ (j < n)\implies S[i+j] = S[(i+n)-(j+1)]$$

When we do this for the Rudin-Shapiro sequence we get



So the only palindrome lengths in Rudin-Shapiro are

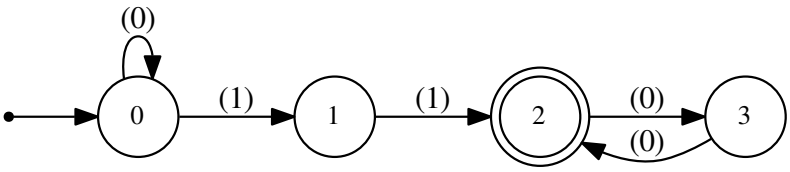
$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14\}.$$

A *maximal palindrome* in a word S is a palindrome x such that axa does not appear in S for any a.

Formula:

$$\begin{aligned}\exists i\ (\forall j\ (j < n) \implies S[i+j] = S[(i+n)-(j+1)]) \wedge \\ (\forall l\ (((l > 0) \wedge (\forall\ m\ (m < n) \implies \\ S[l+m] = S[i+m])) \implies (S[l-1] \neq S[l+n])))\end{aligned}$$

When we do this for the Thue-Morse sequence, we get only the lengths $3 \cdot 4^i$ for $i \geq 0$.



Exercise: How could you use `Walnut` to prove that the only maximal palindromes in the Thue-Morse sequence are $\mu^{2n}(010)$ and $\mu^{2n}(101)$ for $n \geq 0$?

11.7 Reversal-freeness

Formula:

$$\forall i\ \forall j\ \exists k\ (k < n)\ \wedge\ S[i+k] \neq S[(j+n)-(k+1)].$$

This says that for any word of length n beginning at position i, all other words beginning at all positions j have their reversal $S[j..j+n-1]$ differing at some position k from $S[i..i+n-1]$.

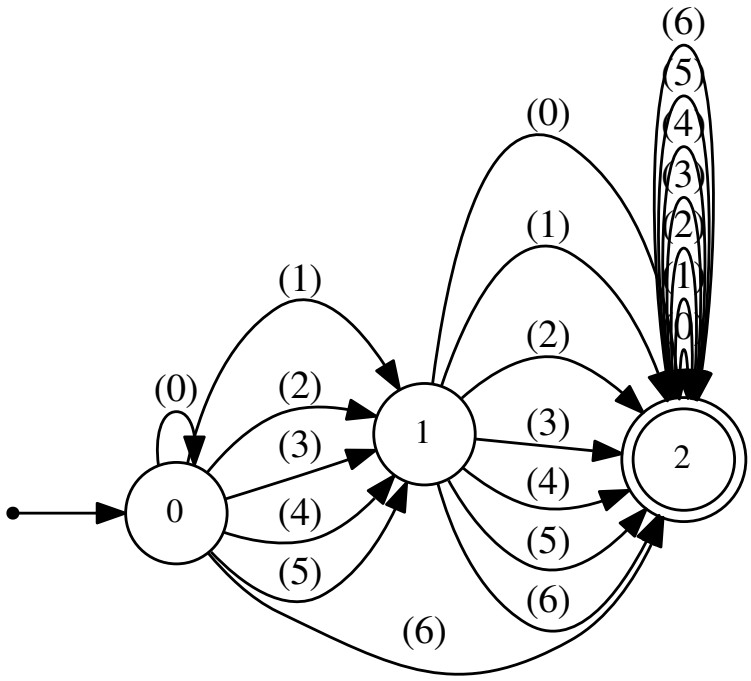
Here is an example: take the morphism defined by

$$0 \to 0001011; \qquad 1 \to 0010111.$$

This gives a 7-automatic sequence. We encode this in a file `RSR.txt`. Then we use the `Walnut` command:

```
?msd_7 Ai Aj Ek (k<n) & RSR[i+k] != RSR[(j+n)-(k+1)]
```

and we get the automaton below. So the only lengths for which words and their reversals are both present are $0, 1, 2, 3, 4, 5$.



11.8 Recurrence

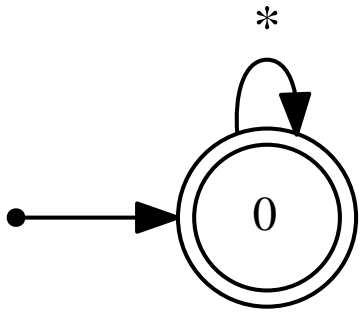
Recall: $\mathbf{x}$ is recurrent if every factor that occurs, occurs infinitely often. This is equivalent to: for every factor that occurs, there is another occurrence at a higher index. Formula:

$$\forall i \; \forall n \; \exists j \; (j > i) \wedge (\forall l (l < n) \implies S[i+l] = S[j+l]).$$

When we run

```
Ai An Ej ((j>i) & (Al (l<n) => T[i+l] = T[j+l]))
```

in `Walnut` for the Thue-Morse sequence we get the automaton



which is `Walnut`'s way to represent an automaton that accepts everything with 0 free variables.

11.9 Bordered and unbordered factors

A nonempty word x is bordered if there is a nonempty word w and a possibly empty word t such that $x = wtw$. For example, `ionization` is bordered with border `ion`.

Formula for $S[i..i+n-1]$ being bordered:

$$\exists l\ (0 < l) \wedge (l < n) \wedge (\forall j\ (j < l) \implies S[i+j] = S[(i+n+j)-l])$$

In `Walnut` we can define a macro for this:

```
def tmbord "El (0<l) & (l<n) & (Aj (j<l) => T[i+j]=T[(i+n+j)-l] )":
```

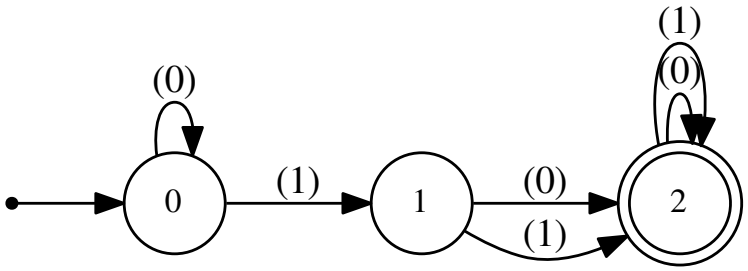
and then use it by saying

```
eval tmborders "Ei $tmbord(i,n)":
```

or

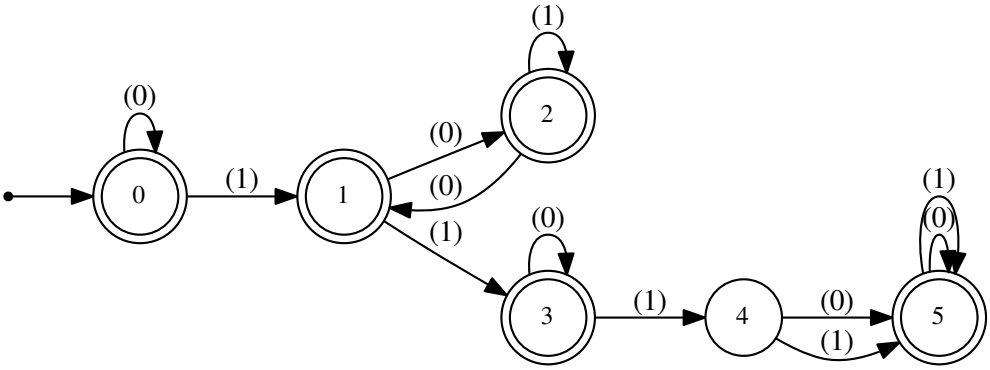
```
eval tmunbord "Ei ~$tmbord(i,n)":
```

When we do this for Thue-Morse we get



for the lengths of bordered factors of Thue-Morse, which shows that there is a bordered factor for all lengths > 1.

When we do this for unbordered factors we get



for the lengths of unbordered factors of Thue-Morse.

So we have proved: there is an unbordered factor of length n of the Thue-Morse sequence iff $(n)_2 \notin 1(01^*0)^*10^*1$. This improves a 2009 result due to Currie and Saari; they proved $\mathbf{t}$ has an unbordered factor of length n if $n \not\equiv 1 \pmod 6$.

11.10 Balanced words

A word x is *balanced* if $||y|_a - |z|_a| \leq 1$ for all equal-length factors y, z of x and all letters a.

It is not clear how to state this in first-order logic. Luckily there is an alternative characterization, which is often quoted as "x is unbalanced iff there exists a palindrome p such that both $0p0$ and $1p1$ are both factors of x". But an even simpler characterization is x is unbalanced iff there exists a word y (not necessarily a palindrome) such that both $0y0$ and $1y1$ are both factors of x. These two characterizations are easily seen to be equivalent.

Here is a formula for unbalanced factors of length n:

$$(n \geq 2) \wedge \exists i\ \exists j\ (\forall k((0 < k) \wedge (k + 1 < n)) \implies (S[i + k] = S[j + k])) \wedge S[i] = 0\ \wedge\ S[j] = 1 \wedge\ S[i + n - 1] = 0\ \wedge\ S[j + n - 1] = 1$$

In `Walnut` this is

```
(n>=2) & Ei Ej (Ak ((0<k)&(k+1<n)) => S[i+k] = S[j+k])
 & S[i]=@0 & S[i+n-1] = @0 & S[j] = @1 & S[j+n-1] = @1
```

11.11 Rich words

We can count the number of distinct palindromes occurring in a word. For example, the word `Mississippi` has the following distinct nonempty palindromes in it:

`M`, `i`, `s`, `p`, `ss`, `pp`, `sis`, `issi`, `ippi`, `ssiss`, `ississi`

Theorem 85. *Every word of length n contains, as factors, at most n distinct palindromes.*

Proof. For each index p of a word w, consider the palindromes ending at this index. Suppose at least two palindromes, x and y occur for the first time ending at p. Then wlog $|x| < |y|$. So then x is a suffix of y, so $x^R = x$ is a prefix of y, contradicting the claim that x occurred for the first time ending at p.

So at each position p at most 1 new palindrome can end. □

We say that a length-n word is `rich` if it contains, as factors, exactly n distinct nonempty palindromes.

We can therefore make a formula for the factor $S[i..i + n - 1]$ being rich as follows: at each position p there is a palindrome ending at p that doesn't occur earlier in that factor.

Exercise. Write a predicate for richness and test it on the Thue-Morse sequence. You should find that there are no rich factors of length > 16.

Exercise. Find a 2-automatic sequence where all factors are rich, and prove it using `Walnut`.

11.12 Primitive words

A nonempty word w is *primitive* if it cannot be written as x^e with $e \geq 2$. So a primitive word is a non-power.

It's easy to see that a word w is a nontrivial power if and only if there is some cyclic shift (by $0 < j < |w|$ positions) of w that is equal to w. So we can write a formula for $S[i..i+n-1]$ being a power as follows:

$$\exists j,\ 0 < j < n,\ ((\forall t < n - j\ S[i+t] = S[i+j+t]) \wedge (\forall u < j\ S[i+u] = S[i+n+u-j]))$$

A formula for being primitive is just the negation of this.

11.13 The "substitute variables" trick

Recall our formula for primitivity:

$$\neg\exists j,\ 0 < j < n,\ ((\forall t < n - j\ S[i+t] = S[i+j+t]) \wedge (\forall u < j\ S[i+u] = S[i+n+u-j]))$$

This formula is correct, but indexing the automatic sequence by four variables (as in $i+n+u-j$) could be prohibitively expensive for our algorithm when the underlying automaton has many states.

To reduce the running time, use the substitution of variables $t' = i+t$ and $u' = i+u+n$ to get

$$\neg\exists j,\ 0 < j < n,\ ((\forall t',\ i \leq t' < n+i-j,\ S[t'] = S[t'+j]) \wedge (\forall u', n+i \leq u' < n+i+j,\ S[u'-n] = S[u'-j]))$$

This one is about twice as fast for the Thue-Morse sequence.

11.14 Privileged words

A word x is *privileged* if is of length ≤ 1, or it has a border w with $|x|_w = 2$ that is itself privileged. For example, `abracadabra` has a border `abra` that appears only at the beginning and end. And `abra` has a border `a` that occurs only at the beginning and end. Finally, `a` is privileged, and so is `abra` and so is `abracadabra`.

As stated it is not obvious that we can state this property in first-order logic.

However, there is another way to state the property (due to Luke Schaeffer): a word is privileged if for all n with $1 \leq n < |w|$ there exists a word x of length $\leq n$ such x is a border of w and there is exactly one occurrence of x in the first n symbols of w and one occurrence of x in the last n symbols of w.

Exercise: write a predicate for the privileged property, and run it on the Thue-Morse word.

11.15 Closed words

A word x is called `closed` if it is of length ≤ 1, or if it has a border w with $|x|_w = 2$.

For example, `alfalfa` is a closed word because of the border `alfa`. On the other hand, although `academia` is bordered, it is not closed.

Theorem 86. *There is a closed factor of the Thue-Morse word* $\mathbf{t}$ *of every length.*

11.16 Common factors

The existence of common factors of length n between two k-automatic sequences can be checked using the assertion

$$\exists i\ \exists j\ \forall t\ (t < n) \implies R[i+t] = S[j+t].$$

Exercise. Find an appropriate bound $B(s,t)$ such that if two k-automatic sequences, generated by automata of s and t states, respectively, have a factor of length $\ell > B(s,t)$ in common, then they have arbitrarily long factors in common.

11.17 Exercises

1. Construct a two-dimensional 2-automatic sequence with the property that no row is equal to the shift of any other row, nor any column equal to the shift of any other column.

2. If $z = z[0..n-1]$ is a finite word, define its a-cyclic shift to be the word $z[a..n-1]z[0..a-1]$.

   Let $\mathbf{x}$ be an infinite word. Write a predicate in first-order logic — more precisely, the first-order theory $\mathrm{FO}(\mathbb{N}, +, n \to \mathbf{x}[n])$ — with free variables i, j, n, a, asserting the claim that the word of length n beginning at position i in $\mathbf{x}$ equals the a-cyclic shift of the word of length n beginning at position j in $\mathbf{x}$.

3. Let $\mathbf{t}$ be the Thue-Morse sequence. Using the predicate in the problem above as a subroutine, write and execute code in `Walnut` to compute a DFA that on input i and n in parallel in base 2, accepts (or outputs 1) if both of the following hold, and rejects (or outputs 0) otherwise:

   (a) all conjugates of $y = \mathbf{t}[i..i+n-1]$ appear somewhere as factors of $\mathbf{t}$;

   (b) the earliest appearance of a conjugate of y as a factor of $\mathbf{t}$ is $\mathbf{t}[i..i+n-1]$.

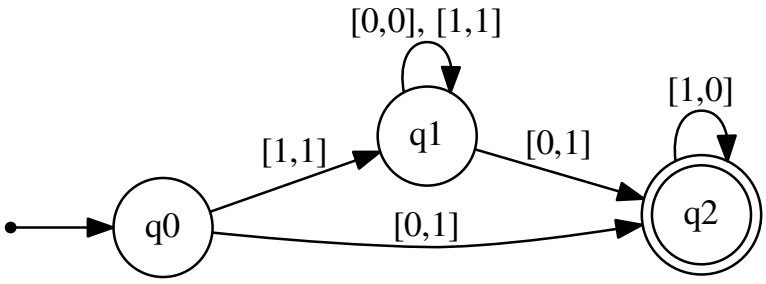
   Explain your construction in English, give the `Walnut` code, and give your automaton.

Chapter 12

The k-synchronized sequences

We study the properties of a new class of sequences, the k-synchronized sequences. These are sequences $f(n)$ for which the set $\{(n, f(n))_k : n \geq 0\}$ is regular, that is, recognized by a finite automaton. Recall that we represent pairs of integers in base-k over the alphabet $(\Sigma_k \times \Sigma_k)^*$.

Example 87. As a simple example of such a function, consider $f(n) = n + 1$. This is recognized by the following DFA for the case $k = 2$:



As we will see, many aspects of k-automatic sequences are k-synchronized. Indeed, this is the case if we can write a first-order formula in two variables n and s, which is true iff $s = f(n)$.

12.1 Appearance

The appearance function $A(n)$ of a sequence $\mathbf{x}$ is the length of the shortest prefix of $\mathbf{x}$ that contains all length-n factors of $\mathbf{x}$. We claim that $A(n)$ is k-synchronized for k-automatic

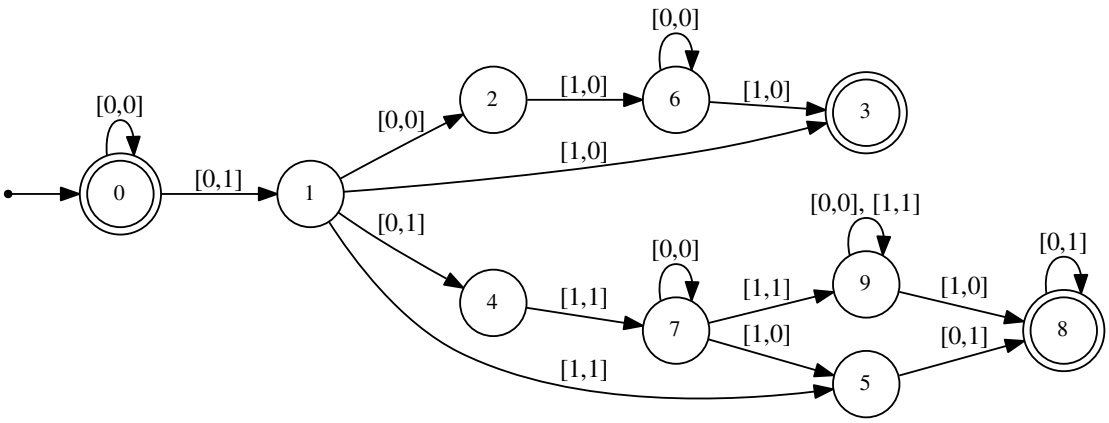
sequences. To see this, first write a first-order formula $\mathrm{pr}(n, s)$ that states that the prefix of length s contains all length-n factors of $\mathbf{x}$:

$$\forall j\ \exists i\ (i + n \le s)\ \wedge\ \forall t\ (t < n)\ \Longrightarrow\ \mathbf{x}[j + t] = \mathbf{x}[i + t].$$

Next, write the first-order formula stating that s is the smallest such:

$$\mathrm{app}(n, s) := \mathrm{pr}(n, s)\ \wedge\ \neg\, \mathrm{pr}(n, s - 1).$$

We can do this for the Thue-Morse sequence. We obtain the following automaton.



We claim that for Thue-Morse we have, for $n \ge 1$, that

$$\begin{aligned} A(2n+1) &= A(2n) + 1 \\ A(4n) &= 2A(2n) + 1 \\ A(8n+2) &= 2A(4n+2) - 1 \\ A(8n+6) &= 2A(4n+2) + 3 \end{aligned}$$

We can verify these claims with `Walnut`, as follows:

```
def pr "Aj Ei (i+n<=s) & At (t<n) => T[j+t]=T[i+t]":
def app "$pr(n,s) & ~$pr(n,s-1)":
eval test1 "A n,s,t ((n>=1) & $app(2*n,s) & $app(2*n+1,t)) => (t=s+1)":
eval test2 "A n,s,t ((n>=1) & $app(2*n,s) & $app(4*n,t)) => (t=2*s+1)":
eval test3 "A n,s,t ((n>=1) & $app(4*n+2,s) & $app(8*n+2,t)) => (t+1=2*s)":
eval test4 "A n,s,t ((n>=1) & $app(4*n+2,s) & $app(8*n+6,t)) => (t=2*s+3)":
```

From this, an easy induction gives

$$A(n) = \begin{cases} 0, & \text{if } n = 0; \\ 2, & \text{if } n = 1; \\ 7, & \text{if } n = 2; \\ 7 \cdot 2^j + i - 1, & \text{if } n = 2^j + i \text{ for } j \geq 0 \text{ and } 2 \leq i \leq 2^j + 1. \end{cases}$$

In general, recurrences like the ones above can often be guessed from a small amount of data, and then verified using `Walnut`.

12.2 Uniform recurrence

We saw in Lecture 11 that the property of being recurrent is first-order expressible, and hence decidable, for k-automatic sequences.

A stronger property of sequences is *uniform recurrence*: for each n there is a constant s such that every block of size s contains, as a factor, every factor of length n that appears anywhere in the sequence. This property is also expressible, as follows:

$$\forall n \; \exists s \; \forall i \; \forall j \; \exists \ell \; (\ell \geq i) \; \wedge \; (\ell + n \leq i + s) \; \wedge \; \forall t \; (t < n) \implies \mathbf{x}[\ell + t] = \mathbf{x}[j + t].$$

If a sequence is indeed uniformly recurrent, then its recurrence function $R(n)$ is defined to be the smallest such s. Define

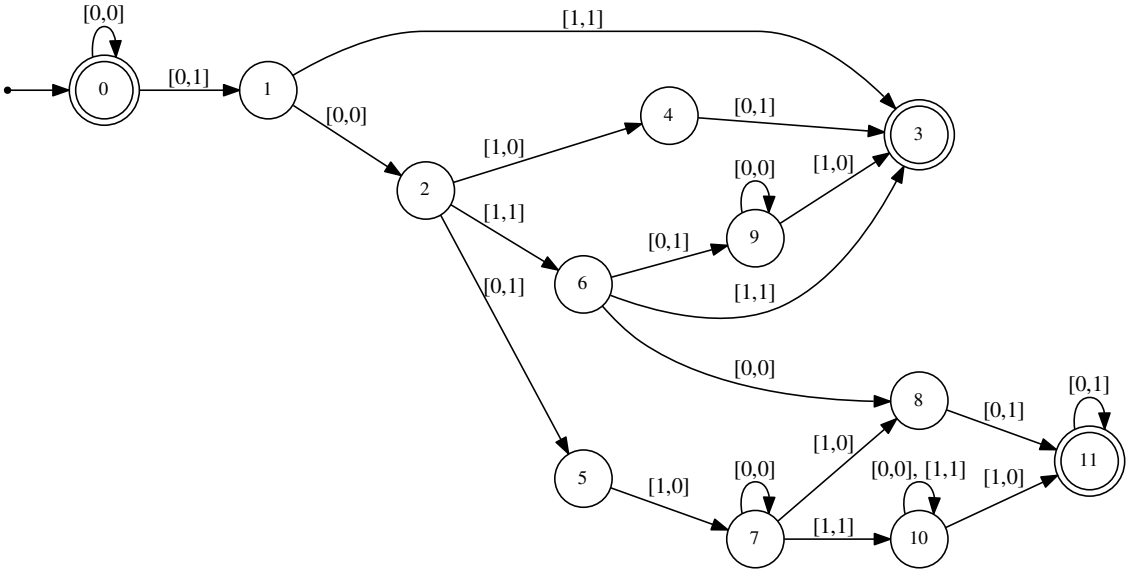
$$\mathrm{rc}(n, s) := \forall i \; \forall j \; \exists \ell \; (\ell \geq i) \; \wedge \; (\ell + n \leq i + s) \; \wedge \; \forall t \; (t < n) \implies \mathbf{x}[\ell + t] = \mathbf{x}[j + t].$$

Then $\mathrm{rc}(n, s)$ is true iff every block of size s contains, as a factor, every block of size n. We now write a formula saying that s is the smallest such:

$$\mathrm{R}(n, s) := \mathrm{rc}(n, s) \; \wedge \; \neg(\mathrm{rc}(n, s - 1)).$$

Thus R is k-synchronized.

Example 88. Let us compute the automaton R for the Thue-Morse sequence.



12.3 Fast computation of k-synchronized sequences

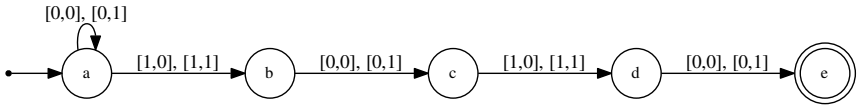
Suppose $f(n)$ is k-synchronized. How quickly can we compute $f(n)$?

Theorem 89. *If $f(n)$ is k-synchronized, we can compute it in $O(\log n)$ time and space.*

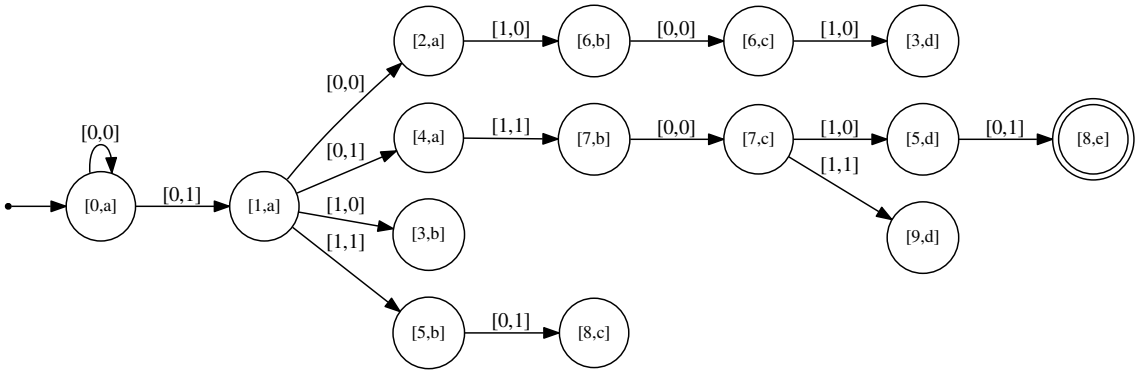
Proof. Since f is k-synchronized, there is an automaton M recognizing the language $L = \{(n, f(n))_k \ : \ n \geq 0\}$. Given n, we can now easily create an DFA M' of $O(\log n)$ states accepting those words over $(\Sigma_k \times \Sigma_k)^*$ where the first component is of the form $0^*(n)_k$ and the second component is arbitrary. Now, using the direct product construction, we can form a new DFA M'' for the intersection of these two languages. Now using any traversal algorithm for directed graphs, such as breadth-first search, we can look for a path from the initial state of M'' to any final state. Breadth-first search is particularly suited to our task, because we can maintain a queue of states to be explored further, and halt as soon as an accepting state in M'' is discovered. Such a path will be labeled with a representation of n in the first component, and a representation of $f(n)$ in the second component. □

Example 90. Continuing the example of Section 12.1, let us compute $A(10)$ for the Thue-Morse sequence.

First we create the DFA M':



Then we form the DFA M'' by breadth-first search. Only reachable states are shown.



The accepting path

$$[0,a] \xrightarrow{[0,1]} [1,a] \xrightarrow{[0,1]} [4,a] \xrightarrow{[1,1]} [7,b] \xrightarrow{[0,0]} [7,c] \xrightarrow{[1,0]} [5,d] \xrightarrow{[0,1]} [8,e]$$

is labeled 001010 in the first component, and 111001 in the second component, showing that that $A(10) = 57$.

12.4 Bounds on synchronized sequences

It turns out that the growth rate of a synchronized sequence is very constrained.

Theorem 91. *Let f be a k-synchronized sequence. Then*

(a) $f(n) = O(n)$;

(b) If $f \neq O(1)$, then there exists a constant c such that $f(n) > cn$ infinitely often.

Proof. (a) Suppose $f \neq O(n)$. Then there exists a subsequence $(n_i)_{i\geq 0}$ such that $f(n_i)/n_i \to \infty$. Suppose the DFA accepting $\{(n, f(n))_k : n \geq 0\}$ has t states; then t is the pumping lemma constant. Choose i such that $f(n_i)/n_i > k^t$, and in the pumping lemma let $z = (n_i, f(n_i))_k$. Then $|z| \geq t$, and furthermore the first component of z starts with at least t 0's, while the second component starts with a nonzero digit. When we pump

(that is, write $z = uvw$ with $|uv| \leq t$ and $|v| \geq 1$ and consider uv^2w) we only add to the number of leading 0's in the first component, but the second component's base-k value increases in size (since it starts with a nonzero digit). This implies that f is not a function, a contradiction.

(b) Since $L = \{(n, f(n))_k : n \geq 0\}$ is regular, so is $L^R = \{(n, f(n))_k^R : n \geq 0\}$. Let M be a DFA recognizing L^R, and let t be the number of states of M (which is the pumping lemma constant). Since $f \neq O(1)$, there must be an n_0 for which $f(n_0) > k^t$. Let $z = (n_0, f(n_0))_k^R$. Then $|z| > t$. Write $z = uvw$ with $|uv| \leq t$ and $|v| \geq 1$, and consider $z_i = uv^iw \in L^R$ for $i \geq 1$. Then $z_i = (a_i, b_i)_k^R$ for some integers a_i, b_i and hence $f(a_i) = b_i$. Let r, s be integers such that $k^r \leq n_0 < k^{r+1}$ and $k^s \leq f(n_0) < k^{s+1}$. Then $k^{r+(i-1)|v|} \leq a_i < k^{r+1+(i-1)|v|}$ and $k^{s+(i-1)|v|} \leq b_i < k^{s+1+(i-1)|v|}$. Then $b_i/a_i > k^{s-r-1}$, and hence $f(n) > cn$ infinitely often, where $c = k^{s-r-1}$.

□

As a corollary, we immediately get that the subword complexity of automatic sequences is small:

Corollary 92. *Let* $\mathbf{x}$ *be a* k*-automatic sequence, and let* $\rho_{\mathbf{x}}(n)$ *be the number of distinct factors of length* n *appearing in* $\mathbf{x}$*. Then* $\rho_{\mathbf{x}}(n) = O(n)$*.*

Proof. Clearly the number of distinct factors of length n is bounded above by the appearance function $t := A_{\mathbf{x}}(n)$ (because $\mathbf{x}[0..t-1]$ contains all factors of length n. However, we proved that $A_{\mathbf{x}}(n)$ is k-synchronized, and from Theorem 91 we know that $A_{\mathbf{x}}(n) = O(n)$. □

12.5 Closure properties of k-synchronized sequences

Theorem 93. *The class of* k*-synchronized sequences is closed under the following operations:*

(a) sum

(b) $\mathbb{N}$*-linear finite combination*

(c) $f(n) \to \lfloor \alpha f(n) \rfloor$ *for* α *rational*

(d) term-wise maximum and minimum

(e) running maximum: $g(n) = \max_{0 \leq i < n} f(i)$ *and running minimum:* $g(n) = \min_{0 \leq i < n} f(i)$

(f) discrete inverse: $g(n) = \min\{i : f(i) \geq n\}$

(g) composition

Proof. (a) Suppose f, g are k-synchronized. Then there exists a DFA M_1 recognizing $0^*\{(n, f(n))_k : n \geq 0\}$ and a DFA recognizing $0^*\{(n, g(n))_k : n \geq 0\}$. From this we can easily create a DFA recognizing $0^*\{(n, f(n) + g(n))_k : n \geq 0\}$.

(b) Follows immediately from (a).

(c) Write $\alpha = p/q$. If M recognizes the language $0^*\{(n, f(n))_k : n \geq 0\}$ we can call it $\text{sync}(n, s)$. then we can use the first-order formula $\text{sync}'(n, t) := (qt \leq ps) \wedge (ps < q(t+1)) \wedge \text{sync}(n, s)$.

(d) Suppose sync_1 recognizes the relation $(n, s_1) : s_1 = f(n)$ and sync_2 recognizes the relation $(n, s_2) : s_2 = g(n)$. Then $\text{sync}'(n, t) := ((t = s_1) \wedge (s_1 \geq s_2)) \vee ((t = s_2) \wedge (s_1 < s_2))$ is the relation for max. The relation for min is similar.

(e) Suppose sync recognizes the relation $(n, s) : s = f(n)$. Then

$$\text{summ}(n, t) := (\forall i \; ((i < n) \wedge \text{sync}(i, s)) \implies s \leq t) \wedge (\exists i \; (i < n) \wedge \text{sync}(i, s) \wedge s = t).$$

Running minimum is similar.

(f) Suppose sync recognizes the relation $(n, s) : s = f(n)$. Then

$$\text{di}(n, i) := \text{sync}(i, s) \wedge (s \geq n) \wedge \forall j, t \; ((j < i) \wedge \text{sync}(j, t)) \implies t < n.$$

(g) Suppose sync_1 recognizes the relation $(n, s) : s = f(n)$ and sync_2 recognizes the relation $(n, t) : t = g(n)$. Then $\text{comp}(n, t) := \text{sync}_1(n, s) \wedge \text{sync}_2(s, t)$ recognizes $(n, g(f(n))$.

□

12.6 Subword complexity is synchronized

Let $\rho_{\mathbf{x}}(n) =$ number of distinct length-n factors of $\mathbf{x}$. It is known that $\rho_{\mathbf{x}}(n) = O(n)$, which suggests it could be k-synchronized. We prove this result.

Call a length-n factor *novel* at position i if it occurs there but in no earlier location. Here is a first-order formula for novel factors:

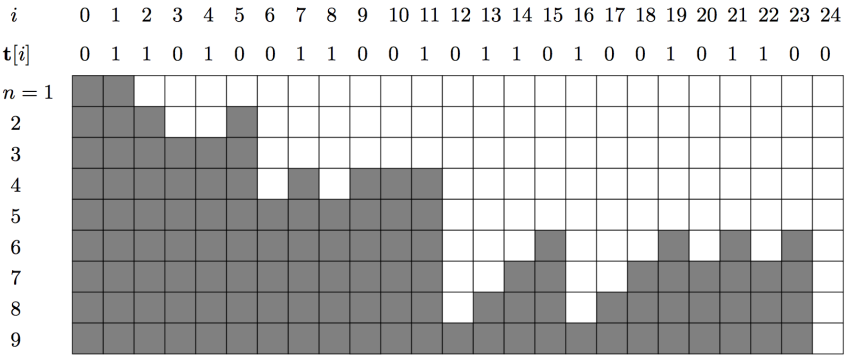
$$\{(n, i)_k : \forall j, 0 \leq j < i \quad \mathbf{x}[i..i+n-1] \neq \mathbf{x}[j..j+n-1]\}$$

Theorem 94. *In any sequence of linear complexity, the starting positions of novel occurrences of factors are "clumped together" in a bounded number of contiguous blocks.*

Example 95. Consider the Thue-Morse sequence

$$\mathbf{t} = t_0 t_1 t_2 \cdots = 0110100110010110 \cdots,$$

The gray squares in the rows below depict the evolution of novel length-n factors in the Thue-Morse sequence for $1 \leq n \leq 9$.



Theorem 96. *Let* $\mathbf{x}$ *be an infinite word. For* $n \geq 1$*, the number of contiguous blocks of starting occurrences of novel factors in row* n *is at most* $\rho_{\mathbf{x}}(n) - \rho_{\mathbf{x}}(n-1) + 1$.

Proof. By induction on n. The base case is easy. Assume the claim is true for $n-1$. We prove it for n.

Every position marking the start of a novel occurrence is still novel. Further, in every block except the first, we get novel occurrences at one position to the left of the beginning of the block. So if row $n-1$ has t contiguous blocks, then we get $t-1$ novel occurrences at the beginning of each block, except the first.

The remaining $\rho_{\mathbf{x}}(n) - \rho_{\mathbf{x}}(n-1) - (t-1)$ novel occurrences could be, in the worst case, in their own individual contiguous blocks. Thus row n has at most $t + \rho_{\mathbf{x}}(n) - \rho_{\mathbf{x}}(n-1) - (t-1)$ $= \rho_{\mathbf{x}}(n) - \rho_{\mathbf{x}}(n-1) + 1$ contiguous blocks.

For the Thue-Morse example, it can be proved that $\rho_{\mathbf{t}}(n) - \rho_{\mathbf{t}}(n-1) \leq 4$. So the number of contiguous blocks of novel factors is at most 5. This is achieved, for example, for $n = 6$.

Corollary 97. *If the sequence* $\mathbf{x}$ *has linear complexity (that is,* $\rho_{\mathbf{x}}(n) = O(n)$*), then there is a constant* C *such that every row in the evolution of novel occurrences consists of at most* C *contiguous blocks.*

Proof. By a deep result of Cassaigne [17], we know that for every sequence of linear subword complexity (not just the k-automatic sequences), there exists a constant C such that $\rho_{\mathbf{x}}(n) - \rho_{\mathbf{x}}(n-1) \leq C - 1$. Hence from our result, there are at most C contiguous blocks in any row. □

Theorem 98. *Let* $\mathbf{x}$ *be a* k*-automatic sequence. Then its subword complexity function* $\rho_{\mathbf{x}}(n)$ *is* k*-synchronized.*

Proof. Construct a DFA to accept $\{(n, m)_k : n \geq 0 \text{ and } m = \rho_{\mathbf{x}}(n)\}$. There is a finite constant $C \geq 1$ such that the number of contiguous blocks of novel factors is bounded by C.

Nondeterministically "guess" the endpoints of every block and then verify that each factor of length n starting at the positions inside blocks is a novel occurrence, while all other factors are not.

Finally, verify that m is the sum of the sizes of the blocks. □

Example 99. Let us first show that the subword complexity function for Thue-Morse has the property that $\rho_{\mathbf{t}}(n+1) - \rho_{\mathbf{t}}(n) \leq 4$ for all n. To do this, we need the concept of "right special factor". A finite factor w of a binary sequence $\mathbf{x}$ is called *right special* if both $w0$ and $w1$ appear in x. Then it is easy to see that $\rho_{\mathbf{t}}(n+1) - \rho_{\mathbf{t}}(n)$ is the number of right special factors of length n.

To count these, we first make a predicate for the property that the factor of length n beginning at position i is right special:

$$\begin{aligned}\text{rtspec}(i,n) = (\exists j\ (T[j+n] = 0)\ \wedge\ \forall t\ (t < n)\ \Longrightarrow\ T[i+t] = T[j+t]) \\ \wedge\ (\exists k\ (T[k+n] = 1)\ \wedge\ \forall t\ (t < n)\ \Longrightarrow\ T[i+t] = T[k+t]).\end{aligned}$$

Next, we make a predicate for the property that the factor of length n beginning at position i is novel:

$$\text{nf}(i,n) := \forall j\ (j < i)\ \Longrightarrow\ \exists t\ (t < n)\ \wedge\ T[i+t] \neq T[j+t].$$

Next, we make a predicate for the property that the factor of length n beginning at position i is both novel and right special:

$$\text{nrt}(i,n) := \text{nf}(i,n)\ \wedge\ \text{rtspec}(i,n).$$

Finally, we make a predicate for the property that there exists some n with 5 distinct right-special factors of length n:

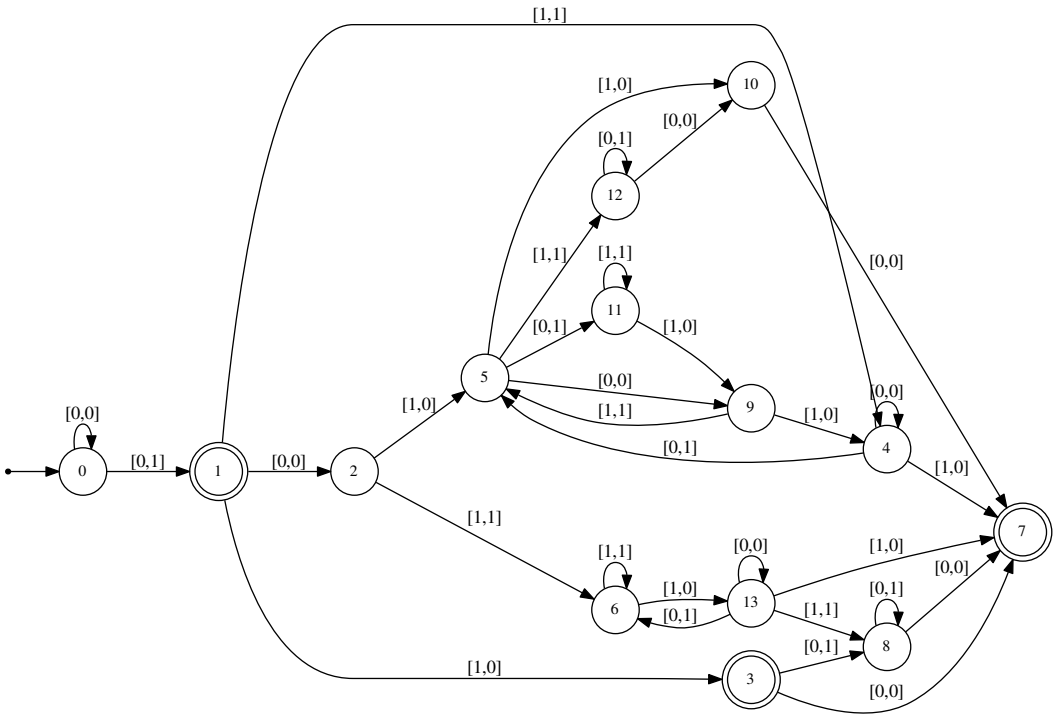
$$\begin{aligned}\text{tmspec5}(i,n) := \exists n, i_1, i_2, i_3, i_4, i_5\ (i_1 < i_2)\ \wedge\ (i_2 < i_3)\ \wedge\ (i_3 < i_4)\ \wedge\ (i_4 < i_5) \\ \wedge\ \text{nrt}(i_1,n)\ \wedge\ \text{nrt}(i_2,n)\ \wedge\ \text{nrt}(i_3,n)\ \wedge\ \text{nrt}(i_4,n)\ \wedge\ \text{nrt}(i_5,n).\end{aligned}$$

When we evaluate this with `Walnut`, it evaluates to false, so there are at most 4 special factors of every length.

So, from the arguments above we know that Thue-Morse novel factors are "clumped" into at most 5 intervals. Suppose these clumps are at indices in $[a_1, b_1), [a_2, b_2), \ldots, [a_5, b_5)$ for $0 = a_1 \leq b_1, a_2 \leq b_2, \ldots, a_5 \leq b_5$. (We allow $a_1 = b_1$, etc., because some of these 5 intervals could be empty.) Here is the predicate $\text{tmsc}(n,r)$ asserting that the subword complexity of Thue-Morse for words of length n is r:

$$\begin{aligned}&\text{tmsc}(n,r) := \exists a_2, a_3, a_4, a_5, b_1, b_2, b_3, b_4, b_5 \\ &(b_1 \leq a_2\ \wedge\ a_2 \leq b_2\ \wedge\ b_2 \leq a_3\ \wedge\ a_3 \leq b_3\ \wedge\ b_3 \leq a_4\ \wedge\ a_4 \leq b_4\ \wedge\ b_4 \leq a_5\ \wedge\ a_5 \leq b_5) \\ &\quad\wedge\ (\forall i\ (i < b_1)\ \Longrightarrow\ \text{nf}(i,n))\ \wedge\ (\forall i\ (a_2 \leq i\ \wedge\ i < b_2)\ \Longrightarrow\ \text{nf}(i,n)) \\ &\quad\wedge\ (\forall i\ (a_3 \leq i\ \wedge\ i < b_3)\ \Longrightarrow\ \text{nf}(i,n))\ \wedge\ (\forall i\ (a_4 \leq i\ \wedge\ i < b_4)\ \Longrightarrow\ \text{nf}(i,n)) \\ &\quad\wedge\ (\forall i\ (a_5 \leq i\ \wedge\ i < b_5)\ \Longrightarrow\ \text{nf}(i,n))\ \wedge\ (\forall i\ (i \geq b_1\ \wedge\ i < a_2)\ \Longrightarrow\ \neg\,\text{nf}(i,n)) \\ &\quad\wedge\ (\forall i\ (i \geq b_2\ \wedge\ i < a_3)\ \Longrightarrow\ \neg\,\text{nf}(i,n))\ \wedge\ (\forall i\ (i \geq b_3\ \wedge\ i < a_4)\ \Longrightarrow\ \neg\,\text{nf}(i,n)) \\ &\quad\wedge\ (\forall i\ (i \geq b_4\ \wedge\ i < a_5)\ \Longrightarrow\ \neg\,\text{nf}(i,n))\ \wedge\ (\forall i\ (i \geq b_5)\ \Longrightarrow\ \neg\,\text{nf}(i,n))\ \wedge \\ &\qquad\qquad r = b_1 + (b_2 - a_2) + (b_3 - a_3) + (b_4 - a_4) + (b_5 - a_5).\end{aligned}$$

This predicate asserts first that all indices inside the intervals correspond to the starting points of novel factors, and second that all indices outside the intervals do not correspond to the starting points of to novel factors. When we evaluate this predicate in `Walnut` (warning: it needs at least 16 gigs of memory and runs for 100 seconds on a laptop), we get the automaton below.



Corollary 100. *Given a k-automatic sequence $\mathbf{x}$, there is an algorithm that, on input n in base k, will compute the subword complexity $\rho_{\mathbf{x}}(n)$ expressed in base k in time $O(\log n)$.*

12.7 Many aspects of k-automatic sequences are k-synchronized

$A_{\mathbf{x}}(n)$ = length of shortest prefix of $\mathbf{x}$ containing all length-n factors of $\mathbf{x}$

= the smallest integer t such that every length-n factor of $\mathbf{x}$ occurs at least once in $\mathbf{x}[0..t-1]$.

= t such that every length-n factor of $\mathbf{x}$ occurs in $\mathbf{x}[0..t-1]$ but the length-n factor ending at position $t-1$ occurs exactly once in $\mathbf{x}[0..t-1]$

$$L = \{(n,t)_k \ : \ \forall\, i \geq 0 \ \exists\, j \leq t-n \\ \text{such that } \mathbf{x}[i..i+n-1] = \mathbf{x}[j..j+n-1] \\ \text{and } \forall\, \ell < t-n \\ \mathbf{x}[\ell..\ell+n-1] \neq \mathbf{x}[t-n..t-1]\}.$$

12.8 Other synchronized functions

- **separator function**: length of the shortest factor of $\mathbf{x}$ beginning at position n that never appeared previously in $\mathbf{x}$ (Carpi & Maggi, 2001)

- **repetitivity index**: the minimal distance between two consecutive occurrences of the same length-n factor in $\mathbf{x}$ (Carpi & D'Alonzo, 2009)

- **recurrence function**: size of the smallest "window" always guaranteed to contain all length-n factors in $\mathbf{x}$ (Charlier & Rampersad & S, 2011)

12.9 Applications: an improvement on Goldstein

Corollary. There is an algorithm, that, given a k-automatic sequence $\mathbf{x}$, will compute

- $\sup_{n\geq 1} \rho_{\mathbf{x}}(n)/n$,
- $\limsup_{n\geq 1} \rho_{\mathbf{x}}(n)/n$,
- $\inf_{n\geq 1} \rho_{\mathbf{x}}(n)/n$, and
- $\liminf_{n\geq 1} \rho_{\mathbf{x}}(n)/n$.

Proof. We already showed how to construct an automaton accepting $\{(n, \rho_{\mathbf{x}}(n))_k \ : \ n \geq 1\}$. Using Schaeffer & S (2012), we can compute the sup, lim sup etc.

Theorem. If $\mathbf{x}$ is k-automatic, then the following are k-synchronized:

- the function counting the number of distinct length-n factors that are powers;
- the function counting the number of distinct length-n factors that are primitive words.

Sketch of proof: Main ideas:

- A word x is a power if and only if there exist nonempty words y, z such that $x = yz = zy$.

- Thus, we can express the formula $P(i,j) :=$ "$\mathbf{x}[i..j]$ is a power" as follows: "there exists d, $0 < d < j-i+1$, such that $\mathbf{x}[i..j-d] = \mathbf{x}[i+d..j]$ and $\mathbf{x}[j-d+1..j] = \mathbf{x}[i..i+d-1]$".

- Furthermore, we can express the formula $P'(i,n) :=$ "$\mathbf{x}[i..i+n-1]$ is a length-n power and is a novel occurrence of that factor in $\mathbf{x}$".

- We show that once again the novel occurrences of length-n powers are clustered into a finite number of blocks.

Sketch of proof

- Then we can nondeterministically guess the endpoints of these blocks, and verify that the length-n factors beginning at the positions inside the blocks are novel occurrences of powers, while those outside are not, and sum the lengths of the blocks, using a finite automaton built from M.

- So the counting function for powers is k-synchronized.

- The number of length-n primitive words in $\mathbf{x}$ is then also k-synchronized, since it is expressible as the total number of words of length n, minus the number of length-n powers.

12.10 Unsynchronized sequences

Are other aspects of k-automatic sequences always k-synchronized?

No.

Recall that a word w is *bordered* if it has a nonempty prefix, other than w itself, that is also a suffix. Alternatively, w is bordered if it can be written in the form $w = tvt$, where t is nonempty. Otherwise a word is *unbordered*.

Theorem 101. *The characteristic sequence of the powers of 2* $\mathbf{c} = 0110100010\cdots$ *is 2-automatic, but the function* $u_{\mathbf{c}}(n)$ *counting the number of unbordered factors is not 2-synchronized.*

Proof. It is not hard to verify that $\mathbf{c}$ is 2-automatic and that $\mathbf{c}$ has exactly $r+2$ unbordered factors of length 2^r+1, for $r \geq 2$ — namely, the factors beginning at positions 2^i for $0 \leq i \leq r-1$, and the factors beginning at positions 2^{r+1} and $3 \cdot 2^r$. However, if $u_{\mathbf{c}}(n)$ were 2-synchronized, then reading an input where the first component looks like $0^i 10^{r-1} 1$ (and hence a representation of 2^r+1) for large r would force the transitions to enter a cycle. If the transitions in or before the cycle contained a nonzero entry in the second component, this would force $u_{\mathbf{c}}(n)$ to grow linearly with n when n is of the form 2^r+1. Otherwise, the corresponding transitions for the second component are just 0's, in which case $u_{\mathbf{c}}(n)$ is bounded above by a constant, for n of the form 2^r+1. Both cases lead to a contradiction. □

12.11 Notes

The material in Section 12.6 is from [30]. In [39], the authors showed that the number of length-n palindromic factors is not always synchronized.

12.12 Exercises

1. An obvious generalization of the notion of k-synchronization is (k, ℓ)-synchronization: a function f is (k, ℓ)-synchronized if there is a DFA recognizing the language $\{(n, f(n))_{k,\ell} : n \geq 0\}$, where $(a, b)_{k,\ell} \in (\Sigma_k \times \Sigma_\ell)^*$ is the base-k and base-ℓ representations of a and b, respectively, padded with zeroes on the left if necessary.

   (a) Show that if f is (k, ℓ)-synchronized and g is (ℓ, m)-synchronized, then $g \circ f$ is (k, m)-synchronized.

   (b) Show that the function $f(n) = n^2$ is not (k, k^2)-synchronized for any $k \geq 2$.

   (c) State and prove the analogous results for Theorem 91 for (k, ℓ)-synchronized functions.

□

Chapter 13

The k-regular sequences

Although the k-automatic sequences form a large and interesting class, one drawback is that they need to take their values in a finite set. But many interesting sequences, such as $(s_2(n))_{n\geq 0}$ (counting the sum of the bits in the base-2 representation of n), take their values in $\mathbb{N}$ (or $\mathbb{Z}$, or any ring). We would like to find a generalization that allows this.

In this section we discuss this generalization of the k-automatic sequences, called the k-regular sequences. There are two different ways to view these sequences, which turn out to be identical.

The first way is to generalize the k-kernel. Instead of demanding that the k-kernel be finite, we require that there exists a finite set S of sequences such that each sequence in the k-kernel is expressible as a linear combination of the sequences in S.

The second way is to generalize the notion of automaton. We can view an NFA M as follows: each input letter a induces a map on the states, which can be represented by a Boolean matrix M_a: we have $M_a[i,j] = 1$ if and only if $q_j \in \delta(q_i, a)$. Define the matrix-valued morphism $\mu(a) = M_a$. Then an easy induction on $|x|$ gives that $\mu(x)[i,j] = 1$ for words x iff $q_j \in \delta(q_i, x)$. (Here in the multiplication of vectors and matrices we use *Boolean matrix multiplication*, where $\wedge$ (AND) replaces scalar multiplication and $\vee$ (OR) replaces scalar addition.) Hence, if we define $u = [1\ 0\ 0 \cdots\ 0]$ and v to be the column vector with 1's corresponding to the final states and 0's elsewhere, then $v\mu(x)w = 1$ iff $x \in L(M)$.

To generalize this to the k-regular sequences, we replace Boolean matrix multiplication with ordinary matrix multiplication. A sequence $\mathbf{a} = (a(n))_{n\geq 0}$ is k-regular if there exist vectors v, w and a matrix-valued morphism μ such that $a(n) = v\mu((n)_k)w$ for all $n \geq 0$. The triple (v, μ, w) is called a *linear representation* for $\mathbf{a}$. (Of course, it is also possible to do this for the reverse representation $(n)_k^R$ by transposing all the matrices and interchanging the roles of v and w. This corresponds to an lsd-first reading of the base-k representation of n.) Linear representations are not unique.

Example 102. Let us work this all out for $s_2(n)$, the sum of the bits in the base-2 representation of n. Clearly

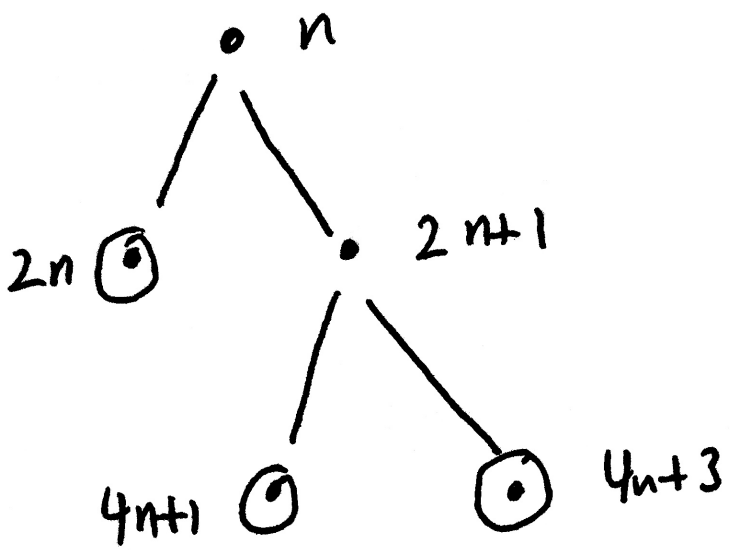
$$s_2(2n) = s_2(n)$$
$$s_2(2n+1) = s_2(n) + 1,$$

and therefore $s_2(k^e \cdot n + i) = s_2(n) + s_2(i)$ for $e \geq 0$, $0 \leq i < k^e$. Thus every element of the k-kernel of $(s_2(n))_{n\geq 0}$ can be written as a $\mathbb{Z}$-linear combination of the sequences $(s_2(n))_{n\geq 0}$ and the constant sequence 1.

We can also find a representation for $s_2(n)$ in terms of "smaller" sequences in the k-kernel only:

$$\begin{aligned} s_2(2n) &= s_2(n) \\ s_2(4n+1) &= s_2(2n+1) \\ s_2(4n+3) &= -s_2(n) + 2s_2(2n+1) \end{aligned}$$

To know we have "enough" relations we draw a k-ary tree and consider a node to be a leaf if there is an expression for the corresponding sequence in the k-kernel in terms of "smaller sequences". For example, for the system above we have



To find a linear representation for $(s_2(n))_{n\geq 0}$, we write down the basis elements $(s_2(n))$ and the constant sequence 1 and consider the effect of *right* multiplication by matrices M_0 and M_1. We want M_0 and M_1 to act as follows:

$$\begin{aligned} [s_2(n) \quad 1] \cdot M_0 &= [s_2(2n) \quad 1] \\ [s_2(n) \quad 1] \cdot M_1 &= [s_2(2n+1) \quad 1] \end{aligned}$$

and so we can choose

$$\begin{aligned} v &= \begin{bmatrix} 0 & 1 \end{bmatrix} \\ M_0 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ M_1 &= \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \\ w &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \end{aligned}$$

The choice for u corresponds to setting $n = 0$, and the choice for v corresponds to picking out the particular sequence we are interested in.

Alternatively, we can find a representation from the sequences of the k-kernel. Here the basis elements are

$$(s_2(n))_{n\geq 0} \text{ and } (s_2(2n+1))_{n\geq 0}.$$

We want

$$\begin{aligned} [s_2(n) \quad s_2(2n+1)] \cdot M_0 &= [s_2(2n) \quad s_2(4n+2)] \\ [s_2(n) \quad s_2(2n+1)] \cdot M_1 &= [s_2(2n+1) \quad s_2(4n+3)] \end{aligned}$$

We can choose

$$\begin{aligned} v &= \begin{bmatrix} 0 & 1 \end{bmatrix} \\ M_0 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ M_1 &= \begin{bmatrix} 0 & -1 \\ 1 & 2 \end{bmatrix} \\ w &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \end{aligned}$$

This example shows that if a sequence takes non-negative values, the entries of the matrices in the linear representation could be negative.

One fine technical point is that it is always possible to choose M_0 in such a way that $vM_0 = v$. This is analogous to the condition $\delta(q_0, 0) = q_0$ that we often require for DFAO's in automatic sequences.

13.1 Closure properties of k-regular sequences

Theorem 103. *Let* $\mathbf{s} = (s(n))_{n\geq 0}$ *and* $\mathbf{t} = (t(n))_{n\geq 0}$ *be* k*-regular sequences. Then*

(a) $\mathbf{s} + \mathbf{t} = (s(n) + t(n))_{n\geq 0}$

(b) $\mathbf{st} = (s(n)t(n))_{n\geq 0}$

(c) $c\mathbf{s} = (c(s(n))_{n\geq 0}$

are all k-regular sequences.

Proof. Let the k-kernel of $\mathbf{s}$ be contained in $\langle \mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_r \rangle$ and let the k-kernel of $\mathbf{t}$ be contained in $\langle \mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{r'} \rangle$.

(a) The k-kernel of $\mathbf{s} + \mathbf{t}$ is evidently contained in $\langle \mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_r, \mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{r'} \rangle$.

(b) The k-kernel of $\mathbf{st}$ is evidently contained in $\langle A \rangle$, where $A = \{\mathbf{s}_i \mathbf{t}_j \,:\, 1 \leq i \leq r, 1 \leq j \leq r'\}$.

(c) Similarly, the k-kernel of $c\mathbf{s}$ is contained in $\langle c\mathbf{s}_1, c\mathbf{s}_2, \ldots, c\mathbf{s}_r \rangle$.

□

However, the class of k-regular sequences is not closed under some operations that you might suspect it would be. For example, it is not closed under $\mathbf{s}/\mathbf{t} = (s(n)/t(n))_{n\geq 0}$. Take $\mathbf{s} = 1$, and $t(0) = 1$, $t(2n) = n+1$, $t(2n+1) = t(n)+1$. Exercise 1 shows that $\mathbf{t}$ is 2-regular.

For $j \geq 1$ define $t_j(n) = t(2^j \cdot n + 2^{j-1} - 1)$; each $(t_j(n))_{n\geq 0}$ is an element of the 2-kernel. Check that $t_j(n) = n + j$ for $j \geq 1$. Suppose $1/\mathbf{t} = (1/t(n))_{n\geq 0}$ were 2-regular. Then the module generated by

$$(1/t_1(n))_{n\geq 0}, (1/t_2(n))_{n\geq 0}, \ldots,$$

would have finite rank. Then for some $m \geq 1$, the rows of the $m \times m$ matrix $M = M_{i,j}$ defined by

$$M_{i,j} = \frac{1}{t_j(i-1)} = \frac{1}{i+j-1}$$

for $1 \leq i, j \leq m$ would be linearly independent and hence $\det M = 0$. But M is an $m \times m$ Hilbert matrix, well-known to have determinant

$$\frac{1}{\pm \prod_{1\leq k<m} (2k+1) \binom{2k}{k}^2} \neq 0,$$

a contradiction.

Similarly, the class of k-regular sequences is not closed under absolute value. Define $e_i(n)$ for $i \in \{0, 1\}$ to be the number of occurrences of the digit i in the base-2 expansion of n. Define $f(n) = e_0(n) - e_1(n)$. Exercise 2 asks you to show that $f(n)$ is 2-regular.

Now $(f(2^j \cdot n))_{n\geq 0}$ is a sequence in the 2-kernel of $(f(n))_{n\geq 0}$, and we have

$$|f(2^j \cdot n)| = |e_0(n) - e_(n) + j| \text{ for } n \geq 1,\ j \geq 0\ .$$

Assume (to get a contradiction) that there is a finite linear combination, with not all $c_i = 0$, such that $\sum_{0\leq i\leq b} c_i f(2^i \cdot n) = 0$ for all n. Choose the least i such that $c_i \neq 0$, and call it a. Then

$$f(2^a \cdot n) = \sum_{a+1\leq i\leq b} -(c_i/c_a) f(2^i n) \tag{13.1}$$

for all $n \geq 0$. Let x_m be the least nonzero integer such that $e_0(x_m) - e_(x_m) = 0$. It is easy to see that $x_m = 2^{-m} - 1$ for $m < 0$ and $x_m = 2^{m+1}$ for $m \geq 0$. Evaluate Eq. (13.1) at $n = x_m$ for $m \in \mathbb{Z}$. On the left-hand side of Eq. (13.1) we have

$$|f(2^a \cdot x_m)| = |e_0(x_m) - e_1(x_m) + a| = |m + a|.$$

On the right-hand side we have

$$\sum_{a+1 \leq i \leq b} -(c_i/c_a)|m + i|.$$

So

$$|m + a| = \sum_{a+1 \leq i \leq b} -(c_i/c_a)|m + i|$$

for all m. But for $m \geq -(a+1)$ the right-hand side of the form $Am + B$ for constants A, B, and hence is monotone. But the left-hand side is 1 for $m = -(a+1)$, 0 for $m = -a$, and 1 for $m = 1 - a$, which is not monotone, a contradiction.

13.2 k-regular sequences and k-automatic sequences

Theorem 104. *Suppose $\mathbf{a} = (a(n))_{n \geq 0}$ is a k-regular sequence over $\mathbb{Z}$ that takes only finitely many values. Then $(a(n))_{n \geq 0}$ is k-automatic.*

Proof. If $\mathbf{a}$ is k-regular, let its k-kernel be generated by the finitely many sequences $\mathbf{a}_i := (a_i(n))_{n \geq 0}$ of the k-kernel, for $1 \leq i \leq r$. This means that if $a(n) = \mathbf{a}_1(n)$ and

$$V(n) = \begin{bmatrix} a_1(n) \\ a_2(n) \\ \vdots \\ a_r(n) \end{bmatrix},$$

then $V(kn + a) = \mu(a) \cdot V(n)$ for $0 \leq a < k$, and $n \geq 0$, and the appropriate matrix $\mu(a)$. Without loss of generality we can assume $\mu(0) \cdot V(0) = V(0)$.

Since $\mathbf{a}$ takes only finitely many values, the same is true of $(V(n))_{n \geq 0}$. Call the set of these values S. For $x \in S$ define the k-uniform morphism

$$\sigma(x) = (\mu(0) \cdot x)(\mu(1) \cdot x) \cdots (\mu(k-1) \cdot x).$$

Then the infinite word

$$\alpha = V(0)V(1)V(2) \cdots$$

is the fixed point of μ and $(a(n))_{n \geq 0}$ is given by an image of α under the coding that maps each vector to its first component. □

Example 105. Consider the 3-regular sequence $(a(n))_{n\geq 0}$ defined by

$$\begin{aligned} a(n) &= n \text{ for } n = 0, 1, 2 \\ a(3n) &= a(n) \\ a(9n+1) &= a(3n+1) \\ a(9n+2) &= a(3n+2) \\ a(9n+4) &= a(3n+2) \\ a(9n+5) &= -a(n) \\ a(9n+7) &= 0 \\ a(9n+8) &= -a(n) + a(3n+2) \end{aligned}$$

Then Exercise 3 asks you to prove that $(a(n))_{n\geq 0}$ takes the values $\{-3, -2, -1, 0, 1, 2, 3\}$ only (and 3 occurs for the first time at $a(401)$).

It follows that $(a(n))_{n\geq 0}$ is 3-automatic. In fact, it is generated by a 3-DFAO with 56 states. We will how to find it in the next lecture.

13.3 Transformation of k-regular sequences

Lemma 106. *Let $(f(n))_{n\geq 0}$ be a k-regular sequence, and let $\Sigma_k = \{0, 1, \ldots, k-1\}$. Let $T = (Q, \Sigma_k, \Sigma_k, \delta, q_0, \rho)$ be a deterministic finite-state transducer with transitions on single letters only, but allowing arbitrary words as outputs on each transition. More precisely,*

- *$Q = \{q_0, \ldots, q_{r-1}\}$;*
- *$\delta : Q \times \Sigma_k \to Q$ is the transition function; and*
- *$\rho : Q \times \Sigma_k \to \Sigma_k^*$ is the output function.*

Let the domain of δ and ρ be extended to Σ_k^* in the obvious way. Define $g(n) = f(T((n)_k))$. Then $(g(n))_{n\geq 0}$ is also a k-regular sequence.

Proof. Let (v, μ, w) be a rank-s linear representation for f. We create a linear representation (v', μ', w') for g.

The idea is that $\mu'(a)$, $0 \leq a < k$, is an $n \times n$ matrix, where $n = rs$. It is easiest to think of $\mu'(a)$ as an $r \times r$ matrix, where each entry is itself an $s \times s$ matrix. In this interpretation, $(\mu'(a))_{i,j} = \mu(\rho(q_i, a))$ if $\delta(q_i, a) = q_j$.

An easy induction now shows that if $\delta(q_i, x) = q_j$ and $\rho(q_i, x) = y$, then $(\mu'(x))_{i,j} = \mu(y)$. If we now let v' be the vector $[v \quad 0 \quad \cdots \quad 0]$ and w' be the vector $[w \quad w \quad \cdots \quad w]^T$, then it follows that $v'\mu'(x)w' = v\mu(T(x))w$. This gives a linear representation for $(g(n))_{n\geq 0}$. □

Corollary 107. *If $(f(n))_{n\geq 0}$ is k-regular, then so are the sequences*

(a) $(f(an+b))_{n\geq 0}$ for $a, b \in \mathbb{N}$;

(b) $(f(\lfloor n/a \rfloor))_{n\geq 0}$ *for* $a \in \mathbb{N}$, $a \geq 1$.

Proof. First, we build a finite-state transducer T that outputs the base-k representation of $\lfloor n/a \rfloor$ on input $(n)_k$. The idea is just to use long division, keeping track of the carries (which can be at most a) in the state. A slight complication is to avoid outputting leading zeroes, but this is easily handled (see example for $a = 3$, $k = 2$).

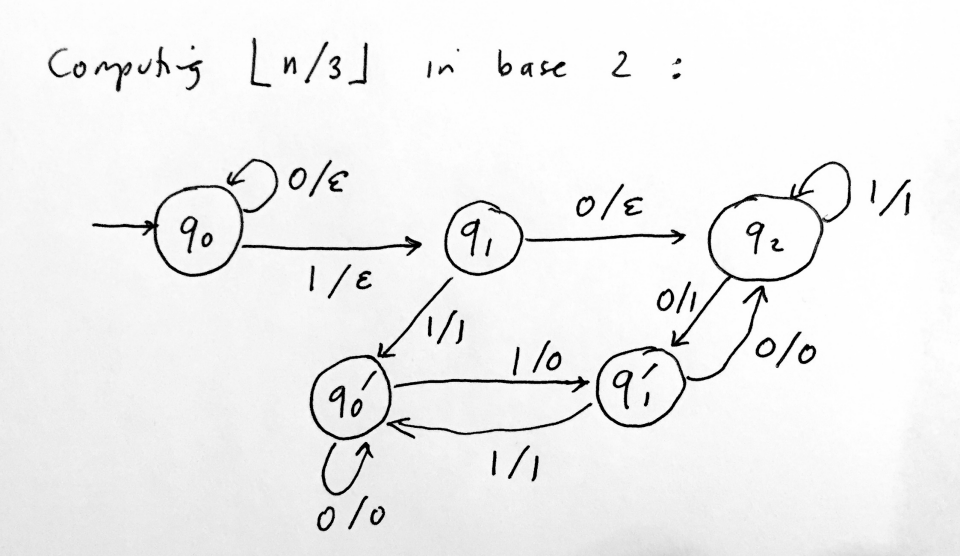


Figure 13.1: Transducer dividing by 3

Next, we use the lemma above to see that $(f(T((n)_k)))_{n\geq 0}$ is k-regular. Thus we have shown that $(f(\lfloor n/a \rfloor))_{n\geq 0}$ is k-regular.

Now consider the periodic sequences $(p_i(n))_{n\geq 0}$ defined by $p_i(n) = 1$ if $n \equiv i \pmod{a}$ and 0 otherwise. Each such sequence is k-automatic and hence k-regular. Let $f_i(n)$ be k-regular sequences for $0 \leq i < a$. By above each sequence $(f_i(\lfloor n/a \rfloor))_{n\geq 0}$ is k-regular. Hence $f(n)$, the a-way merge of the sequence $f_i(n)$, is given by

$$f(n) := \sum_{0\leq i<a} p_i(n) f_i(\lfloor n/a \rfloor),$$

and is k-regular by the closure properties of these sequences. □

13.4 More examples of k-regular sequences

Theorem 108. *Let $p(X)$ be a polynomial with integer coefficients. Then $(p(n))_{n\geq 0}$ is k-regular for all $k \geq 2$.*

Proof. Let $p(x) = \sum_{0 \le i \le t} a_i x^i$. We have

$$\begin{aligned} p(k^e \cdot n + j) &= \sum_{0 \le i \le t} a_i (k^e \cdot n + j)^i \\ &\in \langle 1, n, n^2, \ldots, n^i \rangle, \end{aligned}$$

where we have used the binomial theorem. □

Theorem 109. *Let $0 \le a < k$ be an integer. Define $e_a(n)$ to be the number of occurrences of a in the canonical base-k representation of n. (Note that $e_0(0) = 0$ because $(n)_k = \epsilon$.) Then each $(e_a(n))_{n \ge 0}$ is k-regular.*

Proof. It is easy to see that $e_a(n)$ satisfies the relation

$$e_a(kn + b) = \begin{cases} e_a(n), & \text{if } a \neq b; \\ e_a(n) + 1, & \text{if } a = b \text{ and } n > 0; \\ 0, & \text{if } a = b = n = 0. \end{cases}$$

The case $a = b = n = 0$ needs to be treated specially because, for example, if $k = 2$ then $e_0(2n) = e_0(n) = 0$ for $n = 0$.

So every sequence in the k-kernel of $(e_a(n))_{n \ge 0}$ is a linear combination of the three sequences $(e_a(n))_{n \ge 0}$, the constant sequence 1, and the sequence that is 1 at $n = 0$ and 0 otherwise. □

Corollary 110. *Let $a < k$, and let $p(x_1, x_2, \ldots, x_a)$ be a multivariate polynomial in a indeterminates with rational coefficients. Then*

$$(p(|(n)_k|_1, |(n)_k|_2, \ldots, |(n)_k|_a))_{n \ge 0}$$

is a k-regular sequence.

Proof. Apply the previous theorem and the closure properties of k-regular sequences. □

We can now show that a fundamental decision problem about k-regular sequences over $\mathbb{Z}$ is, in general, undecidable. Recall Hilbert's tenth problem: it is the problem to decide, given a multi-variate polynomial $p(x_1, x_2, \ldots, x_a)$ with integer coefficients, whether there exist natural numbers $c_1, c_2, \ldots, c_a$ such that $p(c_1, c_2, \ldots, c_a) = 0$. A classic result, due to Putnam-Robinson-Matiyasevich, is that this problem is recursively unsolvable (undecidable).

Theorem 111. *Given a k-regular sequence $(f(n))_{n \ge 0}$, it is undecidable (recursively unsolvable) to determine if there exists $n_0 \in \mathbb{N}$ such that $f(n_0) = 0$.*

Proof. We reduce from Hilbert's tenth problem. Given the polynomial $p(x_1, x_2, \ldots, x_a)$, transform it to the k-regular sequence

$$(p(|(n)_k|_1, |(n)_k|_2, \ldots, |(n)_k|_a))_{n \ge 0}$$

where $k = a + 1$. Then $f(n_0) = 0$ for some n_0 iff

$$(p(|(n_0)_k|_1, |(n_0)_k|_2, \ldots, |(n_0)_k|_a)) = 0$$

for some n_0 iff there exist

$$c_1 = |(n_0)_k|_1, c_2 = |(n_0)_k|_2, \ldots, c_a = |(n_0)_k|_a$$

such that $p(c_1, c_2, \ldots, c_a) = 0$. So an algorithm to detect the presence of 0's in a k-automatic sequence would allow us to solve Hilbert's tenth problem. □

13.5 Growth rate of k-regular sequences

In this section we prove a theorem about the growth rate of k-regular sequences. We can use it to show, for example, that $(2^n)_{n\geq 0}$ is not k-regular for any k.

Theorem 112. *Suppose* $\mathbf{f} = (f(n))_{n\geq 0}$ *is a* k*-regular sequence. Then there exists a real number* $\alpha > 0$ *such that* $f(n) = O(n^\alpha)$.

Proof. Since $\mathbf{f}$ is k-regular, there exists a linear representation (v, μ, w) such that

$$f(n) = v\mu((n)_k)w = v\mu(a_1)\mu(a_2)\cdots\mu(a_r)w,$$

where $(n)_k = a_1 a_2 \cdots a_r$.

Let $||\cdot||$ denote the L_∞ norm of a vector, extended to matrices in the usual way. Then

$$\begin{aligned}
|f(n)| &\leq ||v||\ ||\mu(a_1)||\ ||\mu(a_2)||\ \cdots\ ||\mu(a_r)||\ ||w||, \\
&\leq ||v||\ m^r\ ||w|| \\
&\leq cm^{1+\log_k n} \\
&= (cm)m^{\log_k n} \\
&= (cm)n^{\log_k m} \\
&= Cn^\alpha,
\end{aligned}$$

where $m = \max_{0\leq i<k} ||\mu(i)||$ and $C = cm$ and $\alpha = \log_k m$. □

Another result, which we will not prove, is that if $(f(n))_{n\geq 0}$ is unbounded and k-regular, then there exists a constant $c > 0$ such that $|f(n)| > c \log n$ infinitely often. See [8].

13.6 Notes

The two principal papers about k-regular sequences are [5] and [7]. Connections with rational series can be found in [9].

13.7 Exercises

1. Show that if $t(0) = 1$, $t(2n) = n + 1$, $t(2n + 1) = t(n) + 1$, then $(t(n))_{n\geq 0}$ is a 2-regular sequence.

2. Define $e_i(n)$ for $i \in \{0, 1\}$ to be the number of occurrences of the digit i in the base-2 expansion of n. Define $f(n) = e_0(n) - e_1(n)$. Show that $(f(n)_{n\geq 0}$ is a 2-regular sequence.

3. Prove that $(a(n))_{n\geq 0}$ from Example 105 takes the values $\{-3, -2, -1, 0, 1, 2, 3\}$ only (and 3 occurs for the first time at $a(401)$).

4. Say a sequence (s_n) is "Fibonacci-regular" if there is a linear representation for it in Fibonacci representation. That is, there are vectors v, w and a matrix-valued morphism μ such that if w is the Fibonacci representation of n, then $s_n = v\mu(w)w^T$.

   Consider the power series

   $$F(x) = \prod_{n\geq 2}(1 - x^{F_n}) = (1 - x)(1 - x^2)(1 - x^3)\cdots = \sum_{i\geq 0} e_i x^i,$$

   where F_n is the n'th Fibonacci number, with $F_0 = 0$, and $F_1 = 1$. Show that the sequence $(e_i)_{i\geq 0}$ is Fibonacci-regular. Then use the "semigroup trick" to show that $e_i \in \{-1, 0, 1\}$ for all i.

5. Is the class of k-regular sequences closed under composition? That is, if $(f(n))_{n\geq 0}$ and $(g(n))_{n\geq 0}$ are both k-regular sequences taking values in $\mathbb{N}$, must $(f(g(n)))_{n\geq 0}$ be k-regular?

6. Define $h(n)$ to be the length of the longest block of contiguous 1's in the binary expansion of n.

   (a) Show that $h(2n) = h(n)$ and $h(2n + 1) = \max(h(n), \nu_2(n + 1) + 1)$ for $n \geq 0$.

   (b) Show that h is not 2-regular.

7. *Per Nørgård's sequence* $s(n)$ is defined as follows: $s(0) = 0$, and $s(2n) = -s(n)$ and $s(2n + 1) = s(n) + 1$ for $n \geq 0$. (He used it as a basis in some of his musical compositions.)

Chapter 14

Enumeration and k-regular sequences

In this lecture, we will establish a fundamental connection between automatic sequences and k-regular sequences that will allow us to enumerate many aspects of automatic sequences. Here by "enumerate" we mean "give an efficiently computable formula for" the particular quantity.

Many quantities dealing with k-automatic sequences are k-regular. Many quantities dealing with automatic sequences can be "automatically" enumerated; That is, we can algorithmically construct a polynomial-time algorithm to enumerate the quantity, given a first-order formula describing it. We can add this to our "combinatorial arsenal" of techniques, along with more traditional enumeration decision methods (Wilf, Gosper, Zeilberger, etc.)

14.1 What is a formula?

What is a formula? The traditional but vague answer is, an expression involving traditional operation such as addition, subtraction, multiplication, division, exponentiation, and perhaps additional functions such as factorial, binomial coefficient, trig and inverse trig functions, n'th roots, logarithm, floor, ceiling, summation, product, special functions, and so forth.

A more modern and precise answer is that an enumeration formula is an *algorithm* that runs in little-o of the time required to actually list the things being enumerated. A *good* formula is one that runs in time polynomial in n and the output size. A *very good* formula is one that runs in time polynomial in $\log n$ and the output size. See [50].

14.2 Enumerating aspects of automatic sequences

Many papers in the literature are concerned with enumerating various aspects of automatic sequences.

For example: *subword complexity* $\rho(n)$, the number of distinct factors of a sequence of length n. A classic result of Cobham: for automatic sequences $\rho(n) = O(n)$.

For morphic sequences, by contrast, it is possible for the subword complexity to be as high as $\Omega(n^2)$: for example, the fixed point, starting with a, of the morphism $a \to ab$, $b \to bc$,

and $c \to c$:

$$abbcbccbcccbccccbccccbccccccbcccccccbccccccccbcccccccccc\cdots$$

Palindrome complexity: the number of distinct factors of length n that are palindromes. (See Allouche, Baake, et al., 2003)

Unbordered complexity: the number of distinct factors of length n that are unbordered. (See Currie and Saari 2009)

Reversal complexity: the number of distinct factors of length n whose reversals are also factors.

Conjugate complexity: the number of distinct factors of length n whose conjugates are also factors.

Squares: the number of distinct factors of length $2n$ that are squares of order n. (Order of square xx is $|x|$.)

We will see that if P is a property of the factors of a k-automatic sequence that is expressible in a first-order formula, then the number $f_P(n)$ of such length-n factors is k-regular. This means there is a linear representation, in terms of matrices and vectors for computing $f_P(n)$. This always gives an algorithm A_P to compute $f_P(n)$ that runs in $O(\log n)$ time: a very good formula! However, it's not all good news. Given P, *finding* the algorithm A_P might take a huge amount of time (depending on P), and the constant factor in the $O(\log n)$-time algorithm might be ridiculously large.

14.3 Nondeterministic automata and path cardinalities

Our results are based on the following.

Theorem 113. *Given an NFA $M = (Q, \Sigma, \delta, q_0, F)$ define a matrix-valued morphism*

$$\mu(a)_{i,j} = \text{number of paths labeled } a \text{ from } q_i \text{ to } q_j.$$

Then for words x the quantity $\mu(x)_{i,j}$ is the number of paths labeled x from q_i to q_j.

Proof. By induction on the length of the path. □

Corollary 114. *Let $v = [1\ 0\ 0\ \cdots\ 0]$, where there is 1 in the position of the start state q_0, and w^T a boolean vector with 1's in positions of the final states. Then $v\mu(x)w^T > 0$ if and only if x is accepted by M.*

k-regular sequences and their connections to automata give a framework for enumerating these aspects of automatic sequences.

Basic idea:

Theorem 115. *Let $S \subseteq \mathbb{N} \times \mathbb{N}$. Given a DFA accepting the language*

$$L = \{(i,n)_k \ : \ (i,n) \in S\},$$

the function

$$f_S(n) = |\{i \ : \ (i,n) \in S\}|$$

is k-regular.

Proof. From the DFA M accepting L, make an NFA M' by projecting each label on a transition to its second coordinate. Thus, for example, transitions labeled $[0,1]$ and $[1,1]$ from q_i to q_j get projected to two arrows labeled 1 from q_i to q_j.

Now use the theorem about NFA's. □

14.4 An example: counting palindromic factors

How do we count, for example, the palindromic factors of length n — call it $f(n)$ — of an automatic sequence such as $\mathbf{t}$?

An obvious first try is to consider the language

$$L_{\text{pal}} = \{(i,n)_2 \ : \ S[i..i+n-1] \text{ is a palindrome } \}.$$

However, this doesn't work because each n can have many different i associated with it, and we are double- (or triple- or more) counting the same palindrome many times.

What we need to do is count a single i for each distinct palindrome. The easiest way to do this is to count, not occurrences of palindromes, but *first occurrences* of palindromes in the sequence.

Thus what we *really* want is

$$L_{\text{pal}} = \{(i,n)_2 \ : \ S[i..i+n-1] \text{ is a palindrome and any occurrence of the same factor } S[j..j+n-1] \text{ has } j \geq i\}$$

We can do this with the first-order formula

$$(\forall l \ (l < n) \implies S[i+l] = S[(i+n)-(l+1)]) \wedge (\forall j \ (\forall m \ (m < n) \implies S[i+m] = S[j+m]) \implies (j \geq i)).$$

In `Walnut`, for the Thue-Morse sequence, this is:

```
eval tmpalc "(Al (l<n) => T[i+l] = T[(i+n)-(l+1)]) &
   (Aj (Am (m<n) => T[i+m] = T[j+m]) => (j>=i))":
```

`Walnut` allows one to obtain the matrices (in Maple format) corresponding to a formula. The syntax is

```
eval tmpalc n "(Al (l<n) => T[i+l] = T[(i+n)-(l+1)]) &
     (Aj (Am (m<n) => T[i+m] = T[j+m]) => (j>=i))":
```

Here "`n`" can be replaced by any list of free variables. The result is stored (in this case) in the file `tmpalc.mpl`.

We find, for $f(n)$ the number of palindromes of length n of the Thue-Morse sequence

$$v = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mu(0) = \begin{bmatrix}
1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1\\
0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0
\end{bmatrix}
\quad
\mu(1) = \begin{bmatrix}
0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

$$w = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T$$

When we compute the first few terms of the palindrome complexity using these matrices we find

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$f(n)$	1	2	2	2	2	0	4	0	4	0	4	0	4	0	2	0	2	0	4	0	4

This data suggests that the palindrome complexity for $\mathbf{t}$ is bounded above by 4.

How could we prove this? We use something called the "semigroup trick".

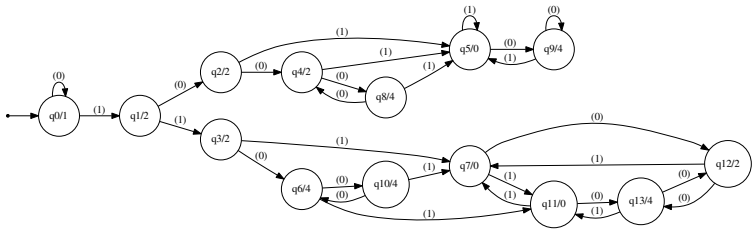
We know that

$$f(n) = v\mu((n)_2)w$$

for the vectors v, w and matrices $\mu(0), \mu(1)$. We can compute the size of the semigroup generated by $\mu(0)$ and $\mu(1)$ using a queue-based algorithm. It is 68.

By computing vMw for all M in this semigroup, we see that $f(n) \in \{0, 1, 2, 4\}$.

Now we can contruct a 2-DFAO out of all possible products Mw, with output of each state Mw equal to vMw. When we do this, we get an automaton with 68 states that can be minimized to one with 14 states:



14.5 Minimal linear representations

There is an algorithm to minimize linear representations. The result is a representation of smallest rank. It may not be unique.

For example, when we run it on the matrices for palindrome complexity of Thue-Morse we get the following minimized representation:

$$v' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mu'(0) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{6}{5} & \frac{1}{5} & \frac{1}{5} & -\frac{7}{10} & \frac{1}{5} \\ 0 & 0 & \frac{2}{5} & \frac{2}{5} & \frac{2}{5} & -\frac{2}{5} & \frac{2}{5} \\ 0 & 0 & \frac{1}{5} & \frac{6}{5} & \frac{1}{5} & -\frac{7}{10} & \frac{1}{5} \end{bmatrix} \quad \mu'(1) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{2}{5} & -\frac{3}{5} & -\frac{3}{5} & \frac{11}{10} & \frac{2}{5} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{2}{5} & \frac{3}{5} & \frac{3}{5} & \frac{2}{5} & -\frac{2}{5} \end{bmatrix}$$

$$w' = \begin{bmatrix} 1 & 2 & 2 & 2 & 2 & 0 & 4 \end{bmatrix}^T$$

We can also find the defining relations for a k-regular sequence. The function $f(n)$ for

palindrome complexity of Thue-Morse satisfies

$$\begin{aligned}
f(4n+3) &= f(4n+1) \\
f(8n) &= f(2n) + f(2n+1) - f(4n+1) \\
f(8n+1) &= f(4n+1) \\
f(8n+4) &= f(8n+2) \\
f(8n+5) &= 0 \\
f(16n+6) &= f(4n+1) + f(4n+2) \\
f(16n+10) &= f(4n+1) + f(4n+2) \\
f(16n+14) &= f(4n+2) \\
f(32n+2) &= f(8n+2) \\
f(32n+18) &= f(8n+6)
\end{aligned}$$

We can mechanically *find* the relations for *any* given k-regular sequence g.

Suppose we are given the linear representation of a k-regular sequence g, that is, vectors v, w and matrices $M_0, M_1, \ldots, M_{k-1}$ such that

$$g(n) = vM_{a_1}M_{a_2}\cdots M_{a_j}w,$$

where $a_1a_2\cdots a_j = (n)_k$.

To make this really work perfectly you need to first insure that $vM_0 = v$. But if you are willing to give up the relations at $n = 0$ this is not absolutely necessary. (This requirement arises from the "leading zeroes" problem; if the canonical representation for n is x, then the canonical representation for $2n$ is $x0$ — *except* if $n = 0$, when it is just x. So if $vM_0w \neq vw$, you have a small problem.)

Now let M be arbitrary and consider vM as a vector with variable entries, say $[a_1, a_2, \ldots, a_d]$.

Successively compute vMM_yw for words y of length $0, 1, 2, \ldots$ over $\Sigma_k = \{0, 1, \ldots, k-1\}$; this will give an expression in terms of the variables $a_1, \ldots, a_d$.

After at most $d + 1$ such relations, we find an expression for vMM_yw for some y as a linear combination of previously computed expressions. When this happens, you no longer need to consider any expression having y as a suffix. Eventually the procedure halts, and this corresponds to a system of equations for g.

Example 116. Let $k = 2$, $v = [6, 1]$, $w = [2, 4]^T$, and

$$\begin{aligned}
M_0 &= \begin{bmatrix} -3 & 1 \\ 1 & 4 \end{bmatrix} \\
M_1 &= \begin{bmatrix} 0 & 2 \\ -3 & 1 \end{bmatrix}
\end{aligned}$$

Suppose M is some product of M_0 and M_1, and suppose $vM = [a, b]$. We find

$$\begin{aligned} vMw &= 2a + 4b \\ vMM_0w &= -2a + 18b \\ vMM_1w &= -8a - 2b \\ vMM_0M_0w &= 24a + 70b \\ vMM_1M_0w &= 36a + 24b \end{aligned}$$

Solving the linear system, we get

$$\begin{aligned} vMM_1w &= \frac{35}{11}vMw - \frac{9}{11}vM_0w \\ vMM_0M_0w &= 13vMw + vM_0w \\ vMM_1M_0w &= \frac{174}{11}vMw - \frac{24}{11}vM_0w. \end{aligned}$$

This gives us, for $n \geq 1$, that

$$\begin{aligned} g(2n+1) &= \frac{35}{11}g(n) + \frac{9}{11}g(2n) \\ g(4n) &= 13g(n) + g(2n) \\ g(4n+2) &= \frac{174}{11}g(n) - \frac{24}{11}g(2n) \end{aligned}$$

In practice this could be speeded up by not letting vM be completely symbolic, but computing the transitive closure of $T := (M_0 \text{ OR } M_1)$ and putting 0's in the entries that correspond to 0's in T.

A small variation of our technique allows us to compute the number $g(n)$ of nonempty palindromes (not necessarily distinct) occurring in prefixes of length n of $\mathbf{t}$. We need a formula asserting that a palindrome occurs in a prefix of length n:

$$(i + \ell \leq n) \ \wedge \ (\forall j < l \ \mathbf{t}[i+j] = \mathbf{t}[i + l - (j+1)]).$$

In `Walnut` this is

```
eval palpref n "(l>0) & (i+l<=n) & (Aj (j<l) =>
                (T[i+j] = T[i+l-(j+1)]))":
```

The result is a linear representation of rank 29. It can be minimized to a linear representation of rank 9:

$$v = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mu(0) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 2 & -3 & -1 & 3 & 1 & 0 & -4 & 3 \\ 0 & 4 & -7 & 1 & 2 & 3 & 3 & -10 & 5 \\ 0 & 10 & -17 & 2 & 5 & 8 & 6 & -24 & 11 \end{bmatrix} \quad \mu(1) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 2 & 0 \\ 0 & 1 & -2 & 1 & 0 & 1 & 0 & -2 & 2 \\ 0 & 3 & -5 & 0 & 2 & 3 & 1 & -7 & 4 \\ 0 & 5 & -9 & 2 & 2 & 4 & 3 & -12 & 6 \\ 0 & 11 & -18 & 2 & 4 & 10 & 6 & -26 & 12 \end{bmatrix}$$

$$w = \begin{bmatrix} 0 & 1 & 2 & 4 & 6 & 8 & 10 & 12 & 18 \end{bmatrix}^T$$

If we want to get asymptotics for $g(n)$, we can consider n of the form 2^k. This corresponds to understanding the asymptotics of the entries of $v\mu(1)\mu(0)^k w$. For this it suffices to understand the asymptotics of $\mu(0)^k$.

Since $\mu(0)$ satisfies its own minimal polynomial $p(X)$, the entries of $\mu(0)^k$ all satisfy a linear recurrence of order at most 9, which can be deduced from $p(X)$.

In `Maple` the minimal polynomial can be computed with the commands

```
with(linalg);
factor(minpoly(m0,X));
```

and we get

$$p(X) = (X+2)(X-2)^2(X-1)^3(X+1)^3.$$

From the fundamental theorem of linear recurrences this means that $v\mu(1)\mu(0)^k w$ can be expressed in the form

$$c_1(-2)^k + (c_2 + c_3 k)2^k + c_4 + c_5 k + c_6 k^2 + (c_7 + c_8 k + c_9 k^2)(-1)^k.$$

When we solve for the constants, we get

$$c_1 = 1/24 \quad c_2 = 37/72 \quad c_3 = 5/12 \quad c_4 = 1/3$$
$$c_5 = 0 \quad c_6 = 0 \quad c_7 = 1/9 \quad c_8 = 0 \quad c_9 = 0$$

$$\text{so } g(2^k) = \frac{1}{24}(-2)^k + \frac{37}{72}2^k + \frac{1}{3} + \frac{5}{12}k2^k + \frac{1}{9}(-1)^k.$$

This gives the asymptotics of $g(2^k)$ as $\Theta(k2^k)$, and so $g(n) = \Theta(n \log n)$.

In general, more detailed asymptotics may require understanding the *joint spectral radius*, which is not easy to compute.

Exercise: Do the same thing for counting the number $g'(n)$ of *distinct* palindromes occurring in a prefix of length n of the Thue-Morse sequence.

– Find the formula – Find a linear representation – Find a closed form for $g'(2^k)$

14.6 Summary of results provable with the method

If $\mathbf{a} = (a_n)_{n \geq 0}$ is a k-automatic sequence, then the following associated sequences are (effectively) k-regular:

- its subword complexity function, $n \to$ number of distinct factors of length n
  - Previously known for fixed points of k-uniform morphisms (Mossé, 1996)
- its palindrome complexity function, $n \to$ number of distinct factors of length n that are palindromes

  - Previously known for fixed points of primitive k-uniform morphisms (Allouche, Baake, Cassaigne, Damanik, 2003)

- its sequence of separator lengths (length of smallest factor that begins at position n and does not occur previously)

  - Previously known for fixed points of k-uniform circular morphisms (Garel, 1997)

14.7 Summary of results

If $\mathbf{a} = (a_n)_{n\geq 0}$ is a k-automatic sequence, then the following associated sequences are k-regular sequences:

- the number of distinct square factors of length n; the number of squares beginning at (centered at, ending at) position n; the length of the longest square beginning at (centered at, ending at) position n; the number of palindromes beginning at (centered at, ending at) position n; the number of distinct recurrent factors of length n; etc.,

  - Previously known for the Thue-Morse sequence (Brown, Rampersad, Shallit, Vasiga, 2006)

If $(a_n)_{n\geq 0}$ is a k-automatic sequence, then the following associated sequences are k-regular sequences:

- The recurrence function of $\mathbf{a}$, $n \to$ the smallest integer t such that every factor of length t of $\mathbf{a}$ contains every factor of length n

- The appearance function of $\mathbf{a}$, $n \to$ the smallest integer t such that the prefix of length t of $\mathbf{a}$ contains every factor of length n

14.8 Other computable functions

- Given a regular language L encoding a set S of pairs of integers, the quantity $\sup_{(p,q)\in S} \frac{p}{q}$ is either infinite or rational, and it is computable

- The critical exponent of an automatic sequence (exponent of the largest power of any factor) is a rational number and is computable.

- The optimal constant for linear recurrence for an automatic sequence is rational and computable.

A sequence $\mathbf{a} = (a_n)_{n \geq 0}$ is linearly recurrent if there is a constant C such that for all $\ell \geq 0$, and all factors x of length ℓ occurring in $\mathbf{a}$, any two consecutive occurrences of x are separated by at most $C\ell$ positions.

Given $\mathbf{a}$, can we determine the smallest value of C that works?

The idea is, given the automaton for $\mathbf{a}$, to construct an automaton accepting the language of pairs (d, ℓ) such that

(a) there is some factor of length ℓ for which there is another occurrence at distance d and

(b) this occurrence is actually the very next occurrence.

Then $\sup_{(d,\ell) \in S} \frac{d}{\ell}$ gives the optimal C.

Chapter 15

Cobham's big theorem

Recall that two integers $k, \ell \geq 2$ are *multiplicatively dependent* if any of the following (equivalent) conditions hold:

(a) there exists $i, j \geq 1$ such that $k^i = \ell^j$;

(b) there exists integers $z \geq 2$ and $r, s \geq 1$ such that $k = z^r$, $\ell = z^s$;

(c) $\log k$ and $\log \ell$ are linearly dependent over $\mathbb{Q}$;

(d) $\log_k \ell \in \mathbb{Q}$;

(e) $\log_\ell k \in \mathbb{Q}$.

Otherwise we say k and ℓ are *multiplicatively independent*.

In this lecture we will prove the following theorem, due to Cobham in 1969 [22]:

Theorem 117. *Let $\mathbf{f} = (f(n))_{n\geq 0}$ be a sequence that is both k-automatic and ℓ-automatic, for k and ℓ multiplicatively independent. Then $\mathbf{f}$ is ultimately periodic.*

This theorem is not easy to prove, but a recent clever proof by Krebs has substantially simplified things. We start by discussing representations in redundant number systems.

15.1 Redundant number systems

Normally base-k representation uses the digits $\{0, 1, \ldots, k-1\}$. However, we could allow a larger digit set D with the usual interpretation

$$[w]_k := \sum_{0 \leq i < n} d_i k^{n-1-i}$$

if $w = d_0 d_1 \cdots d_{n-1}$, where each $d_i \in D$.

We say a sequence $\mathbf{a} = (a(n))_{n\geq 0}$ is (D, k)-*automatic* if there exists a DFAO $M = (Q, \Sigma, \Delta, \delta, q_0, \tau)$ such that $a(n) = \tau(\delta(q_0, w))$ for all $w \in D^*$ such that $[w]_k = n$. In other

words, the DFAO M must return the correct result, *no matter what representation of n is input*. This is a generalization of the requirement we previously imposed that a DFAO should work correctly even if the input contained leading zeros.

Notice that since D might contain negative digits, some representations might correspond to negative n. For our purposes, we don't care what the output is in this case. We could, for example, say that it is always 0.

Lemma 118. *Let $\{0, 1, \ldots, k-1\} \subseteq D \subseteq \mathbb{Z}$. Then a sequence* **a** *is k-automatic iff it is (D, k)-automatic.*

Proof. One direction is trivial: if **a** is (D, k)-automatic, then by removing all transitions on the letters $D - \{0, 1, \ldots, k-1\}$ from the DFAO M, we get a (D', k)-automaton where $D' = \{0, 1, \ldots, k-1\}$.

For the other direction, assume **a** is k-automatic. We show it is (D, k)-automatic.

Note that **a** is also k-automatic in the "reverse intepretation" of the input, where we read the digits in the reverse order, lsd-first, using a DFAO $M = (Q, \Sigma_k, \Delta, \delta, q_0, F)$. The idea of the proof is to create a new DFAO M' that "normalizes" the reversed representation of n on the fly, reading an input consisting of digits from D, maintaining a "carry" as each new digit is read, and simulating M on the normalized representation. There are two issues to handle: how do we know the carries don't become arbitrarily large, and how do we handle the "end" of the input, where there might be unresolved carries.

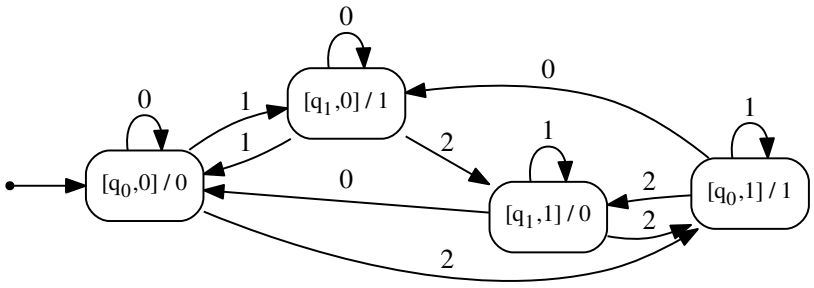
For the first issue, let m be the largest element of D in absolute value. We claim that carries are bounded by $m/(k-1)$ in absolute value. This can be proved by induction on the number of digits of the input: initially, before reading any digits, the carry is 0, which is $\leq m/(k-1)$. Now suppose the current carry is c with $|c| \leq m/(k-1)$. If the next digit is d, with $|d| \leq m$, then the new carry c also satisfies $|c| \leq (|c| + |d|)/k \leq (m/(k-1) + m)/k = m/(k-1)$.

So in the state of M' we can store the carry (which is bounded) and "normalize" the input to use the digits $\{0, 1, \ldots, k-1\}$ only. At the end of the input, we have a leftover carry c, and the appropriate input is then the one corresponding to $(c)_k^R$. This gives the following construction, whose correctness is left to the reader.

- $M = (Q, \Sigma_k, \Delta, \delta, q_0, \tau)$ is the DFAO for the sequence, expecting its input in lsd-first format;
- $M = (Q', D, \Delta, \delta', q_0', \tau')$ is the DFAO we construct for reversed representations, using digits in D;
- $Q = Q \times \{-m', \ldots, -1, 0, 1, \ldots, m'\}$, where $m' = \lfloor m/(k-1) \rfloor$;
- $q_0' = [q_0, 0]$;
- $\delta'([p, c], a) = [q, d]$, where $d = \lfloor (a + c)/k \rfloor$ and $\delta(p, (a + c) \bmod k) = q$.
- $\tau'([q, c]) = \tau(\delta(q, (c)_k^R))$ for $c \geq 0$.

□

Example 119. Consider the Thue-Morse sequence $\mathbf{t}$. Take $D = \{0, 1, 2\}$. The construction gives the following 2-DFAO that takes an input $x \in D^*$ and computes t_n, where $n = [x^R]_k$.



15.2 Approximation of powers

Dirichlet's theorem concerns approximation of real numbers by rationals. In this section we prove a version of this theorem that deals with powers of integers $k, \ell \geq 2$.

Lemma 120. *Let $k, \ell \geq 2$ be integers and let $\epsilon > 0$ be an arbitrary real number. Then there exist positive integers m, n such that $|k^m - \ell^n| \leq \epsilon \min(k^m, \ell^n)$.*

Proof. Without loss of generality assume $k \geq \ell$. Define $f(i) = \lfloor \log_\ell k^i \rfloor$ for $i \geq 0$. Hence

$$(\log_\ell k^i) - 1 < f(i) \leq \log_\ell k^i$$

and so

$$f(i+1) - f(i) < ((\log_\ell k^{i+1}) - 1) - \log_e llk^i = (\log_\ell k) - 1 \geq 0.$$

So $(f(i))_{i \geq 0}$ is strictly increasing.

Choose t sufficiently large so that $\epsilon \geq (\ell - 1)/(t - 1)$ and consider the rational numbers $k^i/\ell^{f(i)} \in [1, \ell)$ for $0 \leq i < t$. By the pigeonhole principle, there exist integers i, j with $0 \leq i < j < t$ such that $|k^j/\ell^{f(j)} - k^i/f\ell^{f(i)}| \leq \epsilon$. Now multiply this by $\ell^{f(j)}/k^i$ to get

$$\begin{aligned} |k^{j-i} - \ell^{f(j)-f(i)}| &\leq \epsilon \ell^{f(j)}/k^i \\ &\leq \epsilon \min(k^{j-i}, \ell^{f(j)-f(i)}), \end{aligned}$$

because $k^r \geq \ell^{f(r)}$ for all r. □

Example 121. Take $k = 3$, $\ell = 2$. For various ϵ here are the smallest corresponding exponents:

ϵ	m	n
0.1	5	8
0.01	53	84
0.001	665	1054
0.0001	665	1054
0.00001	31867	50508

We remark that an efficient way to find more examples is to look at the convergents to the continued fraction of $\log_\ell k$.

15.3 Local periods

An *interval* is a block of consecutive integers. We say a sequence $f : \mathbb{N} \to \Delta$ has *local period* $p \geq 1$ on an interval $I \subseteq \mathbb{N}$ if $f(i) = f(i+p)$ for all i such that both i and $i+p$ both belong to I. Note that every finite interval has some local period (because in the worst case we can take the local period to be the cardinality of the interval).

Lemma 122. *Suppose $f : \mathbb{N} \to \Delta$ has local period p on an interval I and local period q on an interval J. If $|I \cap J| \geq p + q$, then f has local period p on $I \cup J$.*

Proof. Choose i such that i and $i+p$ are both in $I \cup J$. If $i, i+p \in I$ then $f(i) = f(i+p)$ by hypothesis.

Case (a): $i, i+p \in J$. Then since $I \cap J$ has cardinality $\geq p+q$, there must exist j such that both j and $j+p$ are in $I \cap J$ and $j \equiv i \pmod{q}$. In this case

$$\begin{aligned} f(i) &= f(j) \quad \text{by local periodicity of } J; \\ &= f(j+p) \quad \text{by local periodicity of } I; \\ &= f(i+p) \quad \text{by local periodicity of } J. \end{aligned}$$

Case (b): $i \in I - J$, $i + p \in J - I$. This implies that $p > p + q$, a contradiction. So this case cannot occur. □

15.4 Proof of Cobham's theorem

We are now ready to complete Krebs' proof of Cobham's theorem.

Proof. For $x, r \in \mathbb{R}$ define the intervals

$$\begin{aligned} B(x, r) &= (x - r, x + r) \cap \mathbb{Z} \\ B[x, r] &= [x - r, x + r] \cap \mathbb{Z}. \end{aligned}$$

Let $k, \ell \geq 2$ be multiplicatively independent. Let $\mathbf{f} = (f(n))_{n \geq 0}$ be k-automatic and ℓ-automatic. Define the digit sets

$$\begin{aligned} D_k &= B(0, k) = \{1 - k, \ldots, -1, 0, 1, \ldots, k - 1\} \\ D_\ell &= B(0, \ell) = \{1 - \ell, \ldots, -1, 0, 1, \ldots, \ell - 1\}. \end{aligned}$$

Then there exists a (D_k, k)-DFAO

$$M_k = (Q_k, D_k, \Delta, \delta_k, q_{0,k}, \tau_k$$

and a (D_ℓ, ℓ)-DFAO

$$M_\ell = (Q_\ell, D_\ell, \Delta, \delta_\ell, q_{0,\ell}, \tau_\ell)$$

computing $\mathbf{f}$.

For each state $s \in Q_k$ and $t \in Q_\ell$, define the sets of natural numbers

$$X_s = \{[w]_k \in \mathbb{N}_{>0} : \exists w \in D_k^* \text{ with } \delta_k(q_{0,k}, w) = s\}$$
$$X_t = \{[w]_\ell \in \mathbb{N}_{>0} : \exists w \in D_\ell^* \text{ with } \delta_\ell(q_{0,\ell}, w) = t\}.$$

Thus, the families $(X_s)_{s \in Q_k}$ and $(Y_t)_{t \in Q_\ell}$ each form a finite cover of $\mathbb{N}_{>0}$, and f is constant on each set X_s and Y_t.

Let i, j be integers in X_s. Let w_i be any base-k representation of i using the digit set D_k, and similarly w_j for j. Let $n \geq 0$ and let $|m| < k^n$. Let w_m be any representation for m. Now

$$\delta_k(q_{0,k}, w_i) = s = \delta_k(q_{0,k}, w_j)$$

So $\delta_k(q_{0,k}, w_i w_m) = \delta_k(q_{0,k} w_j w_m)$. Thus we have shown

$$f(i \cdot k^n + m) = f(j \cdot k^n + m) \tag{15.1}$$

for all $n \geq 0$, $m \in B(0, k^n)$, and $i, j \in X_s$.

By exactly the same argument for Y_t, we get

$$f(i \cdot \ell^n + m) = f(j \cdot \ell^n + m) \tag{15.2}$$

for all $n \geq 0$, $m \in B(0, k^n)$, and $i, j \in Y_t$.

Now consider all the pairwise intersections $X_s \cap Y_t$ for $s \in Q_k$ and $t \in Q_\ell$. There are finitely many such intersections, and together they form a finite cover of $\mathbb{N}_{>0}$. Let

$$S = \{(s, t) : |X_s \cap Y_t| \geq 2\}.$$

The infinite pigeonhole principle shows that S is nonempty.

For each $(s, t) \in S$ choose $i_{st}, j_{st} \in X_s \cap Y_t$ with $i_{st} < j_{st}$. By Lemma 120 there exist m, n such that

$$|k^m - \ell^n| \leq \frac{1}{4N} \min(k^m, \ell^n),$$

where N is the maximum of all the j_{st}.

Without loss of generality, assume $k^m > \ell^n$. We now claim that for each $(s, t) \in S$ and $i \in X_s \cap Y_t$ that f has local period

$$p_{st} = (j_{st} - i_{st})(k^m - \ell^n) \leq \ell^n / 4$$

on the interval $I_i := B[i \cdot \ell^n, \frac{3}{4}\ell^n]$.

To see this, pick $q \in B[0, \frac{3}{4}\ell^n]$. Then $q, q + p_{st} \in B(0, \ell^n)$. Furthermore

$$\begin{aligned} |q - i_{st}(k^m - \ell^n)| &\leq |q| + i_{st}(k^m - \ell^n) \\ &\leq \frac{3}{4}\ell^n + \frac{1}{4}\ell^n \\ &= \ell^n < k^m, \end{aligned}$$

so $|q - i_{st}(k^m - \ell^n)| \in B(0, k^m)$.

Hence we have

$$\begin{aligned} f(i \cdot \ell^n + q) &= f(i_{st} \cdot \ell^n + q) \quad \text{(because } i, i_{st} \in X_s \cap Y_t \text{ and Eq. (15.2))} \\ &= f(i_{st} \cdot k^m + q - i_{st}(k^m - \ell^n)) \quad \text{(by rearrangement)} \\ &= f(j_{st} \cdot k^m + q - i_{st}(k^m - \ell^n)) \quad \text{(because } i_{st}, j_{st} \in X_s \cap Y_t \text{ and Eq. (15.1))} \\ &= f(j_{st} \cdot \ell^n + q + p_{st}) \quad \text{(by definition of } p_{st}) \\ &= f(i \cdot \ell^n + q + p_{st}) \quad \text{(because } i, j_{st} \in X_s \cap Y_t \text{ and Eq. (15.2)).} \end{aligned}$$

Now $\{X_s \cap Y_t : (s,t) \in S\}$ contains all but finitely many elements of $\mathbb{N}$, so there exists i_0 such that this set covers $i_0 + \mathbb{N}$. Let p_{st} be a local period of f on I_{i_0}. We now show by induction that f has local period p_{st} on $\bigcup_{i_0 \leq j \leq r} I_j$ for each $r \geq i_0$. The base case, $r = i_0$, is trivial. Otherwise, f has local period p_{st} on $\bigcup_{i_0 \leq j < r} I_j$ by induction, and local period $p_{s't'}$ on I_r. We claim that Lemma 122 implies that it has local period p_{st} on $\bigcup_{i_0 \leq j \leq r} I_j$. To see this, observe that

$$\left(\bigcup_{i_0 \leq j < r} I_j \right) \cap I_r = B[(r - \frac{1}{2})\ell^n, \frac{1}{4}\ell^n]$$

has cardinality $\geq \lfloor \ell^n/2 \rfloor \geq 2\lfloor \ell^n/4 \rfloor \geq p_{st} + p_{s't'}$. So f has local period p_{st} on $\bigcup_{j \geq i_0} I_j$, and hence f is ultimately periodic. □

15.5 More about Cobham's theorem

In this section we mention some other important aspects of Cobham's theorem, including generalizations. These are stated without proof.

15.5.1 The Cobham-Semenov theorem

Recall that $V_k(n) = \max_{\{} k^i : k^i \mid n\}$. A set $S \subseteq \mathbb{N}^t$ is said to be *linear* if it can be written as

$$S = \{v_0 + a_1 v_1 \cdots + a_r v_r : a_1, a_2, \ldots, a_r \in \mathbb{N}\}$$

for some vectors $v_0, v_1, \ldots, v_r \in \mathbb{N}^t$. A set is *semilinear* if it is the finite union of linear sets. Let $t \geq 1$ in what follows.

Theorem 123. *A set $S \subseteq \mathbb{N}^t$ is semilinear iff it is definable in* $\mathrm{FO}(\mathbb{N}, +)$.

Theorem 124. *A set $S \subseteq \mathbb{N}^t$ is semilinear iff it is definable in* $\mathrm{FO}(\mathbb{N}, +, V_k)$ *and* $\mathrm{FO}(\mathbb{N}, +, V_\ell)$ *for two multiplicatively independent integers $k, \ell \geq 2$.*

15.5.2 Generalization to k-regular sequences

Jason Bell proved

Theorem 125. *Let* $\mathbf{f} = (f(n))_{n\geq 0}$ *be a sequence taking values in* $\mathbb{Z}$*. Then* $\mathbf{f}$ *is both* k*- and* ℓ*-regular, for* k *and* ℓ *multiplicatively independent, iff*

$$\sum_{n\geq 0} f(n)X^n$$

is the power series expansion of a rational function all of whose poles are roots of unity only.

Remark 126. Examples of such sequences include all integer-valued polynomial sequences $(p(n))_{n\geq 0}$ and the r-way merge of such sequences.

15.5.3 Generalization to more general morphic sequences

Recall that a sequence is called *pure morphic* if it is the fixed point of a prolongable morphism, and *morphic* if it is the image of such a word under a coding (a 1-uniform morphism). A morphism h is called *primitive* if there exists an integer n such that for all letters a, b we have that a occurs in $h^n(b)$.

Given a morphism h, we can form its associate matrix M_h, as follows: $(M_h)_{i,j} = |h(a_i)|_{a_j}$, where $\Sigma = \{a_1, a_2, \ldots, a_r\}$ and $h : \Sigma^* \to \Sigma^*$. The largest eigenvalue of M_h in absolute value is called the *dominant eigenvalue* of h. An algebraic number α is called *Perron* if $\alpha < 1$, but all of the conjugates of α are less than α in absolute value. The *conjugates* of an algebraic number are all the other zeros of the minimal polynomial for α.

Theorem 127 (Durand)**.** *Let* σ, φ *be two primitive morphisms with dominant eigenvalues* α *and* β*, respectively, with* α, β *multiplicatively independent, and Perron. Then a sequence is a fixed point of* σ *and* φ *iff it is ultimately periodic.*

Example 128. Consider the morphism $h : 1 \to 121, 1 \to 12221$. It is possible to show its fixed point $\mathbf{x}$ is not 2-automatic (see [3]), and hence not 2^i-automatic for any $i \geq 1$. The dominant eigenvalue of h is 2, and $\mathbf{x}$ is not ultimately periodic. Therefore, by Durand's theorem, $\mathbf{x}$ is not k-automatic for any k.

15.5.4 A Cobham density theorem

Recently Byszewski and Konieczny [14] proved the following beautiful generalization of Cobham's theorem: if $(a_n)_{n\geq 0}$ is k-automatic and $(b_n)_{n\geq 0}$ is ℓ-automatic and k, ℓ are multiplicatively independent and $\{n < N : a_n \neq b_n\} = o(N)$, then there is a periodic sequence $(c_n)_{n\geq 0}$ such that $\{n < N : a_n \neq c_n\} = o(N)$.

15.5.5 Common factors between automatic sequences

Recently Byszewski, Konieczny, and Krawczyk [15] proved that the set of common factors between a k-automatic sequence and an ℓ-automatic sequence must be very restricted.

Theorem 129. *Let k, ℓ be multiplicatively independent. Then if $\mathbf{a} = (a_n)_{n\geq 0}$ is k-automatic and $\mathbf{b} = (b_n)_{n\geq 0}$ is ℓ-automatic, the set of finite factors occurring in both sequences is a finite union of the factors occurring in u^*vw^* for finite words u, v, w.*

15.5.6 Longest common prefix

Let $\mathbf{a} = (a_n)_{n\geq 0}$ be a k-automatic sequence, and $\mathbf{b} = (b_n)_{n\geq 0}$ be an ℓ-automatic sequence, with k and ℓ multiplicatively independent. Cobham's theorem shows that $\mathbf{a}$ and $\mathbf{b}$ cannot agree forever, unless they are both the same ultimately periodic sequence. However, even if they are both not ultimately periodic, they can agree on an exponentially long prefix (measured in terms of the minimal DFAO's generating $\mathbf{a}$ and $\mathbf{b}$).

As an example, consider the sets $A(R) := \{k^r : r \geq R\}$ and $B(S) := \{\ell^s : s \geq S\}$. Let $\mathbf{a}$ be the characteristic sequence of $A(R)$ and $\mathbf{b}$ be the characteristic sequence of $B(S)$. If $k^R \geq \ell^S$, then these two sequences agree on the first ℓ^S terms. But $\mathbf{a}$ can be generated by a DFAO with $O(R)$ states and $\mathbf{b}$ can be generated by a DFAO with $O(S)$ states.

So it would be interesting to obtain an upper bound on how long two such sequences can agree.

Mol et al. [37] recently proved the following upper bound: if $\mathbf{a} = (a_n)_{n\geq 0}$ is an aperiodic k-automatic sequence, and $\mathbf{b} = (b_n)_{n\geq 0}$ is an aperiodic ℓ-automatic sequence, with k and ℓ multiplicatively independent, then $\mathbf{a}$ and $\mathbf{b}$ can agree for at most the first

$$\theta^{\theta^{cR^4S^4}}$$

terms, where $\theta = \max(k, \ell)$, and R is the number of states in the DFAO for $\mathbf{a}$ and S is the number of states in the DFAO for $\mathbf{b}$, and c is a constant depending only on k and ℓ.

15.6 Notes

Cobham's original proof of his theorem is in [22]. Krebs' proof of Cobham's theorem is in [32].

Chapter 16

Automatic real numbers

Let b be an integer ≥ 2. Given a real number x, we can consider its base-b representation

$$x = a_0 + \sum_{i \geq 1} a_i b^{-i}$$

where $a_0 \in \mathbb{Z}$ and $a_i \in \Sigma_b = \{0, 1, \ldots, b-1\}$ for $i \geq 1$. For example, $-\pi$ is associated with the sequence $(-4, 8, 5, 8, 4, 0, 7, \ldots)$.

Notice that some numbers have two distinct representations as sequences. For example, in base k the number 0 has two representations: $(0, 0, 0, \ldots)$ and $(-1, k-1, k-1, k-1, \ldots)$.

We say that such a real number x is (k, b)-automatic if its base-b sequence $(a_i)_{i \geq 0}$ is k-automatic. The set of all such real numbers is $L(k, b)$.

16.1 Basic results

Proposition 130. *The number* $\mathbf{x}$ *is rational iff* $x \in L(k, b)$ *for all* $k, b \geq 2$.

Proof. $\Longrightarrow$: the number $\mathbf{x}$ is rational implies that its base-b representation is ultimately periodic.

$\Longleftarrow$: fix b. If the base-b representation of x is in $L(k, b)$ for all k, then by Cobham's theorem it is ultimately periodic, and hence x is a rational number. □

Corollary 131. *Let* j, k *be multiplicatively independent integers. Then* $L(j, b) \cap L(k, b) = \mathbb{Q}$.

Proposition 132. *If* $x \in L(k, b)$, *then* $-x \in L(k, b)$.

Proof. Suppose $x = a_0 + \sum_{i \geq 1} a_i b^{-i}$, and assume $a_0 \geq 0$. Then it is easy to check that

$$-x = (-a_0 - 1) + \sum_{i \geq 1} (b - 1 - a_i) b^{-i}.$$

However, if $\mathbf{a} := (a_i)_{i \geq 1}$ is k-automatic, then the sequence $(b - 1 - a_i)_{i \geq 1}$ is just a coding applied to $\mathbf{a}$, and hence also k-automatic. □

16.2 Lehr's theorem

In this section, we prove Lehr's theorem [33].

Theorem 133 (Lehr)**.** *The set $L(k, b)$ forms a $\mathbb{Q}$-vector space.*

The proof has several parts. The first part is the following:

Lemma 134. *If $x, y \in L(k, b)$, then $x + y \in L(k, b)$.*

The original proof of this fact was rather complicated (and so is the simplification in the textbook). Using our logical characterization of automatic sequences, however, it becomes much simpler.

Proof. The whole difficulty of the proof is that when we add x to y, the carries that influence a given position could, potentially, come from arbitrarily far to the right of that position.

Let $b \geq 2$ and define

$$x = a_0 + \sum_{i \geq 1} a_i b^{-i}$$
$$y = b_0 + \sum_{i \geq 1} b_i b^{-i}$$
$$x + y = c_0 + \sum_{i \geq 1} c_i b^{-i}$$

where $c_i = a_i + b_i$ for $i \geq 0$. Using the cross-product construction, we see that $\mathbf{c} = (c_i)_{i \geq 0}$ is a k-automatic sequence, but may not be a "legitimate" base-b expansion, because some of the c_i for $i \geq 1$ could exceed $b - 1$. Thus, we have to "normalize" the representation of $x + y$ and show that this process still results in a k-automatic sequence. Normalization involves correctly processing the carries.

Just like when we add 1 to $999 \cdots 9$ in base 10, a carry arising from a b in position i can influence arbitrarily many digits to the left, but only if there is a string of consecutive digits equal to $b - 1$. For $i \geq 0$ define

$$d_i := \begin{cases} 1, & \text{if } \exists j > i \text{ such that } c_j \geq b \text{ and } c_{i+1}, \ldots, c_{j-1} = b - 1; \\ 0, & \text{otherwise.} \end{cases}$$

and $e_i = (c_i + d_i) \bmod b$. Then the reader can check again that $x + y = \sum_{i \geq 0} e_i b^{-i}$ and $0 \leq e_i < b$ for $i \geq 1$. So it just remains to prove that the sequence $\mathbf{e} = (e_i)_{i \geq 0}$ is k-automatic.

To see this, we use the logical characterization of automatic sequences discussed in Lectures 10 and 11. The definition of d_i is clearly specified by a first-order formula and hence is $(d_i)_{i \geq 0}$ is k-automatic. Hence $(e_i)_{i \geq 0}$ is k-automatic, and the result is proved. □

Example 135. Consider the number x whose base-2 representation $\mathbf{a} = (a_i)_{i \geq 0}$ is given by $a_i = 1$ if $(i)_2$ starts with 10, and 0 otherwise, and the number y whose base-2 representation $\mathbf{b} = (b_i)_{i \geq 0}$ is given by $b_i = 1$ if $(i)_2$ starts with 101, and 0 otherwise. Here are the first few terms of the sequences in the proof:

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
a_i	0	0	1	0	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1
b_i	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1
c_i	0	0	1	0	1	2	0	0	1	1	2	2	0	0	0	0	1	1	1	1	2	2
d_i	0	0	0	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1
e_i	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	1

Here are the corresponding automata:

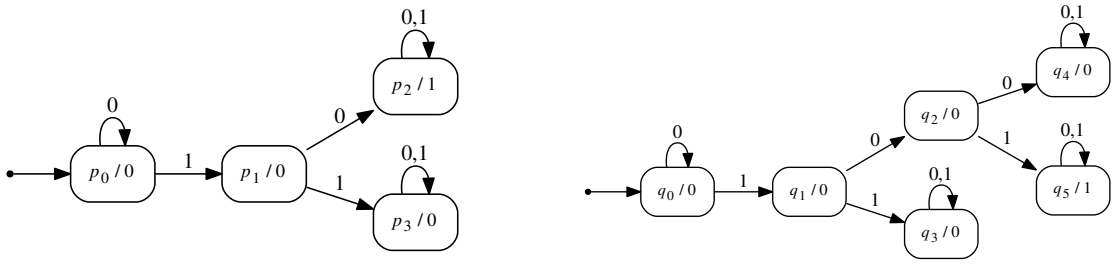


Figure 16.1: Automaton for the sequences **a** and **b**

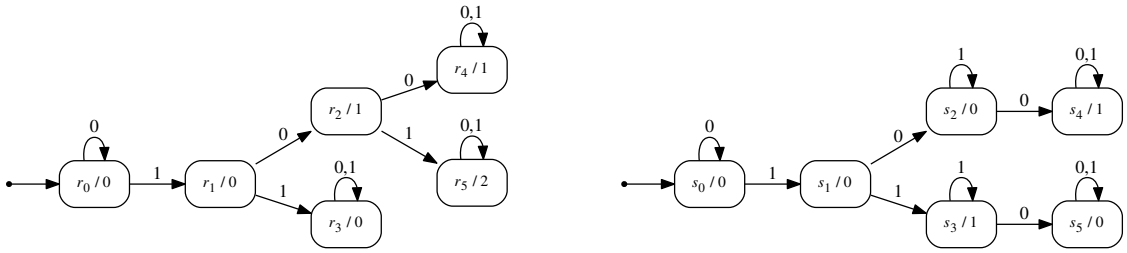


Figure 16.2: Automaton for the sequences **c** and **d**

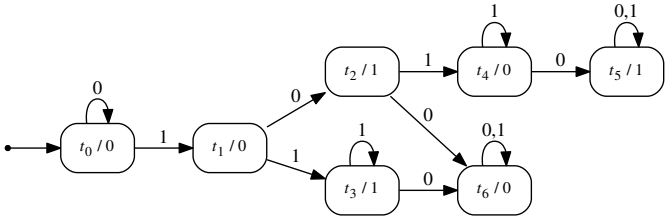


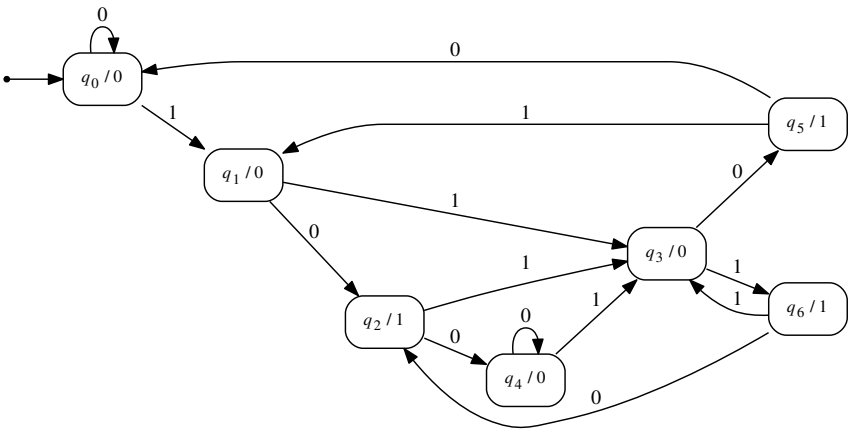
Figure 16.3: Automaton for the sequence **e**

The second part of the proof of Theorem 133 is the following:

Lemma 136. *If $x \in L(k,b)$ and c is an integer ≥ 1 then $x/c \in L(k,b)$.*

Proof. Let $x = a_0 + \sum_{i \geq 1} a_i b^{-i}$. Then long division by c in base b can be carried out by a finite-state transducer acting on the k-automatic sequence $(a_i)_{i \geq 0}$. By Section 6.9 of the course text, the resulting sequence is k-automatic. □

Example 137. Consider the Thue-Morse real number $\tau := \sum_{i \geq 0} t_i 2^{-i}$, where $\mathbf{t} = t_0 t_1 \cdots$ is the Thue-Morse sequence. Then $\tau/3 = \sum_{i \geq 0} t'_i 2^{-i}$ where $(t'_i)_{i \geq 0}$ is generated by the following 2-DFAO:



We can now complete the proof of Theorem 133:

Proof. It suffices to show that if $x \in L(k,b)$ and α is rational, then $\alpha x \in L(k,b)$. Write $\alpha = \pm p/q$ for integers $p \geq 0$, $q \geq 1$. By Lemma 136 $x/q \in L(k,b)$. By repeated application of Lemma 134 $px/q \in L(k,b)$. By Proposition 132 if necessary, $\alpha \in L(k,b)$. □

One nice thing about the k-automatic real numbers is that we can decide basic things about them.

Theorem 138. *The following decision problem is recursively solvable: given a k-DFAO computing the base-b representation of a real number x, is x rational?*

Proof. Suppose x has base-b representation $a_0 + \sum_{i \geq 1} a_i b^{-i}$. Then x is rational iff the sequence $\mathbf{a} = (a_i)_{i \geq 0}$ is ultimately periodic. This is represented by the first-order formula

$$\exists p, n_0 \ (p \geq 1) \ \wedge \ \forall t \ (t \geq n_0) \implies \mathbf{a}[t] = \mathbf{a}[t+p],$$

and hence its truth can be decided by the decision procedure in Lecture 11. □

We can also tell if two k-automatic real numbers are equal. At first glance this seems like a triviality, but recall that numbers may have two representations.

Theorem 139. *The following decision problem is recursively solvable: given two k-DFAO's computing the base-b representation of numbers x and y, respectively, decide if $x = y$.*

Proof. Form the minimal DFAO for the difference $x - y$ using the algorithms implied by the proof of Theorem 134 and Proposition 132. This DFAO generates either the sequence $(0, 0, 0, \ldots)$ or $(-1, k-1, k-1, \ldots)$ (both of which are easily checkable) iff $x = y$. □

16.3 Non-closure of automatic real numbers

It is natural to wonder, after seeing Theorem 133, whether the automatic real numbers are closed under operations like multiplication and division. Unfortunately, this is not the case.

Theorem 140. *Let $b, k \geq 2$ be integers. Define $y = \sum_{r \geq 0} b^{-k^r}$ and $z = \sum_{\substack{m \geq 1 \\ n \geq 0}} b^{-(k^m - 1)k^n}$. Then $y, z \in L(k, b)$, but $yz \notin L(k, b)$.*

Proof. Define $f(X) = \sum_{r \geq 0} X^{k^r}$ and $g(X) = \sum_{\substack{m \geq 1 \\ n \geq 0}} X^{(k^m - 1)k^n}$. Then $y = f(1/b)$ and $z = g(1/b)$. We claim that $y, z \in L(k, b)$. For y this is because its base-b digits are all 0's and 1's with the i'th digit 1 iff $(i)_k \in 10^*$. For z this is because its base-b digits are all 0's and 1's with the i'th digit 1 iff $(i)_k \in (k-1)^+0^*$.

Now

$$\begin{aligned} f(X)g(X) &= \sum_{\substack{m \geq 1,\ n \geq 0 \\ r \geq 0}} X^{k^r} X^{(k^m - 1)k^n} \\ &= \sum_{\substack{m \geq 1,\ n \geq 0 \\ r \geq 0}} X^{k^r + (k^m - 1)k^n} \\ &= S(X) + T(X) + U(X), \end{aligned}$$

where

$$\begin{aligned} S(X) &= \sum_{\substack{r < n \\ m \geq 1,\ n \geq 0 \\ r \geq 0}} X^{k^r + (k^m - 1)k^n} \\ T(X) &= \sum_{\substack{r = n \\ m \geq 1,\ n \geq 0 \\ r \geq 0}} X^{k^r + (k^m - 1)k^n} \\ U(X) &= \sum_{\substack{r > n \\ m \geq 1,\ n \geq 0 \\ r \geq 0}} X^{k^r + (k^m - 1)k^n}. \end{aligned}$$

Now

$$\begin{aligned} S(X) &= \sum_{\substack{r<n \\ m\geq 1,\ n\geq 0 \\ r\geq 0}} X^{k^r+(k^m-1)k^n} \\ &= \sum_{\substack{r<n \\ m\geq 1,\ n\geq 0 \\ r\geq 0}} X^{k^r(1+k^{n-r}(k^m-1)} \\ &= \sum_{\substack{m\geq 1,\ p\geq 1 \\ r\geq 0}} X^{k^r(1+k^p(k^m-1))} \\ &= \sum_{\substack{m\geq 1,\ p\geq 1 \\ r\geq 0}} X^{k^r(k^{p+m}-k^p+1)}. \end{aligned}$$

Now

$$(k^r(k^{p+m}-k^p+1))_k = (k-1)^m\, 0^{p-1}\, 1\, 0^r$$

so $S(X) = \sum_{i\geq 0} s_i X^i$ where

$$s_i = \begin{cases} 1, & \text{if } (i)_k \in (k-1)^+\, 0^*\, 1\, 0^*; \\ 0, & \text{otherwise.} \end{cases}.$$

So $(s_i)_{i\geq 0}$ is k-automatic and $S(1/b) \in L(k,b)$.

Similarly

$$\begin{aligned} U(X) &= \sum_{\substack{r>n \\ m\geq 1,\ n\geq 0}} X^{k^r+(k^m-1)k^n} \\ &= \sum_{\substack{r>n \\ m\geq 1,\ n\geq 0}} X^{k^n(k^{r-n}+k^m-1)} \\ &= \sum_{\substack{m\geq 1,\ n\geq 0 \\ q\geq 1}} X^{k^n(k^q+k^m-1))}, \end{aligned}$$

where in the last summation we took $q = r-n$.

Now

$$(k^n(k^q+k^m-1))_k = \begin{cases} 1\, 0^{q-m}\, (k-1)^m\, 0^n, & \text{if } m<q; \\ 1\, (k-1)^m\, 0^n, & \text{if } m=q; \\ 1\, 0^{m-q}\, (k-1)^q\, 0^m, & \text{if } m<q. \end{cases}$$

So $U(X) = \sum_{i\geq 0} u_i X^i$ where

$$u_i = \begin{cases} 2, & \text{if } (i)_k \in 1\, 0^+\, (k-1)^+\, 0^*; \\ 1, & \text{if } (i)_k \in 1\, (k-1)^+\, 0^*; \\ 0, & \text{otherwise.} \end{cases}.$$

So $(u_i)_{i\geq 0}$ is k-automatic. If $b \geq 3$, then $U(1/b) \in L(k,b)$ immediately. Otherwise, if $b = 2$ we need to appeal to the "normalization" to get rid of the 2's, as we did in the proof of Theorem 134.

Now

$$\begin{aligned} T(X) &= \sum_{\substack{r=n \\ m\geq 1,\ n\geq 0 \\ r\geq 0}} X^{k^r + (k^m-1)k^n} \\ &= \sum_{\substack{r\geq 0 \\ m\geq 1}} X^{k^{m+r}} \\ &= \sum_{n\geq 1} nX^{k^n}, \end{aligned}$$

because there are n ways to write n as the sum of a non-negative integer and a positive integer.

Consider the base-b expansion of $T(1/b)$, say

$$T(1/b) = c_0 + \sum_{i\geq 1} c_i b^{-i}.$$

From above we see that the base-b digits to the left of position k^n are $(n)_b$. So every element of Σ_b^* appears as a subword of $\mathbf{c} = c_0 c_1 \cdots$. So the subword complexity of $\mathbf{c}$ is b^n. But a k-automatic sequence has subword complexity $O(n)$, as we saw in Lecture 12. So $\mathbf{c}$ is not k-automatic.

Finally, to get a contradiction, suppose that $L(k,b)$ is closed under multiplication. Then $yz \in L(k,b)$ by assumption, and we showed above that $S(1/b), U(1/b) \in L(k,b)$. So $yz - S(1/b) - U(1/b) = T(1/b) \in L(k,b)$. But then $\mathbf{c}$ would be k-automatic, a contradiction. □

Corollary 141. *The set $L(k,b)$ is not closed under the map $x \to x^2$.*

Proof. Suppose it were. Since

$$yz = \frac{1}{4}((y+z)^2 - (y-z)^2),$$

then $L(k,b)$ would be closed under multiplication, a contradiction. □

Corollary 142. *The set $L(k,b)$ is not closed under the map $x \to 1/x$.*

Proof. Suppose it were. Since

$$y^2 = y + \frac{1}{\frac{1}{y-1} - \frac{1}{y}},$$

then $L(k,b)$ would be closed under squaring, a contradiction. □

Chapter 17

Transcendence of automatic real numbers

Recall that a real number α is said to be *algebraic* if it is the zero of a polynomial with integer coefficients. Examples include $(1+\sqrt{5})/2$ (zero of $X^2 - X - 1$) and $\sqrt[3]{2}$ (zero of $X^3 - 2$). Otherwise α is transcendental.

Famous transcendental numbers include π, e, $\log 2$, and many other "natural" constants.

How about automatic real numbers like the Thue-Morse real number, whose base 2 expansion is $.0110100110010110\cdots$, equal to 0.41245403364 in base 10?

There is a beautiful theorem that answers this question, originally stated by Cobham in a 1968 technical report [20, 21], but his proof was flawed. Later, Loxton and van der Poorten tried to prove it [36], but their proof was also flawed. Many contributed to the final proof (including Adamczewski, Bugeaud, Luca, Ferenczi, Mauduit, Allouche, Zamboni), but the version I will present in the notes comes from a survey paper of Yann Bugeaud [13].

Theorem 143. *Let $k, b \geq 2$. Let $\alpha \in L(k, b)$. Then α is rational or transcendental.*

17.1 Transcendence by rational approximation

Many proofs of transcendence are based on an observation originally due to Liouville in 1844 [34]: *an algebraic real irrational number cannot be approximated too closely by rationals.* Liouville used this to prove that the number

$$\alpha = \sum_{n \geq 1} 10^{-n!}$$

is transcendental. This was the first explicit example of a transcendental number. He showed that for all r, there exist integers p, q such that

$$\left| \alpha - \frac{p}{q} \right| = \frac{1}{q^r}.$$

Liouville's criterion was improved repeatedly (by Thue, Siegel, Dyson) until Roth's theorem:

Theorem 144. *Let α be a real algebraic irrational number and let $e > 2$. Then there are only finitely many solutions to the inequality*

$$\left|\alpha - \frac{p}{q}\right| < \frac{1}{q^e} \tag{17.1}$$

in integers p, q.

The *irrationality measure* $\mu(\alpha)$ of a real number α is the supremum, over all exponents e, for which the inequality (17.1) holds for infinitely many integers p, q. Thus Roth's theorem says that if $\mu(\alpha) > 2$, then α is transcendental.

As a consequence of Roth's theorem, we can give an example of transcendental numbers in $L(k, b)$:

Corollary 145. *The numbers*

$$\alpha_{k,b} := \sum_{n\geq 1} b^{-k^n}$$

are transcendental for integers $k \geq 3$ and $b \geq 2$.

Proof. We have

$$b^{k^n}\alpha_{k,b} = p + \sum_{i\geq 1} b^{k^n - k^{n+i}}$$

for some integer p, so

$$\begin{aligned}\left|\alpha_{k,b} - \frac{p}{b^{k^n}}\right| &= \sum_{i\geq 1} b^{-k^{n+i}} \\ &< 2 \cdot b^{-k^{n+1}},\end{aligned}$$

so letting $q = b^{k^n}$ we have

$$\left|\alpha_{k,b} - \frac{p}{q}\right| = 2q^{-k}$$

for infinitely many n and $k \geq 3$. By Roth's theorem, $\alpha_{k,b}$ is transcendental. □

Another example is the following, from [49]:

Theorem 146. *Let $\alpha = \sum_{n\geq 2} 2^{-F_n}$, where F_n is the n'th Fibonacci. Then α is transcendental and its irrationality measure $\mu(\alpha)$ is at least $3(1+\sqrt{5})/4 \doteq 2.427$.*

Proof. Define the formal power series $f(X) = \sum_{n\geq 2} X^{-F_n}$. Then you can check that if $q(X) := X^{F_{n-1}+F_{n-4}} + X^{F_{n-2}+F_{n-4}} - X^{F_{n-4}}$ then

$$q(X)f(X) = p(X) + O(X^{-(F_n+F_{n-5})}),$$

where p is a polynomial. So

$$\begin{aligned} f(X) - \frac{p(X)}{q(X)} &= X^{-(F_n+F_{n-5}+F_{n-1}+F_{n-4})} + \text{ lower order terms} \\ &= X^{-(F_{n+1}+F_{n-3})} + \text{ lower order terms.} \end{aligned}$$

Setting $X = 2$, we see that

$$\left| f(2) - \frac{p(2)}{q(2)} \right| \approx q(2)^{(F_{n+1}+F_{n-3})/(F_{n-1}+F_{n-4})}.$$

Now

$$\lim_{n\to\infty} \frac{F_{n-1}+F_{n-3}}{F_{n-1}+F_{n-4}} = \frac{3(1+\sqrt{5})}{4} \doteq 2.427,$$

so $\mu(\alpha) > 2$ and by Roth's theorem α is transcendental. □

17.2 Outline of the proof of Theorem 143

In these notes I have simplified the proof of Bugeaud to work in the specific case of k-automatic sequences. His proof was slightly more general, working for any sequence of $O(n)$ subword complexity, but this generality introduced a bit of complication.

Here is the outline of the proof.

Step 1. We argue that if $\mathbf{a} = (a_i)_{i\geq 0}$ is a k-automatic sequence taking values in Σ_b, then for each n, there is a prefix of $\mathbf{a}$ of length $O(n)$ that can be factored as follows: $W_n U_n V_n U_n$, where $|U_n| = n$. The constant in the big-O only depends on $\mathbf{a}$.

Step 2. We compare the real number

$$\alpha = 0.a_1 a_2 a_3 \cdots$$

to the rational approximation

$$\alpha_n = 0.W_n U_n V_n U_n V_n U_n V_n \cdots, \tag{17.2}$$

which equals

$$\frac{p}{b^r(b^s-1)} \tag{17.3}$$

for some integers p, r, s. This rational approximation will turn out to be good but not, in general, good enough to conclude transcendence by Roth's theorem.

Step 3. We use a powerful theorem called the *Schmidt subspace theorem*, which relies on the special form of the denominator of the approximation (17.3) to conclude the transcendence of $\sum_{i\geq 0} a_i b^{-i}$.

We now give the details of each step.

17.2.1 Prefixes of automatic sequences

Consider an automatic sequence $\mathbf{a} = (a_i)_{i \geq 0}$ and consider the length-n factors appearing in it. Some factor f must appear infinitely often, so some prefix will contain two nonoverlapping occurrences of this factor.

We now argue that this prefix has length $\Theta(n)$. The lower bound is clear; it must be of length at least $2n$ since the two occurrences of f do not overlap. For the upper bound, it suffices to show that the function $s(n)$ defined by

$$s(n) = \text{length of the shortest prefix of } \mathbf{a} \text{ containing two nonoverlapping occurrences of some length-}n \text{ factor}$$

is $O(n)$.

To see this, note that $s(n)$ is k-synchronized in the sense of Lecture 12, because it is representable by the following first-order formula:

$$\exists i \ (i + 2n \leq s) \ \wedge \ \forall t \ (t < n) \implies \mathbf{a}[i+t] = \mathbf{a}[(s+t) - n]$$
$$\wedge \ (\forall t, u \ ((u \geq t + n) \ \wedge \ (\forall j \ (j < n) \implies \mathbf{a}[t+j] = \mathbf{a}[u+j])) \implies u + n \geq s).$$

The first line of this formula asserts that the factor ending at position $s-1$ has an earlier nonoverlapping occurrence, and the second line asserts that every factor having a second occurrence must have that second occurrence end at a position at or to the right of position $s-1$.

Since $(n, s(n))_k$ is k-synchronized, it follows from Theorem 91 that $s(n) = O(n)$. Hence $s(n) = \Theta(n)$. Say $s(n) \leq Cn$. Let $U(n)$ be the corresponding factor of length n. Then the prefix of length $s(n)$ of $(a(n))_{n \geq 0}$ looks like

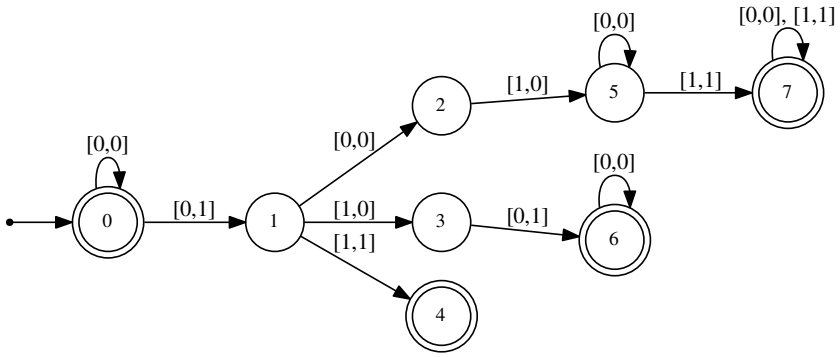
$$W_n U_n V_n U_n$$

where $|U_n| = n$ and $|W_n|, |V_n| \leq (C-2)n$.

Example 147. As an example, let us use `Walnut` to find a 2-DFAO recognizing $(n, s(n))_2$ for the case where $\mathbf{a} = \mathbf{t}$, the Thue-Morse sequence. We can use the `Walnut` command

```
def ss "(Ei (i+2*n<=s) & (At (t<n) => T[i+t]=T[(s+t)-n])) &
(At Au ((u>=t+n) & (Aj (j<n) => T[t+j]=T[u+j])) => (u+n >= s))":
```

It gives the following automaton



from which one can deduce that $s(n) \leq 4n$.

17.2.2 The rational approximation

Suppose α_n is defined as in Eq. (17.2). Let $w_n = |W_n|$, $v_n = |V_n| = n$, and $u_n = |U_n| = n$. Then we have

$$\alpha_n = \frac{p_n}{b^{w_n}(b^{u_n+v_n} - 1)} \tag{17.4}$$

for some integer p_n.

Also by comparing the base-b expansions

$$|\alpha - \alpha_n| < b^{-2u_n+v_n+w_n} \tag{17.5}$$

.

17.3 The Schmidt subspace theorem

We need some ideas from Diophantine approximation. A *linear form* is a linear combination of variables, such as $x_1 + 2x_2 - 3x_3$. We define $|x|_p = p^{-v_p(x)}$, where $v_p(x)$ is the exponent of the highest power of p dividing x. For example, $|48|_2 = 2^{-4}$.

This is due to Wolfgang Schmidt [40, 41, 42].

Theorem 148. *Let $m \geq 2$ be an integer. Let S be a finite set of prime numbers. Let*

$$L_{1,\infty}, L_{2,\infty}, \ldots, L_{m,\infty}$$

be m linearly independent forms with real algebraic coefficients. For a prime $p \in S$, let

$$L_{1,p}, L_{2,p}, \ldots, L_{m,p}$$

be m linearly independent linear forms with integer coefficients. Let $\epsilon > 0$ be a real number. Then there exists an integer t and t proper subspaces of $\mathbb{Q}^m$

$$s_1, s_2, \ldots, s_t \subset \mathbb{Q}^m$$

such that all the solutions

$$\mathbf{x} = (x_1, x_2, \ldots, x_m) \in \mathbb{Z}^m$$

to the inequality

$$\prod_{p \in S} \prod_{1 \le i \le m} |L_{i,p}(\mathbf{x})|_p \prod_{1 \le i \le m} |L_{i,\infty}(\mathbf{x})| \le (\max(1, |x_1|, \ldots, |x_m|))^{-\epsilon}$$

are contained in the union $S_1 \cup S_2 \cup \cdots \cup S_t$.

17.4 Proof of Theorem 143

It remains to see how to apply the Schmidt subspace theorem in our case.

Proof. Assume that $\alpha = a_0 + \sum_{i \ge 1} a_i b^{-i}$ is algebraic. Choose the following linear forms:

$$\begin{aligned} L_{1,\infty}(x_1, x_2, x_3) &= x_1 \\ L_{2,\infty}(x_1, x_2, x_3) &= x_2 \\ L_{3,\infty}(x_1, x_2, x_3) &= \alpha x_1 - \alpha x_2 - x_3 \end{aligned}$$

When we evaluate these forms on the vector

$$\mathbf{x}_n = (b^{u_n+v_n+w_n}, b^{w_n}, p_n),$$

we get

$$\begin{aligned} L_{1,\infty}(\mathbf{x}_n) &= b^{u_n+v_n+w_n} \\ L_{2,\infty}(\mathbf{x}_n) &= b^{w_n} \\ L_{3,\infty}(\mathbf{x}_n) &= b^{u_n+v_n+w_n}\alpha - b^{w_n}\alpha - p_n \end{aligned}$$

Now by Eq. (17.4), we have

$$-b^{w_n}(b^{u_n+v_n} - 1)\alpha_n + p_n = 0 \tag{17.6}$$

Hence

$$\begin{aligned} |L_{3,\infty}(\mathbf{x}_n)| &= |b^{u_n+v_n+w_n}\alpha - b^{w_n}\alpha - p_n| \\ &= |b^{u_n+v_n+w_n}\alpha - b^{w_n}\alpha - p_n - b^{w_n}(b^{u_n+v_n} - 1)\alpha_n + p_n| \quad \text{(by adding Eq. 17.6)} \\ &= |b^{w_n}(b^{u_n+v_n} - 1)(\alpha - \alpha_n)| \\ &\le |b^{u_n+v_n+w_n}||\alpha - \alpha_n| \\ &\le |b^{u_n+v_n+w_n}|b^{-(2u_n+v_n+w_n)} \quad \text{(by Eq. (17.5))} \\ &= b^{-u_n}. \end{aligned}$$

Hence

$$\prod_{1\le j\le 3} |L_{j,\infty}(\mathbf{x}_n)| \le b^{u_n+v_n+w_n} b^{w_n} b^{-u_n} = b^{v_n+2w_n}. \tag{17.7}$$

Let S be the set of all prime divisors of b. Let $p \in S$. Define the three forms

$$\begin{aligned} L_{1,p}(x_1, x_2, x_3) &= x_1 \\ L_{2,p}(x_1, x_2, x_3) &= x_2 \\ L_{3,p}(x_1, x_2, x_3) &= x_3. \end{aligned}$$

Then from the definition

$$\begin{aligned} \prod_{p\in S} |L_{1,p}(\mathbf{x}_n)|_p &= b^{-(u_n+v_n+w_n)} \\ \prod_{p\in S} |L_{2,p}(\mathbf{x}_n)|_p &= b^{-w_n} \\ \prod_{p\in S} |L_{3,p}(\mathbf{x}_n)|_p &\le 1 \end{aligned}$$

So, putting this together with Eq. (17.7), we get

$$\prod_{p\in S} \prod_{1\le i\le 3} |L_{i,p}(\mathbf{x}_n)|_p \prod_{1\le i\le 3} |L_{i,\infty}(\mathbf{x}_n)| \le b^{-u_n}.$$

Now from above we know $|W_n U_n V_n U_n| \le Cn$ and $|U_n| = n$, so we get

$$\begin{aligned} 2u_n + v_n + w_n &\le C_n = Cu_n \\ w_n &\le (C-2)u_n \\ p_n &= \alpha_n b^{w_n}(b^{u_n+v_n} - 1) \\ &\le \lceil \alpha \rceil b^{u_n+v_n+w_n} \\ &\le \lceil \alpha \rceil b^{(C-1)u_n}. \end{aligned}$$

So if we take $\epsilon < 1/c$ we have

$$b^{u_n} \ge \max(b^{u_n+v_n+w_n}, b^{w_n}, p_n)^{\epsilon},$$

and so

$$b^{-u_n} \le \max(b^{u_n+v_n+w_n}, b^{w_n}, p_n)^{-\epsilon}$$

for all n. So the hypotheses of Schmit's subspace theorem are satisfied, and the conclusion of the theorem is that the vectors $\mathbf{x}_n$ lie in a finite number of proper subspaces of $\mathbb{Q}^3$. By the infinite pigeonhole principle, at least one of these proper subspaces is of infinite cardinality.

Since it is a *proper* subspace of $\mathbb{Q}^3$, it must be a vector space of dimension ≤ 2, and so there must be a nontrivial linear relation of the form

$$z_1 b^{u_n+v_n+w_n} + z_2 b^{w_n} + z_3 p_n = 0 \tag{17.8}$$

holding for infinitely many n, for some integer constants z_1, z_2, z_3, not all 0.

Divide Eq. (17.8) by $b^{u_n+v_n+w_n}$ to get

$$z_1 + z_2 b^{-(u_n+v_n)} + z_3 p_n b^{-(u_n+v_n+w_n)} = 0 \tag{17.9}$$

for infinitely many n. From Eq. (17.4) we get

$$\frac{p_n}{b^{u_n+v_n+w_n}} = \alpha_n \frac{b^{u_n+v_n}-1}{b^{u_n+v_n}},$$

and so letting n increase without bound we see

$$\lim_{n\to\infty} \frac{p_n}{b^{u_n+v_n+w_n}} = \alpha.$$

Letting n increase without bound in Eq. (17.9), we get $z_1 + z_3\alpha = 0$. Thus either $\alpha = -z_1/z_3 \in \mathbb{Q}$, a contradiction, or $z_1 = z_3 = 0$. But then Eq. 17.8 implies that $z_2 = 0$, a contradiction.

Hence α must be transcendental. □

17.5 More recent progress

17.5.1 Automatic Liouville numbers

A *Liouville number* is a real transcendental number α such that $|\alpha - \frac{p}{q}| < q^{-r}$ has infinitely many solutions (p, q) for each $r \geq 2$. Adamczewski and Cassaigne [1] proved that no element of $L(k, b)$ is a Liouville number.

17.5.2 Irrationality measure of automatic real numbers

Computing the exact value of the irrationality measure of a given real number α is, in general, an extremely hard problem. For example, currently we only know that $\mu(\pi) \leq 7.6064$. Liouville numbers x have $\mu(x) = \infty$. Almost all real numbers have $\mu(x) = 2$.

It follows from [43] that the automatic real numbers $\sum_{i\geq 0} b^{-2^i} \in L(2, b)$ have irrationality measure 2.

Bugeaud [11] proved that $\mu(t) = 2$, where $t = \sum_{i\geq 0} t_i b^{-i-1}$ is the Thue-Morse real number in base b.

17.5.3 Other kinds of expansions

Similarly one can examine different kinds of expansions, other than base b.

Bugeaud [12] proved that if a real number has an infinite continued fraction given by a k-automatic sequence taking values in $\{1, 2, \ldots d\}$ for some d, then it must either be a quadratic irrational or transcendental.

17.6 Open problems

1. Are any of the "classical" transcendental numbers, like $e, \pi, \log 2$, etc. k-automatic real numbers. Probably not, but currently nobody knows how to prove this.

2. What is $L(k, b) \cap L(k, b')$ where $b, b' \geq 2$ are multiplicatively independent? We expect it is $\mathbb{Q}$, but currently nobody knows how to prove this.

Chapter 18

Sturmian sequences

Throughout this lecture, all the infinite words are indexed starting at position 1, as is the convention for Sturmian words.

Let θ, rho be real numbers with $0 < \theta, rho < 1$. Define

$$s_n = \lfloor (n+1)\theta + \rho \rfloor - \lfloor n\theta + \rho \rfloor$$
$$s'_n = \lceil (n+1)\theta + \rho \rceil - \lceil n\theta + \rho \rceil$$

for $n \geq 1$. The words $(s_n)_{n\geq 0}$ and $(s'_n)_{n\geq 0}$ are called the *Sturmian words* of *slope* θ and *intercept* ρ. Notice that $s_n \in \{0, 1\}$.

A special case of $(s_n)_{n\geq 0}$ is where $\rho = 0$, and is easiest to work with. These are called the *Sturmian characteristic words*, and are written as $\mathbf{f}_\theta$.

Example 149. Take $\theta = (\sqrt{5} - 1)/2 \doteq 0.61803\cdots$. Then

$$\mathbf{f}_\theta := f_\theta(1) f_\theta(2) \cdots = 10110101\cdots$$

We will see later that this is (up to renaming of the letters) the infinite Fibonacci word discussed previously.

18.1 Basic results

Define the coding $r(0) = 1$ and $r(1) = 0$.

Theorem 150. *If $0 < \theta < 1$ is an irrational real, then $\mathbf{f}_{1-\theta} = r(\mathbf{f}_\theta)$.*

Proof. We have

$$f_\theta(n) = \lfloor (n+1)\theta \rfloor - \lfloor n\theta \rfloor$$

and

$$\begin{aligned} f_{1-\theta}(n) &= \lfloor (n+1)(1-\theta) \rfloor - \lfloor n(1-\theta) \rfloor \\ &= \lfloor -(n+1)\theta \rfloor + n + 1 - \lfloor -n\theta \rfloor - n \\ &= \lfloor -(n+1)\theta \rfloor - \lfloor -n\theta \rfloor + 1. \end{aligned}$$

So

$$\begin{aligned} f_\theta(n) + f_{1-\theta}(n) &= \lfloor (n+1)\theta \rfloor + \lfloor -(n+1)\theta \rfloor - \lfloor n\theta \rfloor - \lfloor -n\theta \rfloor + 1 \\ &= \lfloor x \rfloor + \lfloor -x \rfloor - (\lfloor y \rfloor + \lfloor -y \rfloor) + 1, \quad \text{where } x = (n+1)\theta \text{ and } y = n\theta \\ &= (-1) - (-1) + 1 \\ &= 1, \end{aligned}$$

because x and y are irrational. The result follows. □

We now introduce a related infinite word:

$$\mathbf{g}_\alpha := g_\alpha(1) g_\alpha(2) \cdots$$
$$g_\alpha(n) = \begin{cases} 1, & \text{if } n = \lfloor k\alpha \rfloor \text{ for some } k, \\ 0, & \text{otherwise.} \end{cases}$$

Theorem 151. *Let $\alpha > 1$ be an irrational real number. Then*

$$\mathbf{g}_\alpha = \mathbf{f}_{1/\alpha}.$$

Proof.

$$\begin{aligned} g_\alpha(n) = 1 &\iff \exists k \text{ such that } n = \lfloor k\alpha \rfloor \\ &\iff \exists k \text{ such that } n \le k\alpha < n+1 \\ &\iff \exists k \text{ such that } n/\alpha \le k\alpha < (n+1)/\alpha \\ &\iff \exists k \text{ such that } \lfloor n/\alpha \rfloor = k-1 \text{ and } \lfloor (n+1)/\alpha \rfloor = k \\ &\iff \lfloor (n+1)/\alpha \rfloor - \lfloor n/\alpha \rfloor = 1 \\ &\iff f_{1/\alpha}(n) = 1. \end{aligned}$$

□

We now define the *Sturmian characteristic morphisms* $h_n : \{0,1\}^* \to \{0,1\}^*$ for $n \ge 1$.

$$h_n(0) = 0^{n-1}1$$
$$h_n(1) = 0^{n-1}10$$

Theorem 152. *Let α be an irrational real, $0 < \alpha < 1$, and let $k \ge 1$ be an integer. Then $h_k(\mathbf{f}_\alpha) = \mathbf{f}_{1/(k+\alpha)}$.*

Proof. Define $d_i = h_k(f_\alpha(i))$ for $i \ge 1$. So

$$h_k(\mathbf{f}_\alpha) = d_1 d_2 d_3 \cdots .$$

Let n be the position of the m'th 1 in $h_k(\mathbf{f}_\alpha)$. Each d_i contains exactly one 1, so this means we are interested in the 1 appearing in d_m. Then

$$\begin{aligned}|d_1 d_2 \cdots d_{m-1}| &= (m-1)k + f_\alpha(1) + \cdots f_\alpha(m-1) \\ &= (m-1)k + (\lfloor 2\alpha \rfloor - \lfloor \alpha \rfloor) + (\lfloor 3\alpha \rfloor - \lfloor 2\alpha \rfloor) + \cdots + (\lfloor m\alpha \rfloor - \lfloor (m-1)\alpha \rfloor) \\ &= (m-1)k + \lfloor m\alpha \rfloor - \lfloor \alpha \rfloor \\ &= (m-1)k + \lfloor m\alpha \rfloor.\end{aligned}$$

Hence

$$\begin{aligned}n &= |d_1 d_2 \cdots d_{m-1}| + k \\ &= (m-1)k + \lfloor m\alpha \rfloor + k \\ &= \lfloor m(k+\alpha) \rfloor.\end{aligned}$$

And so

$$\begin{aligned}(h_k(\mathbf{f}_\alpha))(n) = 1 &\iff \exists m \text{ such that} n = \lfloor m(k+\alpha) \rfloor \\ &\iff g_{k+\alpha}(n) = 1 \\ &\iff f_{1/(k+\alpha)}(n) = 1.\end{aligned}$$

□

Now an easy induction proves

Theorem 153. *Let α, $0 < \alpha < 1$ have continued fraction expansion $\alpha = [0, a_1, a_2, \ldots]$. Define $\beta_n = [0, a_n, a_{n+1}, \ldots]$ for $n \geq 1$. Then*

$$\mathbf{f}_\alpha = (h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_n})(\mathbf{f}_{\beta_{n+1}}).$$

Corollary 154. *If $\alpha = [0, \overline{a_1, a_2, \ldots, a_n}]$, then $\mathbf{f}_\alpha$ is a fixed point of $h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_n}$.*

Example 155. Take $\alpha = [0, \overline{1}] = [0, 1, 1, 1, \ldots] = \frac{\sqrt{5}-1}{2}$. Then $f_\alpha = 10110\cdots$ is a fixed point of the morphism $1 \to 10$ and $0 \to 1$.

18.2 Sturmian characteristic words

Let $\alpha = [0, a_1, a_2, \ldots]$ be a real number. The *Sturmian characteristic words* are defined by

$$\begin{aligned}X_n &= (h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_n})(0) \\ Y_n &= (h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_n})(1).\end{aligned}$$

Proposition 156. *For $n \geq 1$ we have $Y_n = X_n X_{n-1}$.*

Proof. We have

$$\begin{aligned}
Y_n &= (h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_n})(1) \\
&= (h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_{n-1}})(h_{a_n}(1)) \\
&= (h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_{n-1}})(h_{a_n}(0)0) \\
&= (h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_{n-1}})(h_{a_n}(0))(h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_{n-1}})(0) \\
&= (h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_n})(0)(h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_{n-1}})(0) \\
&= X_n X_{n-1}.
\end{aligned}$$

□

Theorem 157. *We have*

$$X_n = \begin{cases} 0, & \text{if } n = 0; \\ 0^{a_1-1}1, & \text{if } n = 1; \\ X_{n-1}^{a_n} X_{n-2}, & \text{if } n \geq 2. \end{cases}$$

Proof. By induction on n. It is easy to check for $n = 0, 1$. Now assume the result is true for $n' < n$ for $n \geq 2$. We prove it for n. We have

$$\begin{aligned}
X_n &= (h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_n})(0) \\
&= (h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_{n-1}})(h_{a_n}(0)) \\
&= (h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_{n-1}})(0^{a_n-1}1) \\
&= (h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_{n-1}})(0^{a_n-1})(h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_{n-1}})(1) \\
&= ((h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_{n-1}})(0))^{a_n-1}(h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_{n-1}})(1) \\
&= X_{n-1}^{a_n-1} X_{n-1} X_{n-2} \quad \text{(by Prop. (156))} \\
&= X_{n-1}^{a_n} X_{n-2}.
\end{aligned}$$

□

By comparing this recursion for X_n with the recursions in Lecture 5, we see that we have found an analogue, in words, of continued fractions!

Example 158. We have

$$e^{-1} = [0, 2, 1, 2, 1, 1, 4, \ldots],$$

and so

n	-2	-1	0	1	2	3	4	5	6
a_n			0	2	1	2	1	1	4
p_n	0	1	0	1	1	3	4	7	32
q_n	1	0	1	2	3	8	11	19	87

We find

$$\begin{aligned} X_0 &= 0 \\ X_1 &= 01 \\ X_2 &= 010 \\ X_3 &= 01001001 \\ X_4 &= 01001001010 \\ X_5 &= 0100100101001001001, \end{aligned}$$

etc.

Theorem 159. *For $n \geq 0$ we have*

(a) $|X_n| = q_n$ and $|X_n|_1 = p_n$;

(b) $|Y_n| = q_n + q_{n-1}$ and $|Y_n|_1 = p_n + p_{n-1}$.

Proof. By Proposition 156 it suffices to prove (a). This is an easy induction on n (omitted). □

Theorem 160. *For $n \geq 1$ the word X_n is the prefix of f_α of length q_n.*

Proof. Since

$$\mathbf{f}_\alpha = (h_{a_1} \circ \cdots \circ h_{a_n})(\mathbf{f}_{\beta_{n+1}}),$$

we see that $\mathbf{f}_\alpha$ has a prefix of either

$$X_n = (h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_n})(0)$$

or

$$Y_n = (h_{a_1} \circ h_{a_2} \circ \cdots \circ h_{a_n})(1).$$

But $Y_n = X_n X_{n-1}$. So X_n is a prefix of $\mathbf{f}_\alpha$ and by Theorem 159 we have $|X_n| = q_n$. □

The next theorem proves the "almost-commutative" property of Sturmian characteristic words. Define $c(w) = w'ba$ if $w = w'ab$ for letters a, b.

Theorem 161. *For $n \geq 1$ we have $X_n X_{n-1} = c(X_{n-1} X_n)$.*

Proof. By induction on n. For $n = 1$ we have

$$X_1 X_0 = 0^{a_1 - 1}10 = c(0^{a_1 - 1}01) = c(00^{a_1 - 1}1) = c(X_0 X_1).$$

Assume the result is true for $n' < n$. Then

$$\begin{aligned} X_n X_{n-1} &= (X_{n-1}^{a_n} X_{n-2}) X_{n-1} \\ &= X_{n-1}^{a_n} c(X_{n-1} X_{n-2}) \quad \text{(by induction)} \\ &= c(X_{n-1}^{a_n} X_{n-1} X_{n-2}) \\ &= c(X_{n-1} X_{n-1}^{a_n} X_{n-2}) \\ &= c(X_{n-1} X_n). \end{aligned}$$

□

18.3 The Ostrowski numeration system

Given an irrational real $\alpha = [a_0, a_1, \ldots,]$ with convergents $p_n/q_n = [a_0, a_1, \ldots, a_n]$, we can write every integer $N \geq 0$ uniquely as $N = \sum_{0 \leq i \leq j} b_i q_i$ where the digits (b_i) satisfy

(a) $0 \leq b_0 < a_1$;

(b) $0 \leq b_i \leq a_{i+1}$; for $i \geq 1$

(c) For $i \geq 1$, if $b_i = a_{i+1}$, then $b_{i-1} = 0$.

Example 162. Let $\alpha = \sqrt{2} = [1, 2, 2, 2, \ldots]$. Then

n	-2	-1	0	1	2	3	4	5
a_n			1	2	2	2	2	2
p_n	0	1	1	3	7	17	41	99
q_n	1	0	1	2	5	12	29	70

The Ostrowski numeration system corresponding to these (q_n) is the so-called Pell numeration system and is written $(N)_p$.

Here are the first few representations in this system.

N	$(N)_p$	$f_{\alpha-1}(N)$
1	1	0
2	10	1
3	11	0
4	20	1
5	100	0
6	101	0
7	110	1
8	111	0
9	120	1
10	200	0
11	201	0
12	1000	1

The Ostrowski numeration system permits factorization of the prefix of length m of $\mathbf{f}_\alpha$.

Theorem 163. *Let $0 < \alpha < 1$ and $\alpha = [0, a_1, a_2, \ldots]$. Let $m \geq 0$ and let $b_s b_{s-1} \cdots b_0$ be the Ostrowski representation of m, msd first. Then*

$$f_\alpha(1) f_\alpha(2) \cdots f_\alpha(m) = X_s^{b_s} X_{s-1}^{b_{s-1}} \cdots X_0^{b_0}. \tag{18.1}$$

Proof. By induction on m. For $m = 0$ both sides are ϵ. If $0 < m < q_1 = a_1$, then $b_0 = m$, $X_0 = 0$, and

$$f_\alpha(1) f_\alpha(2) \cdots f_\alpha(q_1) = 0^{a_1 - 1}1,$$

so $f_\alpha(1)f_\alpha(2)\cdots f_\alpha(m) = X_0^{b_0}$.

Now let $s \geq 1$. Suppose Eq. (18.1) holds for all $m < q_s$. We prove it for $m < q_{s+1}$. Suppose $q_s \leq m < q_{s+1}$. Write $m = b_s q_s + r$, where $1 \leq b_s \leq a_{s+1}$, $0 \leq r < q_s$. By induction we have

$$\begin{aligned} r &= \sum_{0 \leq i < s} b_i q_i \\ f_\alpha(1)f_\alpha(2)\cdots f_\alpha(r) &= X_{s-1}^{b_{s-1}} \cdots X_0^{b_0}. \end{aligned}$$

Case 1: $b_s < a_{s+1}$. Then $f_\alpha(1)f_\alpha(2)\cdots f_\alpha(m)$ is a prefix of $X_{s+1} = X_s^{a_{s+1}} X_{s-1}$. But $m = b_s q_s + r$, so $m < q_{s+1}$. Hence $f_\alpha(1)f_\alpha(2)\cdots f_\alpha(m)$ is a prefix of $X_s^{b_s+1} = X_s^{b_s} X_s$. Then

$$\begin{aligned} f_\alpha(1)f_\alpha(2)\cdots f_\alpha(m) &= X_s^{b_s} f_\alpha(1)f_\alpha(2)\cdots f_\alpha(r) \\ &= X_s^{b_s} X_{s-1}^{b_{s-1}} \cdots X_0^{b_0}. \end{aligned}$$

Case 2: $b_s = a_{s+1}$. We have $m < q_{s+1}$ and so $m = b_s q_s + r = a_{s+1} q_s + r < q_{s+1}$. Thus $r < q_{s-1}$. Hence $f_\alpha(1)f_\alpha(2)\cdots f_\alpha(r)$ is a prefix of X_{s-1}. But $f_\alpha(1)f_\alpha(2)\cdots f_\alpha(m)$ is a prefix of $X_{s+1} = X_s^{a_{s+1}} X_{s-1}$. So

$$\begin{aligned} f_\alpha(1)f_\alpha(2)\cdots f_\alpha(m) &= X_s^{a_{s+1}} f_\alpha(1)f_\alpha(2)\cdots f_\alpha(r) \\ &= X_s^{b_s} X_{s-1}^{b_{s-1}} \cdots X_0^{b_0}. \end{aligned}$$

□

We can now characterize when $f_\alpha(n) = 1$, in terms of the Ostrowski representation of n.

Theorem 164. *$f_\alpha(n) = 1$ iff $b_s b_{s-1} \cdots b_0$, the Ostrowski representation of n, ends in an odd number of zeros.*

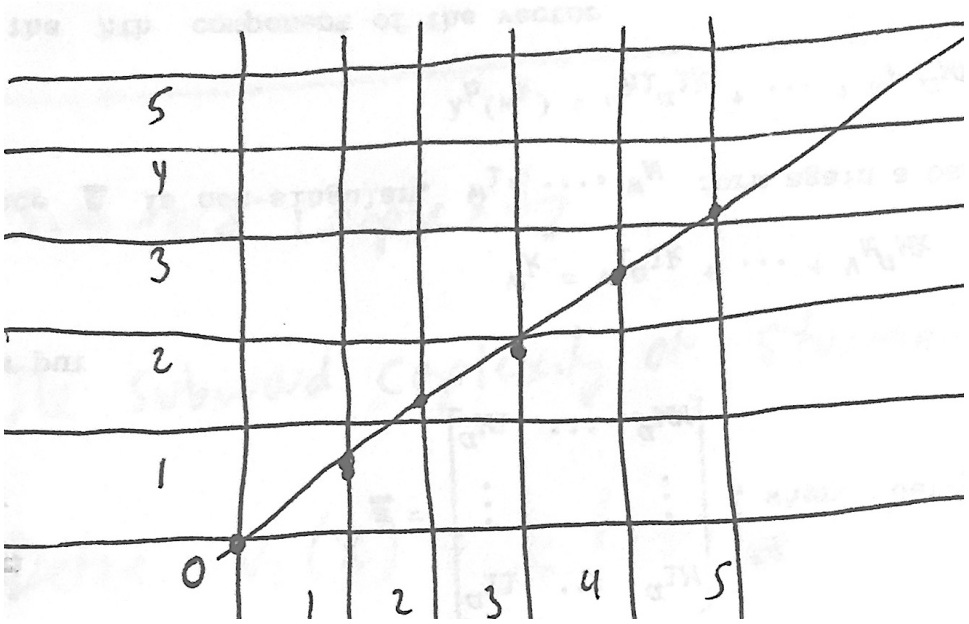
Proof. We have

$$\begin{aligned} f_\alpha(q_n) &= \text{last symbol of } X_n \\ &= \begin{cases} 1, & \text{if } n \text{ is odd;} \\ 0, & \text{if } n \text{ is even.} \end{cases} \end{aligned}$$

Let i be the least index for which $b_i > 0$. Then $f_\alpha(q_n)$ is the last symbol of X_i, which is the last symbol of $f_\alpha(q_i)$, which is 1 if i is odd, iff $b_{i-1} \cdots b_0 = 0^i$ is an odd number of 0's. □

18.4 Geometric interpretation of Sturmian words

Let $\theta > 0$ be an irrational real number, and consider the line $y = \theta x$ through the origin with slope θ. For example, if $\theta = (\sqrt{5}-1)/2$, we have the following picture:



As we travel up and to the left, let us write a 0 or a 1 for each intersection the line makes with the integer grid, as follows:

$$c_i = \begin{cases} 0, & \text{if } L \text{ intersects a vertical line;} \\ 1, & \text{if } L \text{ intersects a horizontal line.} \end{cases}$$

and define $\mathbf{c}_\theta = c_1 c_2 c_3 \cdots$. For this particular value of θ we get $c_\theta = 01001010\cdots$.

Assume $0 < \theta < 1$. As x increase from n to $n+1$, we get

- an extra 0 if $\lfloor (n+1)\theta \rfloor = \lfloor n\theta \rfloor$;
- an extra 10 if $\lfloor (n+1)\theta \rfloor > \lfloor n\theta \rfloor$.

Define $h(0) = 0$, $h(1) = 10$ and $h'(0) = 0$, $h'(1) = 01$. Then $c_\theta = h(0\mathbf{f}_\theta) = h'(\mathbf{f}_\theta)$. So $h' = r \circ h_1$, where $r(0) = 1$, $r(1) = 0$. So $\mathbf{c} = r(h_1(\mathbf{f}_\theta)) = \mathbf{f}_\alpha$, where $\alpha = 1 - 1/(1+\theta) = \theta/(\theta+1)$.

18.5 Subword complexity

The main result is

Theorem 165. *Let α be irrational. The subword complexity of the Sturmian word $\mathbf{s}_{\alpha,\theta}$ is $n+1$.*

Proof. Define $v_i(x) = \lfloor (i+1)\alpha + x \rfloor - \lfloor i\alpha + x \rfloor$. Clearly $v_i(x)$ is periodic of period 1 in x.

If $\mathbf{s}_{\alpha,\theta} = s_1 s_2 s_3 \cdots$, then

$$v_i(j\alpha + \theta) = s_{i+j}.$$

Now consider the $n+2$ numbers

$$0, \{-\alpha\}, \{-2\alpha\}, \ldots, \{-n\alpha\}, 1,$$

where by $\{x\}$ we mean the fractional part of x, i.e., x mod 1. and arrange them in increasing order:

$$0 = c_0(n) < c_1(n) < \cdots < c_n(n) < c_{n+1}(n) = 1.$$

Define the half-open interval $L_j = [c_j(n), c_{j+1}(n))$ for $0 \le j \le n$. We now claim that $v_n(x) = v_0(x)v_1(x)\cdots v_{n-1}(x)$ is constant on the interval $L_j(n)$. To see this, it suffices to show that for all i, $0 \le i < n$, the quantity $i\alpha + x$ is never an integer for $c_j(n) < x < c_{j+1}(n)$.

Suppose there exists x_0, $0 < x_0 < 1$, such that $i\alpha + x_0 = r$, and $c_j(n) < x_0 < c_{j+1}(n)$ for some integer r. Then $-i\alpha = x_0 - r$, so $\{-i\alpha\} = \{x_0\}$. But then this contradicts the definition of $[c_j(n), c_{j+1}(n))$. Define $B_j(n) = v_0(x)\cdots v_{n-1}(x)$ for $x \in L_j(n)$. This proves the claim about $v_n(x)$.

Thus we have shown that if $w = \mathbf{s}_{\alpha,\theta}[m..m+n-1]$ is a length-n subword, we have $w = v_0(x)v_1(x)\cdots v_{n-1}(x)$ for $m = \{m\alpha + \theta\}$. So $w = B_j(n)$ for some j, $0 \le j \le n$, where $x = L_j(n)$. So the subword complexity of length-n words is at most $n+1$ for all $n \ge 0$. But if the subword complexity is $\le n$ for any n, then the word must be ultimately periodic. So the 1's in $\mathbf{s}_{\alpha,\theta}$ have rational density. But the density of 1's in $\mathbf{s}_{\alpha,\theta}$ is α, which is irrational, a contradiction. □

18.6 Notes

A good reference on this topic is the chapter on Sturmian words in [35].

18.7 Exercises

1. Define a sequence of linear polynomials as follows:

$$\begin{aligned} g_1(x) &= x/2, \\ g_2(x) &= (x+1)/2 \\ g_n(x) &= g_{n-1}(g_{n-2}(x)) \quad (n \ge 3). \end{aligned}$$

Find an expression for $g_n(x)$ and for $\lim_{n\to\infty} g_n(0)$.

Chapter 19

Fibonacci and Tribonacci number systems

We can extend the notion of k-automatic, k-synchronized, and k-regular sequences to other numeration systems, such as the Fibonacci numbers and Tribonacci numbers.

What ingredients do we need?

- A method to represent elements of $\mathbb{N}$ as strings
- An automaton to test equality of two such representations (easiest thing: have a notion of *canonical* expansion)
- An "adder": an automaton to test the proposition $x + y = z$

Recall the Fibonacci numbers: $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$. In analogy with base-2 representation, we can represent every non-negative integer in the form

$$\sum_{0 \le i \le t} \epsilon_i F_{i+2} \quad \text{with} \quad \epsilon_i \in \{0, 1\}.$$

To get unique expansions, we impose the additional condition that $\epsilon_i \epsilon_{i+1} = 0$ for all i: never use two adjacent Fibonacci numbers. Usually we write the representation in the form

$$\epsilon_t \epsilon_{t-1} \cdots \epsilon_0,$$

with most significant digit first. So, for example, 19 is represented by 101001. This is called *Fibonacci representation* or *Zeckendorf representation*.

19.1 Fibonacci-automatic infinite words

- Consider a finite automaton that takes Fibonacci representation of n as input
- Outputs are associated with the last state reached

- Invalid inputs (those with two consecutive 1's) are rejected or not considered
- An infinite word results from feeding the canonical representation of each $n \geq 0$ into the automaton
- Example: the Fibonacci infinite word

$$\mathbf{f} = 0100101001001\cdots$$

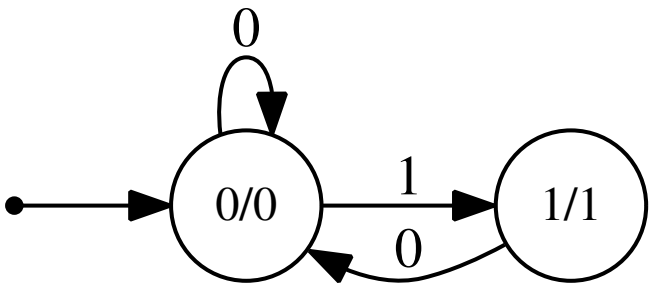
19.2 The Fibonacci decision procedure

- Exactly like before, except now all integers are represented in Fibonacci representation
- Comparison is easy
- Addition is harder; need an adder
- There is a 17-state automaton that on input (x, y, z) in Fibonacci representation will determine whether $x + y = z$
- Based on ideas originally due to Jean Berstel and since elaborated by others: Frougny, Sakarovitch, etc.

The most famous Fibonacci-automatic word is the Fibonacci word

$$\mathbf{f} = 0100101001001010010100100101001001\cdots,$$

which can be defined in various ways. One way is the fixed point of the morphism $\varphi(0) = 01$, $\varphi(1) = 0$. Another way is the automaton



Yet another way is through the recursion

$$\begin{aligned} X_1 &= 1 \\ X_2 &= 0 \\ X_n &= X_{n-1}X_{n-2}, \quad (n \geq 2) \end{aligned}$$

So $X_3 = 01$, $X_4 = 010$, $X_5 = 01001$, etc. Note that $|X_n| = F_n$.

The $(X_n)_{n\geq 1}$ are called the *finite Fibonacci words*, and for $n \geq 2$ they are all prefixes of $\mathbf{f}$. Properties of the infinite Fibonacci word $\mathbf{f}$ have been widely studied, e.g.:

- $\mathbf{f}$ is not ultimately periodic
- $\mathbf{f}$ contains no 4th powers (Karhumäki, 1983)
- All squares in $\mathbf{f}$ are of order F_n for $n \geq 2$, and squares of all these lengths exist (Séébold, 1985)
- There exist palindromes of all lengths in $\mathbf{f}$ (Chuan, 1993)

All of these claims can easily be verified using our method.

19.3 An extended example: avoiding the pattern xxx^R

Recall that by x^R we mean the reversal of the string x. For example, $(\texttt{stressed})^R = \texttt{desserts}$ in English; $(\texttt{relativ})^R = \texttt{vitaler}$ in German. We are interested in avoiding the pattern xxx^R in binary words.

An example of the pattern xxx^R in English is contained in the word

$$\texttt{bepepper}.$$

Examples in German: Wiedererreichen (reattainment) and besessen (obsessed). Are there infinite binary words avoiding this pattern?

An experimental approach: we start by trying depth-first search of the space of binary words If there is a word avoiding the pattern, this procedure will give the lexicographically least such sequence. When we do, we get the word

$$(001)^3(10)^\omega = 001001001101010\cdots .$$

So in particular the word $(10)^\omega = 101010\cdots$ avoids the pattern. (Easy proof!)

This suggests the following question: are there any *other* periodic infinite words avoiding xxx^R? Also: are there any *aperiodic* infinite words avoiding xxx^R?

When we search for other primitive words z such that z^ω avoids the pattern, we find there are some of length 10:

```
0010011011
0011011001
0100110110
0110010011
0110110010
1001001101
1001101100
1011001001
1100100110
1101100100
```

We notice that each of these words is of the form $w\overline{w}$. This suggests looking at words of this form. The next ones are $w = 001001001101100100100$, and its shifts and complements.

To summarize, here are the solutions we've found so far and their lengths. They are words of the form $w\overline{w}$:

w	$\lvert w\rvert$
1	1
00100	5
001001001101100100100	21

The presence of the numbers 1,5,21 suggests some connection with the Fibonacci numbers: these are F_2, F_5, F_8.

Suppose we take the run-length encodings of the strings of length 21. One of them looks familiar: 2122121221221. This is a prefix of the infinite Fibonacci word generated by $2 \to 21$, $1 \to 2$.

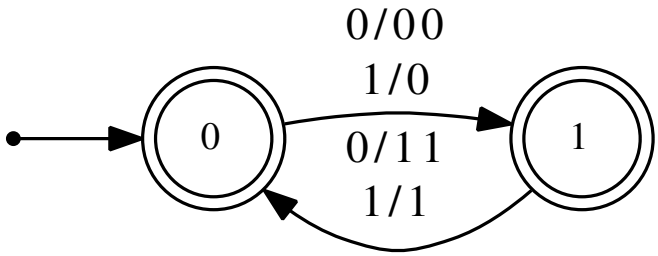
This suggests the construction of an *infinite* aperiodic word avoiding xxx^R: take the infinite Fibonacci word, and use it as "repetition factors" for 0 and 1 alternating. This gives the infinite word

$$\mathbf{R} = 0010011011011001001 10 \cdots$$

which we conjecture avoids xxx^R.

Can we find an automaton generating this sequence? Yes, but now it is not based on base-2 representations, but rather Fibonacci (or "Zeckendorf") representations.

Another way to describe the word $\mathbf{R}$ is as follows: Take the infinite Fibonacci word $\mathbf{f}$ and run it through the following transducer:

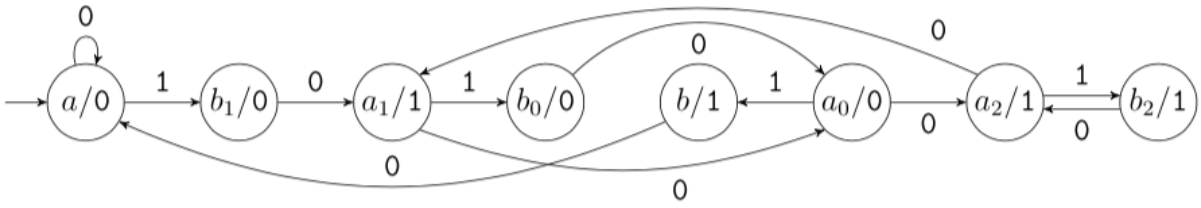


obtaining the infinite word

$$\mathbf{R} = 00100110110110010011011011001001001101100100100110 0 \cdots .$$

Claim: it avoids the patterns xxx^R and also xx^Rx^R.

We can try to find an automaton for $\mathbf{R}$ using a "guess and test" procedure. When we do, we get the following automaton of 8 states.



We now have the conjecture that the word generated by this automaton (a) is aperiodic and (b) avoids xxx^R and (c) avoids xx^Rx^R. All three conjectures can be proved using our decision procedure; we just need to write predicates for them:

- Ultimate periodicity:

$$\exists p \geq 1 \ \exists N \geq 0 \ \forall i \geq N \ \mathbf{R}[i] = \mathbf{R}[i+p].$$

- Has xxx^R:

$$\exists i \geq 0 \ \exists n \geq 1 \ \forall t < n$$
$$(\mathbf{R}[i+t] = \mathbf{R}[i+t+n]) \ \wedge \ (\mathbf{R}[i+t] = \mathbf{R}[i+3n-1-t]).$$

- Has xx^Rx^R:

$$\exists i \geq 0 \ \exists n \geq 1 \ \forall t < n$$
$$(\mathbf{R}[i+t] = \mathbf{R}[i+2n-1-t]) \ \wedge \ (\mathbf{R}[i+n+t] = \mathbf{R}[i+2n+t]).$$

Using `Walnut`, we can prove

Theorem 166. *The Fibonacci-automatic word* $\mathbf{R}$ *generated by the automaton above is*

(a) aperiodic and

(b) has no instances of the pattern xxx^R *for* x *nonempty and*

(c) also has no instances of the pattern xx^Rx^R *for* x *nonempty.*

19.4 Theorems about the finite Fibonacci words

Since every finite Fibonacci word is a prefix of length F_n of the infinite Fibonacci word, we can rephrase many claims about the finite Fibonacci words in terms of our logical language. There are two possible approaches: we can state these claims for length-n prefixes and ask for which n they are satisfied. Or we can additionally restrict n in our logical language to have Fibonacci representation of the form 10^*.

To illustrate this idea, consider one of the most famous properties of the Fibonacci words, the *almost-commutative* property: letting $\eta(a_1a_2\cdots a_n) = a_1a_2\cdots a_{n-2}a_na_{n-1}$ be the map that interchanges the last two letters of a string of length at least 2, we have

Theorem 167. *$X_{n-1}X_n = \eta(X_nX_{n-1})$ for $n \geq 2$.*

We can verify this, and prove even more, using our method.

Theorem 168. *Let $x = \mathbf{f}[0..i-1]$ and $y = \mathbf{f}[0..j-1]$ for $i > j > 1$. Then $xy = \eta(yx)$ if and only if $i = F_n$, $j = F_{n-1}$ for $n \geq 3$.*

Proof. The idea is to check, for each $i > j > 1$, whether

$$\mathbf{f}[0..i-1]\mathbf{f}[0..j-1] = \eta(\mathbf{f}[0..j-1]\mathbf{f}[0..i-1]).$$

We can do this with the following formula:

$$\begin{aligned}(i > j) \ \wedge \ (j \geq 2) \ \wedge \ (\forall t, \ j \leq t < i, \ \mathbf{f}[t] = \mathbf{f}[t-j]) \ \wedge \\ (\forall s \leq j-3 \ \mathbf{f}[s] = \mathbf{f}[s+i-j]) \ \wedge \ (\mathbf{f}[j-2] = \mathbf{f}[i-1]) \ \wedge \ (\mathbf{f}[j-1] = \mathbf{f}[i-2]).\end{aligned}$$

The resulting automaton accepts $[1,0][0,1][0,0]^+$, which corresponds to $i = F_n$, $j = F_{n-1}$ for $n \geq 4$. □

In many cases we can count the number $T(n)$ of length-n factors of a Fibonacci-automatic sequence having a particular property P. Here by "count" we mean, give an algorithm A to compute $T(n)$ efficiently, that is, in time bounded by a polynomial in $\log n$. Although *finding* the algorithm A may not be particularly efficient, once we have it, we can compute $T(n)$ quickly.

19.5 Reproving (and fixing) a result of Fraenkel and Simpson

We turn to a result of Fraenkel and Simpson [28]. They computed the exact number of occurrences of all squares appearing in the finite Fibonacci words X_n.

To solve this using our approach, we generalize the problem to consider *any* length-n prefix of $\mathbf{f}$. The total number of square occurrences in $\mathbf{f}[0..n-1]$:

$$L_{\text{dos}} := \{(n,i,j)_F \ : \ i + 2j \leq n \text{ and } \mathbf{f}[i..i+j-1] = \mathbf{f}[i+j..i+2j-1]\}.$$

Let $b(n)$ denote the number of occurrences of squares in $\mathbf{f}[0..n-1]$. First, we use our method to find a DFA M accepting L_{dos}. This (incomplete) DFA has 27 states.

Next, we compute matrices M_0 and M_1, indexed by states of M, such that $(M_a)_{k,l}$ counts the number of edges (corresponding to the variables i and j) from state k to state l on the digit a of n. We also compute a vector u corresponding to the initial state of M and a vector v corresponding to the final states of M. This gives us the following linear representation of the sequence $b(n)$:

if $x = a_1a_2\cdots a_t$ is the Fibonacci representation of n, then

$$b(n) = uM_{a_1}\cdots M_{a_t}v, \tag{19.1}$$

which, incidentally, gives a fast algorithm for computing $b(n)$ for any n.

Now let $B(n)$ denote the number of square occurrences in the finite Fibonacci word X_n.

This corresponds to considering the Fibonacci representation of the form 10^{n-1}; that is, $B(n+1) = b([10^n]_F)$.

The matrix M_0 is the following 27×27 array

$$\begin{bmatrix}
1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0
\end{bmatrix}$$

The matrix M_0 has minimal polynomial

$$X^4(X-1)^2(X+1)^2(X^2-X-1)^2.$$

It follows from the theory of linear recurrences that there are constants $c_1, c_2, \ldots, c_8$ such that

$$B(n+1) = (c_1 n + c_2)\alpha^n + (c_3 n + c_4)\beta^n + c_5 n + c_6 + (c_7 n + c_8)(-1)^n$$

for $n \geq 3$, where $\alpha = (1+\sqrt{5})/2$, $\beta = (1-\sqrt{5})/2$ are the roots of $X^2 - X - 1$. We can find these constants by computing $B(4), B(5), \ldots, B(11)$ and then solving for the values of the constants $c_1, \ldots, c_8$.

When we do so, we find

$$\begin{aligned}
c_1 &= \frac{2}{5} & c_2 &= -\frac{2}{25}\sqrt{5} - 2 & c_3 &= \frac{2}{5}\\
c_4 &= \frac{2}{25}\sqrt{5} - 2 & c_5 &= 1 & c_6 &= 1\\
c_7 &= 0 & c_8 &= 0
\end{aligned}$$

A little simplification, using the fact that $F_n = (\alpha^n - \beta^n)/(\alpha - \beta)$, leads to

Theorem 169. *Let $B(n)$ denote the number of square occurrences in X_n. Then*

$$B(n+1) = \frac{4}{5}nF_{n+1} - \frac{2}{5}(n+6)F_n - 4F_{n-1} + n + 1$$

for $n \geq 3$.

This statement corrects a small error in their paper.

19.6 Counting cube occurrences in finite Fibonacci words

In a similar way, we can count the cube occurrences in X_n. Using analysis exactly like the square case, we easily find

Theorem 170. *Let $C(n)$ denote the number of cube occurrences in the Fibonacci word X_n. Then for $n \geq 3$ we have*

$$C(n) = (d_1 n + d_2)\alpha^n + (d_3 n + d_4)\beta^n + d_5 n + d_6$$

where

$$d_1 = \frac{3-\sqrt{5}}{10} \qquad d_2 = \frac{17}{50}\sqrt{5} - \frac{3}{2}$$
$$d_3 = \frac{3+\sqrt{5}}{10} \qquad d_4 = -\frac{17}{50}\sqrt{5} - \frac{3}{2}$$
$$d_5 = 1 \qquad d_6 = -1.$$

19.7 Beyond Fibonacci... Tribonacci!

Define the Tribonacci numbers $(T_n)_{n\geq 0}$ by

$$T_n = \begin{cases} 0, & \text{if } n = 0; \\ 1, & \text{if } n = 1 \text{ or } n = 2; \\ T_{n-1} + T_{n-2} + T_{n-3}, & \text{if } n \geq 3. \end{cases}$$

Here are the first few terms:

n	0	1	2	3	4	5	6	7	8	9	10	11	12
T_n	0	1	1	2	4	7	13	24	44	81	149	274	504

Theorem 171 (Carlitz, Scoville, and Hoggatt [16])**.** *Every integer $n \geq 0$ has a unique representation as a sum of Tribonacci numbers of index ≥ 2, provided no three consecutive indices are used.*

Thus, for example,

$$\begin{aligned} 43 &= T_7 + T_6 + T_4 + T_2 \\ &= 24 + 13 + 4 + 2. \end{aligned}$$

We can associate each such representation of n with a binary word $(n)_T$ indicating whether a term is included in the representation. Thus, $(43)_T = 110110$.

The *infinite Tribonacci word* **TR** is the fixed point, starting with 0, of the morphism

$$0 \to 01, \qquad 1 \to 02, \qquad 2 \to 0.$$

Here are the first few terms:

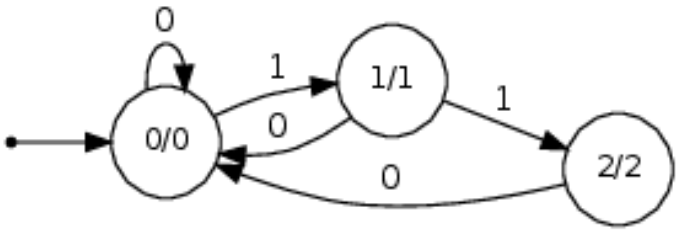
$$\mathbf{TR} = 0102010010201010201001020102010010201010201 0 \cdots$$

Alternatively, $\mathbf{TR}[n]$ can be computed by looking at the Tribonacci representation of n. It is

- 0, if the Tribonacci representation of n ends in a 0;
- 1, if the Tribonacci representation of n ends in a single 1;
- 2, if the Tribonacci representation of n ends in two 1's.

19.7.1 Tribonacci-automatic sequences

From the previous slide, it follows that **TR** can be computed by an automaton that takes, as input, the Tribonacci representation of n and outputs $\mathbf{TR}[n]$:



Any sequence that can be computed similarly is called *Tribonacci-automatic*.

Theorem 172. *The word* **TR** *is not ultimately periodic.*

Proof. We construct a formula asserting that the integer $p \geq 1$ is a period of some suffix of **TR**:

$$(p \geq 1) \ \wedge \ \exists n \ \forall i \geq n \ \mathbf{TR}[i] = \mathbf{TR}[i+p].$$

The resulting automaton accepts nothing, so **TR** is not ultimately periodic. □

Theorem 173. TR *contains no fourth powers.*

Proof. A formula for the orders of all fourth powers occurring in **TR**:

$$(n > 0) \ \wedge \ \exists i \ \forall t < 3n \ \mathbf{TR}[i+t] = \mathbf{TR}[i+n+t].$$

However, this did not run to completion on our prover. (It ran out of space while trying to determinize an NFA with 24904 states.)

Instead, substitute $j = i + t$, obtaining the new formula

$$(n > 0) \ \wedge \ \exists i \ \forall j \ ((j \geq i) \wedge (j < i + 3n)) \implies \mathbf{TR}[j] = \mathbf{TR}[j+n].$$

The resulting automaton accepts nothing, so there are no fourth powers. The largest intermediate automaton in the computation had 86711 states. □

19.7.2 Orders of squares

The *order* of a square xx is $|x|$, the length of x.

Theorem 174 (Glen [29])**.** *All squares in* **TR** *are of order T_n or $T_n + T_{n-1}$ for some $n \geq 2$. Furthermore, for all $n \geq 2$, there exists a square of order T_n and $T_n + T_{n-1}$ in* **TR**.

Proof. A natural formula for the orders of squares is

$$(n > 0) \ \wedge \ \exists i \ \forall t < n \ \mathbf{TR}[i+t] = \mathbf{TR}[i+n+t].$$

but this did not run to completion on our prover.

Instead, introduce a new variable $j = i + t$. This gives

$$(n > 0) \ \wedge \ \exists i \ \forall j \ ((i \leq j) \wedge (j < i + n)) \implies \mathbf{TR}[j] = \mathbf{TR}[j+n].$$

□

By modifying our previous formula, we get

$$(n > 0) \ \wedge \ \forall j \ ((i \leq j) \wedge (j < i + n)) \implies \mathbf{TR}[j] = \mathbf{TR}[j+n]$$

which encodes those (i, n) pairs such that there is a square of order n beginning at position i of **TR**.

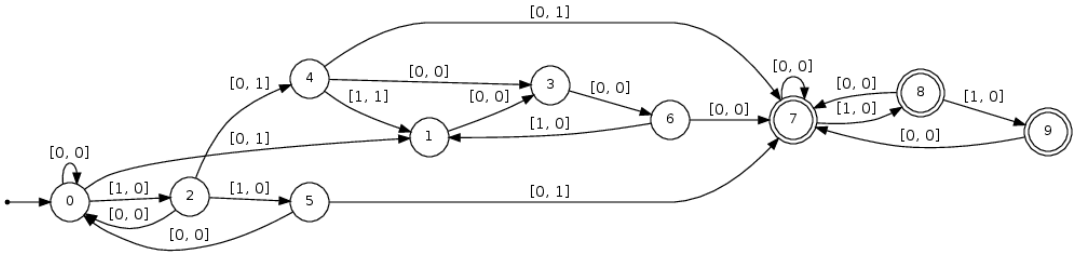
This automaton has only 10 states and efficiently encodes both the orders and starting positions of each square in **TR**.

Thus we have proved the following new result:

Theorem 175. *The language*

$$\{(i,n)_T \ : \ \textit{there is a square of order } n \textit{ beginning at position } i \textit{ in } \mathbf{TR}\}$$

is accepted by the following automaton:



19.7.3 Cubes

Theorem 176 (Glen [29])**.** *The cubes in* **TR** *are of order* T_n *for* $n \geq 5$*, and a cube of each such order occurs.*

Proof. We use the formula

$$(n > 0) \ \wedge \ \exists i \ \forall j \ ((i \leq j) \wedge (j < i + 2n)) \implies \mathbf{TR}[j] = \mathbf{TR}[j+n].$$

When we run our program, we obtain an automaton accepting exactly the language $(1000)0^*$, which corresponds to T_n for $n \geq 5$. The largest intermediate automaton had 60743 states. □

19.7.4 Enumeration

We can also mechanically *enumerate* many properties of Tribonacci-automatic sequences. For example, we can encode the factors having a given property property in terms of paths of an automaton. This gives the concept of *Tribonacci-regular sequence*.

Every Tribonacci-regular sequence $(a(n))_{n\geq 0}$ has a *linear representation* (u, μ, v) where u and v are row and column vectors, respectively, and $\mu : \Sigma_2 \to \mathbb{N}^{d\times d}$ is a matrix-valued morphism, where $\mu(0) = M_0$ and $\mu(1) = M_1$ are $d \times d$ matrices for some $d \geq 1$, such that

$$a(n) = u \cdot \mu(x) \cdot v$$

whenever $[x]_T = n$. The *rank* of the representation is the integer d.

If $\mathbf{x}$ is an infinite word, the subword complexity function $\rho_{\mathbf{x}}(n)$ counts the number of distinct factors of length n.

Theorem 177. *If* $\mathbf{x}$ *is Tribonacci-automatic, then the subword complexity function of* $\mathbf{x}$ *is Tribonacci-regular.*

Using our implementation, we can obtain a linear representation of the subword complexity function for **TR**. An obvious choice is to use the language

$$\{(n,i)_T \ : \ \forall j < i \ \mathbf{TR}[i..i+n-1] \neq \mathbf{TR}[j..j+n-1]\},$$

based on a formula that expresses the assertion that the factor of length n beginning at position i has never appeared before. Then, for each n, the number of corresponding i gives $\rho_{\mathbf{TR}}(n)$.

However, this does not run to completion in our implementation.

Instead, substitute $u = j + t$ and and $k = i - j$ to get the formula

$$\forall k \ ((k > 0) \wedge (k \leq i)) \implies (\exists u \ ((u \geq j) \wedge (u < n + j) \wedge (\mathbf{TR}[u] \neq \mathbf{TR}[u + k]))).$$

This formula is close to the upper limit of what we can compute using our program.

The largest intermediate automaton had 1230379 states and the program took 12323.82 seconds, giving us a linear representation (u, μ, v) of rank 22.

When we minimize this representation, we get the rank-12 linear representation

$$u = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$$

$$M_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 2 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -4 & 0 & 2 & 0 & 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -5 & 0 & 2 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ -6 & 0 & 2 & 0 & 3 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ -10 & 0 & 3 & 0 & 4 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad M_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$v = [1\ 3\ 5\ 7\ 9\ 11\ 15\ 17\ 21\ 29\ 33\ 55]^R.$$

Comparing this to an independently-derived linear representation of the function $n \to 2n + 1$, we see they are the same. Thus we get

Theorem 178 (Droubay-Justin-Pirillo, 2001)**.** *The subword complexity function of* **TR** *is* $2n + 1$.

19.7.5 The finite Tribonacci words

The *finite Tribonacci words* $(Y_n)_{n\geq 0}$ are defined as follows:

$$\begin{aligned} Y_0 &= \epsilon \\ Y_1 &= 2 \\ Y_2 &= 0 \\ Y_3 &= 01 \\ Y_n &= Y_{n-1}Y_{n-2}Y_{n-3} \text{ for } n \geq 4. \end{aligned}$$

Note that Y_n, for $n \geq 2$, is the prefix of length T_n of **TR**.

Our method can also prove interesting things about the finite Tribonacci words.

What is the exact number of square occurrences in the finite Tribonacci words Y_n? To solve this using our approach, we first *generalize* the problem to consider *any* length-n prefix of Y_n, and not simply the prefixes of length T_n.

The formula represents the number of distinct squares in $\mathbf{TR}[0..n-1]$:

$$\begin{aligned} L_{\text{ds}} := \{(n,i,j)_T \ : \ & (j \geq 1) \text{ and } (i+2j \leq n) \\ & \text{and } \mathbf{TR}[i..i+j-1] = \mathbf{TR}[i+j..i+2j-1] \\ & \text{and } \forall i' < i \ \mathbf{TR}[i'..i'+2j-1] \neq \mathbf{TR}[i..i+2j-1]\}. \end{aligned}$$

This formula asserts that $\mathbf{TR}[i..i+2j-1]$ is a square occurring in $\mathbf{TR}[0..n-1]$ and that furthermore it is the first occurrence of this particular word in $\mathbf{TR}[0..n-1]$.

This represents the total number of occurrences of squares in $\mathbf{TR}[0..n-1]$:

$$\begin{aligned} L_{\text{dos}} := \{(n,i,j)_T \ : \ & (j \geq 1) \text{ and } (i+2j \leq n) \text{ and} \\ & \mathbf{TR}[i..i+j-1] = \mathbf{TR}[i+j..i+2j-1]\}. \end{aligned}$$

This formula asserts that $\mathbf{TR}[i..i+2j-1]$ is a square occurring in $\mathbf{TR}[0..n-1]$.

Unfortunately, applying our enumeration method to this suffers from the same problem as before, so we rewrite it as

$$(j \geq 1) \wedge \ (i+2j \leq n) \ \wedge \forall u \ ((u \geq i) \wedge (u < i+j)) \implies \mathbf{TR}[u] = \mathbf{TR}[u+j]$$

When we compute the linear representation of the function counting the number of such i and j, we get a linear representation of rank 63.

Now we compute the minimal polynomial of M_0, which is $(x-1)^2(x^2+x+1)^2(x^3-x^2-x-1)^2$. Solving a linear system in terms of the roots (or, more accurately, in terms of the sequences 1, n, T_n, T_{n-1}, T_{n-2}, nT_n, nT_{n-1}, nT_{n-2}) gives

Theorem 179. *The total number of occurrences of squares in the Tribonacci word Y_n is*

$$c(n) = \frac{n}{22}(9T_n - T_{n-1} - 5T_{n-2}) + \frac{1}{44}(-117T_n + 30T_{n-1} + 33T_{n-2}) + n - \frac{7}{4}$$

for $n \geq 5$.

19.7.6 Cube occurrences

In a similar way, we can count the occurrences of cubes in the finite Tribonacci word Y_n. Here we get a linear representation of rank 46. The minimal polynomial for M_0 is $x^4(x^3 - x^2 - x - 1)^2(x^2 + x + 1)^2(x - 1)^2$. Using analysis exactly like the square case, we find

Theorem 180. *Let $C(n)$ denote the number of cube occurrences in the Tribonacci word Y_n. Then for $n \geq 3$ we have*

$$C(n) = \frac{1}{44}(T_n + 2T_{n-1} - 33T_{n-2}) + \frac{n}{22}(-6T_n + 8T_{n-1} + 7T_{n-2}) + \frac{n}{6} - \frac{1}{4}[n \equiv 0 \pmod 3] + \frac{1}{12}[n \equiv 1 \pmod 3] - \frac{7}{12}[n \equiv 2 \pmod 3].$$

Here $[P]$ is Iverson notation, and equals 1 if P holds and 0 otherwise.

19.7.7 Orders and positions of cubes

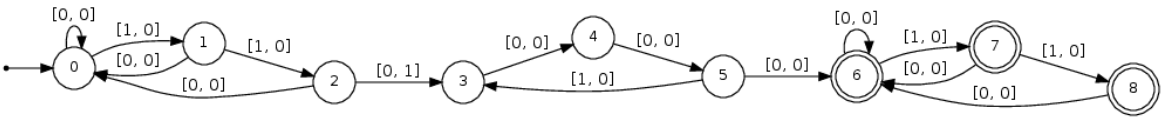
Next, we encode the orders and positions of all cubes. We build a DFA accepting the language

$$\{(i, n)_T \ : \ (n > 0) \ \wedge \ \forall j \ ((i \leq j) \wedge (j < i + 2n)) \implies \mathbf{TR}[j] = \mathbf{TR}[j + n]\}.$$

Theorem 181. *The language*

$$\{(n, i)_T \ : \ \textit{there is a cube of order } n \textit{ beginning at position } i \textit{ in } \mathbf{TR}\}$$

is accepted by the automaton below:



19.8 Palindromes

We now turn to a characterization of the palindromes in **TR**. Once again, it turns out that the obvious formula

$$\exists i \ \forall j < n \ \mathbf{TR}[i + j] = \mathbf{TR}[i + n - 1 - j],$$

resulted in an intermediate NFA of 5711 states that we could not successfully determinize.

Instead, we used two equivalent formulas. The first accepts n if there is an even-length palindrome, of length $2n$, centered at position i:

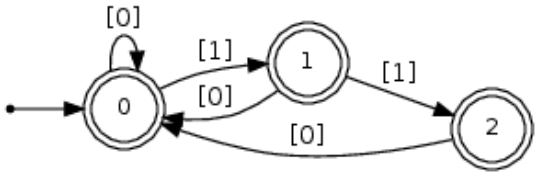
$$\exists i \geq n \ \forall j < n \ \mathbf{TR}[i+j] = \mathbf{TR}[i-j-1].$$

The second accepts n if there is an odd-length palindrome, of length $2n+1$, centered at position i:

$$\exists i \geq n \ \forall j \ (1 \leq j \leq n) \implies \mathbf{TR}[i+j] = \mathbf{TR}[i-j].$$

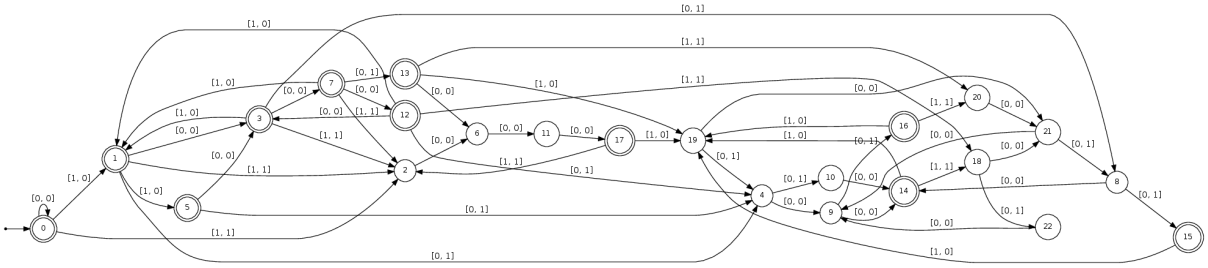
Theorem 182. *There exist palindromes of every length ≥ 0 in* **TR**.

Proof. For the first formula, our program outputs the automaton below. It clearly accepts the Tribonacci representations for all n.



□

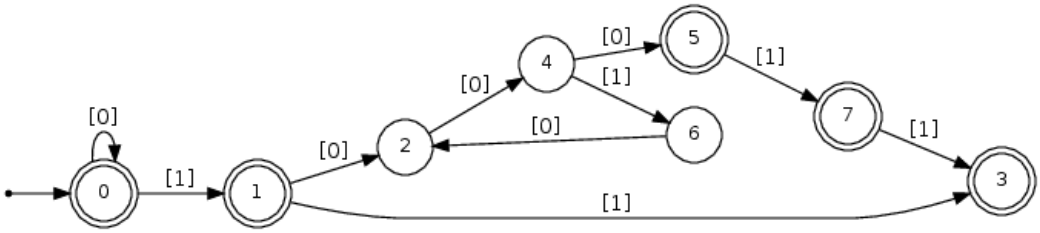
We could also characterize the positions of all nonempty palindromes. To illustrate the idea, we generated an automaton accepting (i, n) such that $\mathbf{TR}[i-n..i+n-1]$ is an (even-length) palindrome.



The prefixes are factors of particular interest. Let us determine which prefixes are palindromes:

Theorem 183. *The prefix* $\mathbf{TR}[0..n-1]$ *of length n is a palindrome if and only if $n = 0$ or $(n)_T \in 1 + 11 + 10(010)^*(00 + 001 + 0011)$.*

Proof. We use the formula $\forall i < n\ \mathbf{TR}[i] = \mathbf{TR}[n-1-i]$. The automaton generated is given below.



□

19.9 Going even further

- Adders exist for numeration systems based on Pisot numbers: these are real numbers > 1 all of whose conjugates lie inside the unit circle. So we can create decision procedures for these numeration systems, too.
- The paperfolding words: this is an uncountable class of non-automatic sequences encoded by infinite words: we can prove theorems about uncountably many different sequences simultaneously!
- The Sturmian words: modulo a few details which still need to be proven, Luke Schaeffer could show that there is a decidable theory for these words, too.

To summarize:

- The logic-based approach gives a powerful way to state, decide, and enumerate properties of automatic sequences and their generalizations
- It allows proving, in generality, many particular cases that already appeared in the literature, using a unified framework
- Although the worst-case running time of the decision procedure is formidable, an implementation often succeeds in proving useful results

19.10 Notes

For more about how to solve decision problems involving Fibonacci- and Tribonacci-automatic sequences, see [25, 26, 38].

Bibliography

[1] B. Adamczewski and J. Cassaigne. Diophantine properties of real numbers generated by finite automata. *Compos. Math.* **142** (2006), 1351–1372.

[2] J.-P. Allouche. Sur la transcendance de la série formelle π. *Séminaire de Théorie des Nombres de Bordeaux* **2** (1990), 103–117.

[3] G. Allouche, J.-P. Allouche, and J. Shallit. Kolams indiens, dessins sur le sable aux îles Vanuatu, courbe de Sierpínski, et morphismes de monoïde. *Ann. Inst. Fourier (Grenoble)* **56** (2006), 2115–2130.

[4] J.-P. Allouche, D. Astoorian, J. Randall, and J. Shallit. Morphisms, squarefree strings, and the Tower of Hanoi puzzle. *Amer. Math. Monthly* **101** (1994), 651–658.

[5] J.-P. Allouche and J. O. Shallit. The ring of k-regular sequences. *Theoret. Comput. Sci.* **98** (1992), 163–197.

[6] J.-P. Allouche and J. O. Shallit. *Automatic Sequences*. Cambridge University Press, 2003.

[7] J.-P. Allouche and J. O. Shallit. The ring of k-regular sequences, II. *Theoret. Comput. Sci.* **307** (2003), 3–29.

[8] J. P. Bell, M. Coons, and K. G. Hare. The minimal growth of a k-regular sequence. *Bull. Austral. Math. Soc.* **90** (2014), 195–203.

[9] J. Berstel and C. Reutenauer. *Noncommutative Rational Series with Applications*, Vol. 137 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, 2011.

[10] V. Bruyère, G. Hansel, C. Michaux, and R. Villemaire. Logic and p-recognizable sets of integers. *Bull. Belgian Math. Soc.* **1** (1994), 191–238. Corrigendum, *Bull. Belg. Math. Soc.* **1** (1994), 577.

[11] Y. Bugeaud. On the rational approximation to the Thue-Morse-Mahler numbers. *Ann. Inst. Fourier (Grenoble)* **61** (2011), 20652076.

[12] Y. Bugeaud. Automatic continued fractions are transcendental or quadratic. *Ann. Sci. École Norm. Sup.* **46** (2013), 1005–1022.

[13] Y. Bugeaud. Expansions of algebraic numbers. In *Four Faces of Number Theory*, EMS Ser. Lect. Math., pp. 31–75. Eur. Math. Soc., 2015.

[14] J. Byszewski and J. Konieczny. A density version of Cobham's theorem. *Acta Arith.* **192** (2020), 235–247.

[15] J. Byszewski, J. Konieczny, and E. Krawczyk. Substitutive systems and a finitary version of Cobham's theorem. Arxiv preprint, available at https://arxiv.org/abs/1908.11244, 2019.

[16] L. Carlitz, R. Scoville, and V. E. Hoggatt, Jr. Fibonacci representations of higher order. *Fibonacci Quart.* **10** (1972), 43–69,94.

[17] J. Cassaigne. Special factors of sequences with linear subword complexity. In J. Dassow, G. Rozenberg, and A. Salomaa, editors, *Developments in Language Theory II*, pp. 25–34. World Scientific, 1996.

[18] G. Christol. Ensembles presque périodiques k-reconnaissables. *Theoret. Comput. Sci.* **9** (1979), 141–145.

[19] G. Christol, T. Kamae, M. Mendès France, and G. Rauzy. Suites algébriques, automates et substitutions. *Bull. Soc. Math. France* **108** (1980), 401–419.

[20] A. Cobham. A proof of transcendence based on functional equations. Technical Report RC-2041, IBM Yorktown Heights, March 25 1968.

[21] A. Cobham. On the Hartmanis-Stearns problem for a class of tag machines. In *IEEE Conference Record of 1968 Ninth Annual Symposium on Switching and Automata Theory*, pp. 51–60, 1968. Also appeared as IBM Research Technical Report RC-2178, August 23 1968.

[22] A. Cobham. On the base-dependence of sets of numbers recognizable by finite automata. *Math. Systems Theory* **3** (1969), 186–192.

[23] A. Cobham. Uniform tag sequences. *Math. Systems Theory* **6** (1972), 164–192.

[24] F. M. Dekking, M. Mendès France, and A. J. van der Poorten. Folds! *Math. Intelligencer* **4** (1982), 130–138, 173–181, 190–195. Erratum, **5** (1983), 5.

[25] C. F. Du, H. Mousavi, E. Rowland, L. Schaeffer, and J. Shallit. Decision algorithms for Fibonacci-automatic words, II: Related sequences and avoidability. *Theoret. Comput. Sci.* **657** (2017), 146–162.

[26] C. F. Du, H. Mousavi, L. Schaeffer, and J. Shallit. Decision algorithms for Fibonacci-automatic words, III: enumeration and abelian properties. *Internat. J. Found. Comp. Sci.* **27** (2016), 943–963.

[27] J. E. Foster. A number system without a zero symbol. *Math. Mag.* **21**(1) (1947), 39–41.

[28] A. S. Fraenkel and J. Simpson. The exact number of squares in fibonacci words. *Theoret. Comput. Sci.* **218** (1999), 95–106.

[29] A. Glen. Occurrences of palindromes in characteristic Sturmian words. *Theoret. Comput. Sci.* **352** (2006), 31–46.

[30] D. Goc, L. Schaeffer, and J. Shallit. The subword complexity of k-automatic sequences is k-synchronized. In M.-P. Béal and O. Carton, editors, *DLT 2013*, Vol. 7907 of *Lecture Notes in Computer Science*, pp. 252–263. Springer-Verlag, 2013.

[31] T. Høholdt, H. E. Jensen, and J. Justesen. Aperiodic correlations and the merit factor of a class of binary sequences. *IEEE Trans. Inform. Theory* **31** (1985), 549–552.

[32] T. J. P. Krebs. A more reasonable proof of Cobham's theorem. Arxiv preprint, available at https://arxiv.org/abs/1801.06704, 2018.

[33] S. Lehr. Sums and rational multiples of q–automatic sequences are q–automatic. *Theoret. Comput. Sci.* **108** (1993), 385–391.

[34] J. Liouville. Sur des classes très étendues de quantités dont la valeur n'est ni algébrique, ni même reductible à des irrationelles algébriques. *C. R. Acad. Sci. Paris* **18** (1844), 883–885,910–911.

[35] M. Lothaire. *Algebraic Combinatorics on Words*. Cambridge University Press, 2002.

[36] J. H. Loxton and A. J. van der Poorten. Arithmetic properties of automata: regular sequences. *J. Reine Angew. Math.* **392** (1988), 57–69.

[37] L. Mol, N. Rampersad, J. Shallit, and M. Stipulanti. Cobham's theorem and automaticity. *Internat. J. Found. Comp. Sci.* **30** (2019), 1363–1379.

[38] H. Mousavi, L. Schaeffer, and J. Shallit. Decision agorithms for Fibonacci-automatic words, I: Basic results. *RAIRO Inform. Théor. App.* **50** (2016), 39–66.

[39] L. Schaeffer and J. Shallit. Closed, palindromic, rich, privileged, trapezoidal, and balanced words in automatic sequences. *Electronic J. Combinatorics* **23**(1) (2016), #P1.25 (electronic).

[40] W. M. Schmidt. Simultaneous approximations to algebraic numbers by rationals. *Acta Math.* **125** (1970), 189–201.

[41] W. M. Schmidt. Norm form equations. *Ann. Math.* **96** (1972), 526–551.

[42] W. M. Schmidt. *Diophantine Approximation*, Vol. 785 of *Lecture Notes in Mathematics*. Springer-Verlag, 1980.

[43] J. O. Shallit. Simple continued fractions for some irrational numbers. *J. Number Theory* **11** (1979), 209–217.

[44] J. O. Shallit. Simple continued fractions for some irrational numbers, II. *J. Number Theory* **14** (1982), 228–231.

[45] J. O. Shallit. Explicit descriptions of some continued fractions. *Fibonacci Quart.* **20** (1982), 77–81.

[46] R. M. Smullyan. *Theory of Formal Systems*, Vol. 47 of *Annals of Mathematical Studies*. Princeton University Press, 1961.

[47] D. Thakur. *Function Field Arithmetic*. World Scientific, 2004.

[48] A. J. van der Poorten and J. O. Shallit. Folded continued fractions. *J. Number Theory* **40** (1992), 237–250.

[49] A. J. van der Poorten and J. Shallit. A specialised continued fraction. *Canad. J. Math.* **45** (1993), 1067–1079.

[50] H. Wilf. What is an answer? *Amer. Math. Monthly* **89** (1982), 289–292.