University of Waterloo

Lecture Notes

Automatic Sequences

Prof. Jeffrey Shallit

typed by
Andrej Vuković

February 27, 2020

Contents

Abstract

This is a series of lecture notes for a class on automatic sequences taught by Jeffrey Shallit.

1 Jan. 7, 2020.

The course code is CS 860. There will be a final project for this course involving a report and presentation. But if you don't like presenting, you can take the option of editing Wikipedia articles related to some topic in the course too. If you do your own research in this course, you can talk about that for the final project too. The course URL is `https://cs.uwaterloo.ca/~shallit/Courses/860`. We can come by whenever the door is open, but Professor Shallit asks that we do not knock if the door is closed. Office hours are 2:30 to 3:20 on Wednesdays in DC 3134. There will be three problem sets. One can also earn bonus marks by solving open problems (automatic 100), finding an interesting sequence that is not in the OEIS, and finding errata in the course textbook that are not already on the errata page. The course textbook should be available at the school bookstore.

Let Σ be some alphabet, usually finite. A *sequence* is a function $s : \mathbb{N} \to \Sigma$. In this course, we use the convention $\mathbb{N} := \{0, 1, 2, \ldots\}$. A *bi-infinite sequence* is a function $s : \mathbb{Z} \to \Sigma$. For example, we might define the sequence

$$p_n := \begin{cases} 1, & \text{if } n \text{ is a prime} \\ 0, & \text{otherwise} \end{cases}$$

By the prime number theorem, we have that

$$\sum_{0 \le n < N} p_n = \Theta\left(\frac{N}{\log N}\right).$$

If S is a set, let

$$\chi_S(n) := \begin{cases} 1, & \text{if } n \in S \\ 0, & \text{if } n \notin S \end{cases}$$

The function $\chi_S(n)$ is known as the *characteristic function* or *indicator function* of S.

Perhaps the most famous non-trivial example of an automatic sequence is the *Thue–Morse sequence* $t := (t_n)_{n \ge 0}$ given by

$$t_n := s_2(n) \pmod 2$$

where $s_2(n)$ is the sum of the bits in n's base 2 representation. The rows in the following table, from top to bottom, are n, $s_2(n)$, and t_n.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	2	1	2	2	3	1	2	2	3	2	3	3	4
0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0

We have $t_0 = 0$, $t_{2n} = t_n$, and $t_{2n+1} = 1 - t_n$. This is because $(n)_2 0$ (i.e., the base 2 representation of n with a right-appended 0) is equal to $(2n)_2$, and $(n)_2 1 = (2n+1)_2$.

Informally, a sequence $(s_n)_{n \geq 0}$ is *k-automatic* if:

(a) the range is finite, and
(b) it satisfies a system of equations where subscripts are $k^i n + a$ where $k \geq 2$ is an integer and $0 \leq a < k^i$.

The first version of the Thue–Morse sequence that Professor Shallit knows about appeared in the literature in 1851. It was later systematically studied by Axel Thue, and independently by Marston Morse.

The simplest sequences are *ultimately periodic*. These are defined by the property that there exist some $p \geq 1$ and $N \geq 0$ such that $s_n = s_{n+p}$ for all $n \geq N$. The value p is referred to as the *period* and N is referred to as the *preperiod*. (Note that our sequences are indexed starting from 0, so going up to the preperiod gives us $(s_0, s_1, \ldots, s_{N-1})$, and there are N terms there.) All ultimately periodic sequences are automatic.

The most complicated sequences are *random*. These are hard to define precisely, but morally they should not be automatic.

Automatic sequences mimic random sequences, in some sense. One example is the *Rudin–Shapiro sequence* $(r_n)_{n \geq 0}$. It is given by

$$r_n := a_{11}(n) \pmod 2$$

where $a_{11}(n)$ is the number of occurrences of the block "11" in the base 2 representation of n. The resulting sequence is tabulated here, the first row from the top down being labelled by n, the second by $a_{11}(n)$, and the third by r_n.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	1	0	0	1	2	0	0	0	1	1	1	2	3
0	0	0	1	0	0	1	0	0	0	0	1	1	1	0	1

We have the following summatory result for this sequence. For every c,

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

The notation on the left-hand side is the *Iverson bracket*, named after Ken Iverson, a Canadian Turing Award winner. It is defined as follows:

$$[x = y] = \begin{cases} 1, & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

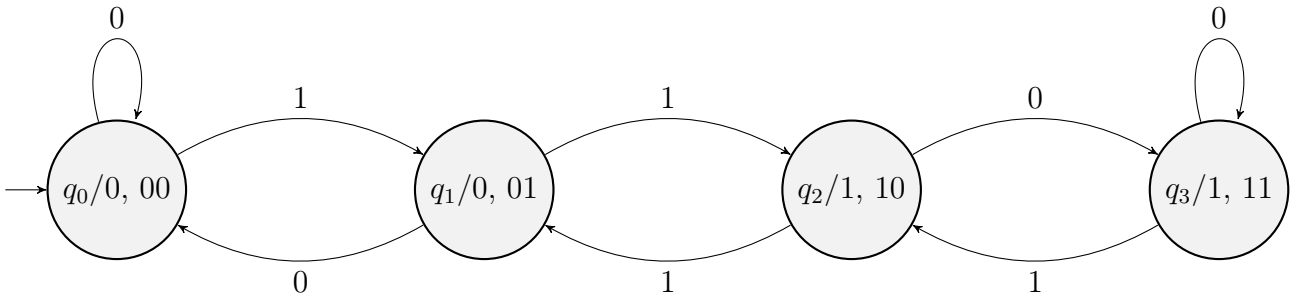
Thus, this summatory result is telling us that the sequence (r_n) is hardly correlated with its shifts. We have that $r_{2n} = r_n$, $r_{4n+1} = r_n$, $r_{8n+3} = 1 - r_n$, and $r_{8n+7} = 1 - r_{4n+3}$.

Apparently a few years ago, a student in a previous installment of this course studied the applications of automatic sequences in music theory; they ended up publishing in a music journal, and there was a collaboration with composer Per Nørgård.

We are interested in the computational, logical, algebraic, combinatorial, and number-theoretic properties of automatic sequences.

Automatic sequences are precisely those sequences generated by *deterministic finite automata with output* (*DFAOs*). To compute a_n, we express n in base $k \geq 2$ and start at the initial state of the corresponding DFAO. Then we follow transitions on input and end at a state whose output is a_n.

For example, the Rudin–Shapiro sequence is generated by the following DFAO.



The notation of a circle with q/a in it represents a state called q whose output is a. The numbers 00, 01, 10, and 11 describe the state.

We are interested in questions like the following.

-What is the dependence on k?
-Does the definition depend on the order you read the digits? (It turns out it does not.)
-How many states are needed?
-What are the sequences that are both k- and k'-automatic?

Here are some more computational questions about automatic sequences.

-What techniques can we use to prove a sequence is not k-automatic?
-What transformations on sequences preserve k-automaticity?

We are also interested in the logical properties of automatic sequences. For $k \geq 2$ and $n \geq 1$, let

$$V_k(n) := \max\left\{k^i \mid k^i \text{ divides } n\right\}.$$

Then $\mathrm{FO}(\mathbb{N}, +, V_k)$, i.e., the first-order logical theory of $\mathbb{N}$, $+$, and V_k, is precisely the theory of k-automatic sequences. It is known to be decidable. We then have the following question.

-What related logical theories are decidable or undecidable?

It is known that $\mathrm{FO}(\mathbb{N}, +, V_2, V_3)$ is undecidable.

Define

$$P_k(n) := \begin{cases} 1, & \text{if } n = k^j, j \geq 0 \\ 0, \text{otherwise} \end{cases}$$

The theory $\mathrm{FO}(\mathbb{N}, +, P_2)$ is decidable. It is not known whether $\mathrm{FO}(\mathbb{N}, +, P_2, P_3)$, which is a subtheory of $\mathrm{FO}(\mathbb{N}, +, V_2, V_3)$, is decidable.

-What questions about automatic sequences are decidable?

We now discuss algebraic properties of automatic sequences. Suppose we make the following formal power series out of the Thue–Morse sequence:

$$T(X) := \sum_{n \geq 0} t_n X^n \in \mathrm{GF}(2)[[X]].$$

Here $\mathrm{GF}(2)$ just means $\mathbb{F}_2$, the field of two elements. We calculate

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

Over $\mathrm{GF}(2)$, we have $(X + Y)^2 = X^2 + Y^2$, so

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

which implies that

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

so

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

Thus, $T(X)$ is algebraic over $\mathrm{GF}(2)[X]$, the collection of polynomails with coefficients mod 2. More generally, we have *Christol's theorem*, which states that $(a_n)_{n \geq 0}$ is p-automatic for p prime if and only if $\sum_{n \geq 0} a_n X^n$ is algebraic over $\mathrm{GF}(p)[X]$.

We now discuss combinatorial properties of automatic sequences.

-Given a sequence $a = (a_i)_{i \geq 0}$, how many distinct length n blocks occur in a?

The function sending n to the number of length n blocks in a is called the *subword complexity* of a and is denoted by $\rho(n)$. If a is an automatic sequence, we have

$$\rho(n) = O(n).$$

Here are some more questions.

-Given a sequence $a = (a_i)_{i \geq 0}$, what is $\sum_{0 \leq n < N} [a_n = c]$ for different c?
-Given two distinct automatic sequences, what blocks occur in common?

Finally, we look at automatic sequences from the perspective of number theory. Given a sequence $(a_n)_{n \geq 0}$, consider the real number

$$\sum_{i \geq 0} a_i b^{-i}$$

for $b \geq 2$. Then we have the result that this number is either rational or transcendental when (a_n) is automatic, for any $b \geq 2$.

2 Jan. 9, 2020.

Today we start discussing some basic concepts for the course.

We let $(n)_k$ denote the canonical base k representation of n, i.e., the unique representation that only uses digits in $\{0, 1, \ldots, k-1\}$ and that has no leading zeroes, starting with the most significant digit (msd). By convention, $(0)_k = \epsilon$, the empty string, since to set $(0)_k$ to 0 would violate the "no leading zeroes" rule.

We let $[w]_k$ denote the evaluation of w, treated as a base k representation, msd first. That is, if $w = a_1 a_2 \ldots a_t$, then

$$[w]_k := \sum_{i=1}^{t} a_i k^{t-i}.$$

For example, $[21]_2 = 5$.

We let $\nu_k(n)$ be the exponent of the highest power of k dividing n. If $\nu_k(n) = e$, we write $k^e \mid\mid n$. We also set $V_k(n) := k^{\nu_k(n)}$. Apparently there was a famous paper that used V_k in place of ν_k but became correct when these were switched.

Let w^R be the reversal of the word w. For example, $(\text{drawer})^R = \text{reward}$. Let $\Sigma_k := \{0, 1, \ldots, k-1\}$. Let Σ^* denote the set of finite words over Σ. Let Σ^w denote the set of

1-sided infinite words over Σ. Let ${}^w\Sigma^w$ and $\Sigma^{\mathbb{Z}}$ denote the set of 2-sided infinite words over Σ. Let x^w denote the infinite string $xxx\ldots$. Let $|x|$ denote the length of x. Let $|x|_a$ denote the number of occurrences of a in x. Let $x[i]$ denote the i^{th} letter of x. Let $x[i\ldots j]$ denote the subword $x[i]x[i+1]\ldots x[j]$ of x.

Theorem 2.1. *Let $k \geq 2$. Every integer $n \geq 0$ has a unique canonical base k representation.*

Proof. Suppose $n < k$. We have that $(0)_k = \epsilon$ is the unique canonical base k representation of $n = 0$. For $1 \leq n < k$, $(n)_k = n$ is the unique canonical base k representation of n. Now suppose $n \geq k$. Then we can write $n = kn' + a$ for some n' and a with $0 \leq a < k$. Suppose $(n')_k =: x$. Then $[xa]_k = n$ because $[x]_k = (n-a)/k$.

Suppose w and x are distinct canonical words such that $[w]_k = [x]_k$. Pad the shorter word with leading zeroes if necessary to obtain that w and x start with digits a_1 and b_1, respectively. Assume $a_1 < b_1$. Then

$$\begin{aligned}
[x]_k - [w]_k &\geq k^{t-1} + (1-k)(1 + k + \cdots + k^{t-2}) \\
&= k^{t-1} + (1-k)\left(\frac{k^{t-1}-1}{k-1}\right) \\
&= k^{t-1} - (k^{t-1} - 1) \\
&= 1.
\end{aligned}$$

□

We now discuss *bijective representation*. Here we use an alternate digit set for base k. Instead of $\{0, 1, \ldots, k-1\}$, we use $\{1, 2, \ldots, k\}$. This is used the fix the issue of ambiguity with leading zeroes in base k representation. In base 2 bijective representation, 0 is written as ϵ, 1 is written as 1, 2 is written as 2, 3 is written as 11, 4 is written as 12, 5 is written as 21, 6 is written as 22, 7 is written as 111, etc.

Next, we discuss *Fibonacci* (or *Zeckendorf*) *representation*. Let $F_0 := 0$, $F_1 := 1$, and $F_n := F_{n-1} + F_{n-2}$ be the Fibonacci numbers. (Professor Shallit says, "This is the standard way to index them, and anyone who indexes them any other way is just wrong.") Then we can express n as

$$n = \sum_{i=1}^{t} a_i F_{t+2-i}$$

where $a_i \in \{0, 1\}$ for each i. In this representation, 0 is written as ϵ, 1 is written as 1, 2 is written as 10, 3 is written as 100, 4 is written as 101, 5 is written as 1000, 6 is written as 1001, 7 is written as 1010, 8 is written as 10000, etc. This representation is unique subject to the constraint that $a_i a_{i+1} \neq 1$ for all i. Many game theory results can be simply expressed if you write the numbers involved in Fibonacci representation.

Example 2.2. Suppose we have a sequence of replacement rules $a \mapsto ab$, $b \mapsto a$. (This is

called a *morphism*.) Applying these rules to a iteratively, we get the successive words

$$a,$$

$$ab,$$

$$aba,$$

$$abaab,$$

$$abaababa.$$

In the limit, we get some infinite word $abaababa\ldots$, which is a fixed point of the morphism. Now, suppose we write out $n = 0, 1, 2, 3, 4, 5, 6, 7$, etc., and below each number we write its Fibonacci representation vertically with the leading digit at the top. Then the bottom digit of each number's Fibonacci representation determines whether the corresponding letter in the word $abaababa\ldots$ is an a or a b. (For example, the second leftmost letter is b, so we look at $n = 1$, write its Fibonacci representation 1 below, and then the bottom digit is a 1, so we expect to get a b. The third leftmost letter is an a, and the corresponding number is $n = 2$, and the bottom digit of its Fibonacci representation 10 is 0, so we expect to get an a.)

We now discuss the *generalized Fibonacci representation*. We let $F_n^{(k)} := 0$ for $0 \leq n \leq k-2$, $F_n^{(k)} = 1$ for $n = k-1$, and $F_n^{(k)} := F_{n-1}^{(k)} + \cdots + F_{n-k}^{(k)}$ for $n \geq k$. For $k = 2$, we get the Fibonacci sequence (F_n). For $k = 3$, we get the *Tribonacci sequence* (T_n). Professor Shallit says, "I won't even tell you what the $k = 4$ case is called because it's too stupid." (Tetranacci, by the way...)" We write

$$n = \sum_{i=1}^{t} a_i F_{t+k-i}^{(k)}$$

with $a_i \in \{0, 1\}$ for all i. This is unqiue provided that $a_i a_{i+1} \cdots a_{i+k-1} \neq 1$ for all i.

We now discuss the *greedy representation*. Let $1 = u_0 < u_1 < u_2 < \cdots$ be a strictly increasing sequence of integers. We write $n = \sum_{0 \leq i \leq r} a_i u_i$ where $a_i \in \mathbb{N}$ for each i. (Recall that $\mathbb{N}$ includes 0 in this course.) We can write the following pseudocode greedy algorithm to generate this representation.

Algorithm 1 Obtaining the greedy representation.

```
1: procedure GREEDY(n)
2:     t := 0
3:     while (u_{t+1} ≤ n) do t := t + 1
4:     for i := t downto 0 do
5:         a_i := ⌊n/v_i⌋
6:         n := n − a_i u_i
7:     return (a_t a_{t−1} ... a_0)
```

We say $m \leq n$ if and only if $(m)_G \leq (n)_G$ where the subscript G denotes the greedy representation and we use radix ordering.

For words x, y, we say $x < y$ if $|x| < |y|$ or if $|x| = |y|$ and there exist w, x_1, y_1, a, b such that $x = wax_1$, $y = wby_1$, and $a < b$.

The following theorem is due to Fraenkel. It can be used to prove uniqueness of all the expansions we have discussed so far.

Theorem 2.3. *Let $1 = u_0 < u_1 < u_2 < \cdots$ 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 $a_s \neq 0$, the a_i's are all in $\mathbb{N}$, and the a_i's satisfy*

$$a_0 u_0 + a_1 u_1 + \cdots + a_i u_i < u_{i+1}$$

for all i.

We now discuss balanced ternary representations. This can be used to describe all integers, not just positive ones. This is essentially ternary representation with digits $-1, 0, 1$. One of the earlier computers ever built used balanced ternary, with "trits" instead of bits. Here 0 is represented as ϵ, 1 is represented as 1, -1 is represented as -1, -2 is represented as -11, -3 is represented as -10, -4 is represented as $-1,-1$, -5 is represented as -111, 2 is represented as $1,-1$, 3 is represented as 10, 4 is represented as 11, etc. One can prove that every integer has a unique balanced ternary representation provided there are no leading zeroes.

Balanced ternary is a special case of the (k, ℓ)-*numeration system*. In this system, we work in base $(k+\ell+1)$ with digits $\{-k, -k+1, \ldots, -1, 0, 1, \ldots, \ell\}$. Balanced ternary is obtained when $k = \ell = 1$.

We can also work in a negative base, say base $(-k)$ for $k \ge 2$. We use the digits $\{0, 1, \ldots, k-1\}$, and every element of $\mathbb{Z}$ has a unique representation. For example, in base -2 we represent -4 as 1100, -3 as 1101, -2 as 10, -1 as 11, 0 as ϵ, 1 as 1, 2 as 110, 3 as 111, 4 as 100, etc.

There are other representation systems for representing, for example, the Gaussian integers. It is not so easy to get unique representation there; the only bases that work are of the form $i - n$ where n is an integer, or something like that.

A *deterministic finite automaton with output* (*DFAO*) is a 6-tuple $M = (Q, \Sigma, \Delta, q_0, \delta, \tau)$ where:

(i) Q is a finite non-empty set of states, often denoted $\{q_0, q_1, \ldots, q_{t-1}\}$;
(ii) Σ is an input alphabet, often the alphabet $\Sigma_k := \{0, 1, \ldots, k-1\}$;
(iii) Δ is an output alphabet;
(iv) q_0 is an initial state;
(v) δ is a transition function $\delta : Q \times \Sigma \to Q$; and
(vi) τ is an output function $\tau : Q \to \Delta$.

We can extend the domain of δ to $Q \times \Sigma^*$ by letting $\delta(q, x)$ be the state we get to on reading an input word x, starting from q. More formally, we set $\delta(q, \epsilon) := q$ and $\delta(q, xa) := \delta(\delta(q, x), a)$ for all $x \in \Sigma^*$ and $a \in \Sigma$. On input x, the output of M is defined to be $\tau(\delta(q_0, x))$.

Then M computes a sequence $(a_n)_{n \geq 0}$ by letting $x = (n)_k$ be the canonical base k representation of n and letting a_n be the output when the input is n.

Next time, we will discuss how we can read the reversal x^R instead of x (with a different automaton) but pay a price. The price is that the number of states can be as large as $|\Delta|^{|Q|}$. It is difficult to actually build automata with more than order a hundred million states on a computer, so this exponential blowup can cause problems in actually building automata to do various tasks, if they involve a word reversal.

3 Jan. 14, 2020.

A sequence (a_n) is k-automatic if its n^{th} term can be obtained by writing n in base k, feeding it into some DFAO, and getting output a_n. Note that even if we put $0^i(n)_k$ (for any $i \geq 0$) into the DFAO instead of $(n)_k$, we can change the DFAO a bit so that $\delta(q_0, 0) = q_0$, and then the sequence will still be accepted by the DFAO. So the notion of "automatic sequence" is robust to small changes like inputting $0^i(n)_k$ instead of $(n)_k$.

Theorem 3.1. *Suppose $f : \Sigma^* \to \Delta$ is a finite-state function (i.e., is computed by a DFAO). Then f^R, defined by $f^R(w) := f(w^R)$, is also a finite-state function.*

This is a generalization of a theorem often seen in a first course on automata theory, but now we're doing it for functions rather than languages. We will get slightly bogged down in notation in the course of the proof.

Proof. Suppose that f is computed by a DFAO $(Q, \Sigma, \Delta, q_0, \delta, \tau)$. Then f^R will be computed by $(S, \Sigma, \Delta, q_0', \delta', \tau')$ where $S := \Delta^Q$ (i.e., the set of all functions from Q to Δ), q_0' is the function $q \mapsto \tau(q)$, $\tau'(h) := h(q_0)$, and $\delta'(g, a) := h$ where $h(q) := g(\delta(q, a))$.

We claim that $\delta'(q_0', w) = h$ where $h(q) := \tau(\delta(q, w^R))$. We prove this by induction on the length of w.

Base case: $|w| = 0$. Then $w = \epsilon$, and $\delta'(q_0', \epsilon) = q_0' = (q \mapsto \tau(q))$, so since $\delta(q, \epsilon^R) = \delta(q, \epsilon) = q$, the base case is proved.

Now assume that our claim is true for $|w| = n$; we wish to prove it for $|w| = n + 1$. Write $w = xa$ where $|x| = n$ and $|a| = 1$. Then

$$\delta'(q_0', xa) = \delta'(\delta'(q_0', x), a)$$

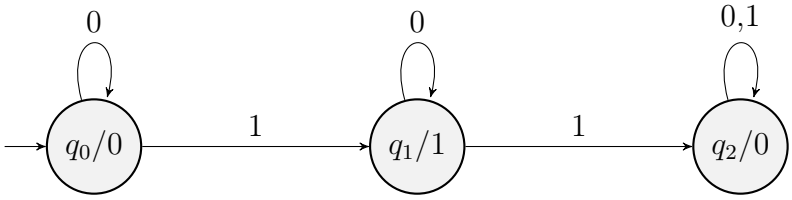
$$= \delta'(g, a) =: h$$

where $g := \delta'(q_0', x)$. By induction, $g(q) = \tau(\delta(q, x^R))$. Let's figure out what h is. For all $q \in Q$,

$$\begin{aligned} h(q) &= g(\delta(q, a)) \\ &= \tau(\delta(\delta(q, a), x^R)) \\ &= \tau(\delta(q, ax^R)) \\ &= \tau(\delta(q, (xa)^R)) \\ &= \tau(\delta(q, w^R)). \end{aligned}$$

This completes the proof. □

Example 3.2. Consider the characteristic sequence of the powers of two: $c_0 = 0$, $c_1 = 1$, $c_2 = 1$, $c_3 = 0$, $c_4 = 1$, $c_5 = 0$, etc. It's generated by the following automaton.

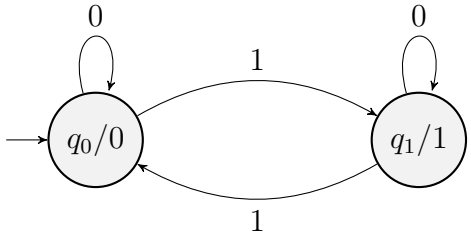


Applying the word reversal construction to this automaton gives a new, more complicated automaton that I will describe in words. Its start state, say state Q_0, is the function $q_0 \mapsto 0$, $q_1 \mapsto 1$, $q_2 \mapsto 0$ with output 0. Its second state, say Q_1, is the function $q_0 \mapsto 1$, $q_1 \mapsto 0$, $q_2 \mapsto 0$ with output 1. Its accepting state, say Q_2, is the function $q_0 \mapsto 0$, $q_1 \mapsto 0$, $q_2 \mapsto 0$ with output 0. There is an arrow from Q_0 to itself labelled 0, an arrow from Q_0 to Q_1 labelled 1, an arrow from Q_1 to itself labelled 0, an arrow from Q_1 to Q_2 labelled 1, and an arrow from Q_2 to itself labelled $0, 1$.

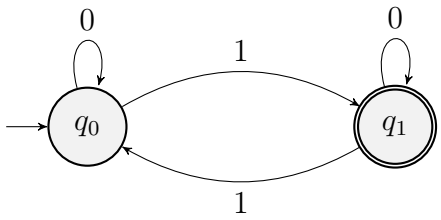
We now get a review of the different types of automata. There are DFAs, which can be thought of as DFAOs with output alphabet $\{0, 1\}$. There are NFAs, which are a generalization of DFAs. DFAs correspond to regular languages (which in Europe are called rational languages) and to regular expressions.

A DFA is like a DFAO, but the output mapping given by Δ and τ is replaced by a subset $F \subseteq Q$ of "final" or "accepting" states. To think of a DFA as a DFAO with output alphabet $\{0, 1\}$, we set $\tau(q) = 1$ if $q \in F$ and $\tau(q) = 0$ if $q \notin F$.

The Thue–Morse sequence is computed by the following DFAO.



The corresponding DFA is as follows.



The language of this DFA is

$$\{x \in \{0,1\}^* \mid |x|_1 \equiv 1 \pmod 2\}.$$

Now we discuss languages. Suppose L_1, L_2, and L are languages (sets of strings). Then we can form their union $L_1 \cup L_2$, their concatenation

$$L_1 L_2 := \{xy \mid x \in L_1, y \in L_2\},$$

the n-fold concatenation given by $L^0 := \{\epsilon\}$ and $L^n := LL^{n-1}$ for $n \geq 1$, and the *Kleene star*

$$L^* := \bigcup_{i \geq 0} L^i = \{x_1 x_2 \cdots_i \mid i \geq 0, x_i \in L \text{ for all } i\}.$$

In the order of operations, $*$ has the highest precedence, then concatenation, then union.

We define the *fibre* of a sequence $(a_n)_{n \geq 0}$ over a finite alphabet Δ to be the sets of the form

$$I_d^k := \{(n)_k \mid a_n = d\}.$$

Then we have the following theorem.

Theorem 3.3. *Let $(a_n)_{n \geq 0}$ be a sequence over a finite alphabet Δ. Then $(a_n)_{n \geq 0}$ is k-automatic if and only if each of the fibres I_d^k is regular for $d \in \Delta$.*

Proof. We give a sketch.

Suppose $(a_n)_{n \geq 0}$ is generated by a DFAO. For each d, turn it into a DFA by making the final states those q for which $\tau(q) = d$. This proves the forward direction.

If each I_d^k is regular, it is recognized by a DFA. Set $M_d := (Q_d, \Sigma, \delta_d, q_{0,d}, F_d)$. Create $M := (Q, \Sigma, \delta, q_0, \Delta, \tau)$ where $\Delta = \{d_1, d_2, \ldots, d_i\}$, $Q := Q_{d_1} \times Q_{d_2} \times \cdots \times Q_{d_i}$,

$$\delta([p_1, p_2, \ldots, p_i], a) := [\delta_{d_1}(p_1, a), \ldots, \delta_{d_i}(p_i, a)],$$

and $\tau([p_1, p_2, \ldots, p_i]) := d_j$ where p_j is the unique final state among $p_1, p_2, \ldots, p_i$. This completes the sketch. □

We now discuss morphisms. A *morphism* is a map $h : \Sigma^* \to \Delta^*$ such that $h(xy) = h(x)h(y)$ for all $x, y \in \Sigma^*$. It suffices to define h on Σ. (Note that $h(\epsilon) = \epsilon$.)

Example 3.4. The morphism given by $\mu(0) := 01$, $\mu(1) := 10$ is called the *Thue–Morse morphism* because it generates the Thue–Morse word. The morphism $\varphi(0) := 01$, $\varphi(1) := 0$ is called the *Fibonacci morphism.* (Can you see why?)

For $k \geq 1$, a morphism is said to be *k-uniform* if $|h(a)| = 1$ for all $a \in \Sigma$. For example, the morphism μ is 2-uniform and φ is not k-uniform for any k. For example, the Thue–Morse morphism is 2-uniform, and the Fibonacci morphism is not k-uniform for any k.

A morphism $h : \Sigma^* \to \Sigma^*$ is called *prolongable* if:

(a) there exists $a \in \Sigma^*$ such that $h(a) = ax$ for some $x \in \Sigma^*$, and
(b) $h^i(x) \neq \epsilon$ for all $i \geq 0$.

If h is prolongable on a, then $\lim_{n\to\infty} h^n(a)$ exists in the sense that there exists a unique infinite word having $h^i(a)$ as a prefix for each $i \geq 0$. Moreover, this limit is an infinite word of the form

$$w = axh(x)h^2(x)h^3(x)\ldots,$$

and we have $h(w) = w$, so w is a fixed point of h. Indeed,

$$\begin{aligned} h(w) &= h(a)h(x)h^2(x)\ldots \\ &= axh(x)h^2(x)h^3(x)\ldots \\ &= w. \end{aligned}$$

We often write $h^\omega(a) = w$.

The following theorem is known as *Cobham's little theorem.*

Theorem 3.5. *Let $(a_n)_{n\geq 0}$ be a sequence. Then (a_n) is k-automatic if and only if there exists a k-uniform morphism h prolongable on some letter a, a finite alphabet Δ, and a coding (1-uniform morphism) τ such that*

$$(a_n)_{n\geq 0} = \tau(h^\omega(a)).$$

We will prove this theorem next class.

Example 3.6. Consider the morphism $g(0) := 01$, $g(1) := 12$, $g(2) := 22$ and the coding $\tau(0) := 0$, $\tau(1) := 1$, $\tau(2) := 0$. We have

$$g^\omega(0) = 0112122212222222\ldots$$

Applying the coding τ gives the sequence

$$011010001000\ldots.$$

This is the characteristic sequence of the powers of 2. Now, the automaton we built for the automaton recognizing this sequence had three states. This sequence has three letters. Coincidence?

4 Jan. 16, 2020.

Problem Set 1 is up on the course website and is due in two weeks. We can hand in a problem set in class or via email if necessary. Outside sources should be cited.

Today we will prove Cobham's little theorem, and then we will see another characterization of automatic sequences in terms of something called the kernel.

Suppose h is a k-uniform morphism with some fixed point w. If $w = a_0a_1a_2\ldots$ where the a_i's are in the alphabet Σ, then $h(a_i) = a_{ki}a_{ki+1}\ldots a_{ki+k-1}$. We now state Cobham's little theorem.

Theorem 4.1. *Let $b = (b_n)_{n\geq 0}$ be a sequence over Δ. Then b is k-automatic if and only if there exist a finite alphabet Γ, a k-uniform morphism $h : \Gamma^* \to \Gamma^*$ with $h(a) = ax$ for some a, and a coding $\tau : \Gamma \to \Delta$ such that $b = \tau(h^\omega(a))$.*

Proof. Suppose that b is k-automatic. Then there exists a DFAO $M = (Q, \Sigma_k, \Delta, \delta, q_0, \tau)$ accepting it. Take $\Gamma := Q$, and let

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

Assume without loss of generality that $\delta(q_0, 0) = q_0$. (We can always modify our DFAO to make this the case.) Take $a := q_0$. Define $w := h^\omega(a)$. We will prove that

$$\delta(q_0, y) = w[[y]_k] \quad (*)$$

for all $y \in \Sigma^*$. (This notation $w[[y]_k]$ means the y in base k^{th} index of the word w.)

Our base case is $|y| = 0$. Then $\delta(q_0, y) = \delta(q_0, \epsilon) = q_0 = a$, by definition of transition functions. We also have $w[0] = a$, so the base case is proved.

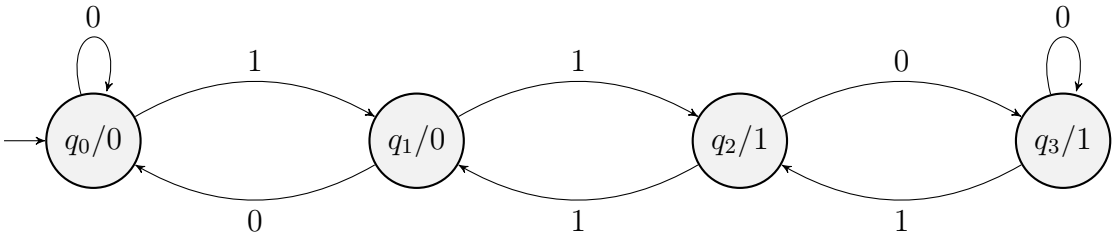
Now assume that (*) holds for all y with $|y| < i$; we will prove it for $|y| = i$. If $|y| = i$, we can write $y = xa$ with $a \in \Sigma$. Then

$$\begin{aligned}
\delta(q_0, y) = \delta(q_0, xa) &= \delta(\delta(q_0, x), a) \\
&= \delta(w[[x]_k], a) \\
&= h(w[[x]_k])[a] \\
&= (w[k[x]_k \ldots k[x]_k + k - 1])[a] \\
&= w[k[x]_k + a] \\
&= w[[xa]_k] \\
&= w[[y]_k].
\end{aligned}$$

We skip the other direction; it's in the course notes on Professor Shallit's website. □

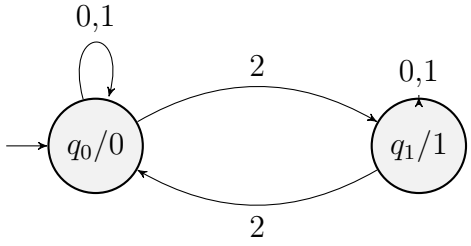
There is a correspondence between k-DFAOs and pairs (τ, h) where τ is a coding applied to a fixed point of morphism h. Under this correspondence, states Q correspond to domain and range elements of h, the transition function δ corresponds to the image of h on a letter, and the output mapping corresponds to the coding τ.

Example 4.2. Recall the Rudin–Shapiro sequence $a_{11}(n) \pmod 2$ where $a_{11}(n)$ is the number of binary 11's in $(n)_2$. Recall also that it is given by the following four-state DFAO.



The corresponding morphism and coding are given by $h(q_0) := q_0q_1$, $h(q_1) := q_0q_2$, $h(q_2) := q_3q_1$, $h(1_3) := q_3q_2$ and $\tau(q_0) := 0$, $\tau(q_1) := 0$, $\tau(q_2) := 1$, $\tau(q_3) := 1$.

Example 4.3. The following sequence, related to the Thue–Morse sequence, is known as the *Mephisto waltz sequence*. It is generated by the morphism $g(0) := 001$, $g(1) := 110$, with infinite word $g^\omega(0) = 001001110\ldots$. It is recognized by the following DFAO.



Note that there are 3 letters in the DFAO's alphabet because the morphism g is 3-uniform.

Definition 4.4. Let $a := (a(n))_{n\geq 0}$ be a sequence. The *k-kernel* of $(a(n))$ is the set of subsequences

$$K_k(a) := \left\{(a(k^i n + j))_{n\geq 0} \mid i \geq 0, 0 \leq j < k^i\right\}.$$

Example 4.5. Let $k := 2$. Then

$$K_2(a) := \{(a(n))_{n\geq 0}, (a(2n))_{n\geq 0}, (a(2n+1))_{n\geq 0}, (a(4n))_{n\geq 0},$$

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

The following result is known as *Eilenberg's theorem*. It is one of the earliest results in the field of automatic sequences.

Theorem 4.6. *Let $a = (a(n))_{n\geq 0}$ be a sequence over a finite alphabet Δ. Then a is k-automatic if and only if $K_k(a)$ is finite.*

Proof. Suppose a is k-automatic, hence computed by a k-DFAO. Then a is also computed by a k-DFAO in the opposite order, so by a machine $(Q, \Sigma_k, \delta, q_0, \Delta, \tau)$ where $a(n) = \tau(\delta(q_0, w^R))$ for all w such that $[w]_k = n$.

Let x be such that $[x]_k = k^i n + j$. If $n \neq 0$, we can write $x = wy$ where $|y| = i$ and $[y]_k = j$. Then

$$\begin{aligned} a(k^i n + j) &= \tau(\delta(q_0, x^R)) \\ &= \tau(\delta(q_0, y^R, w^R)) \\ &= \tau(\delta(\delta(q_0, y^R), w^R)) \\ &= \tau(\delta(q, w^R)). \end{aligned}$$

Since $q = \delta(q_0, y^R)$, the k-kernel is finite. For the $n = 0$ case, see the notes. Thus, the DFAO $(Q, \Sigma_k, \delta, q, \Delta, \tau)$ computes the subsequence $(a(k^i n + j))_{n \geq 0}$. I will describe in words what the corresponding DFAO looks like in the $k = 2$ case. It has five states. The start state is $(a(n))_{n \geq 0}$ with output $a(0)$; call this state q_0. The other four states (call them q_1, q_2, q_3, q_4 in that order) are $(a(2n))_{n \geq 0}$ with output $a(0)$, $(a(4n+2))_{n \geq 0}$ with output $a(2)$, $(a(2n+1))_{n \geq 0}$ with output $a(1)$, and $(a(4n+1))_{n \geq 0}$ with output $a(1)$. There is an arrow labelled 0 from q_0 to q_1, an arrow labelled 1 from q_1 to q_2, an arrow labelled 1 from q_0 to q_3, and an arrow labelled 0 from q_3 to q_4. Note that the output associated with state $(a(k^i n + j))_{n \geq 0}$ is $a(j)$. □

We give a heuristic procedure for guessing whether a given sequence is k-automatic. For $k = 2$, start with $(a(n))_{n \geq 0}$, then $(a(2n))_{n \geq 0}$. If a sequence appears to match a previously computed one, say they are equal and stop examining this subsequence. Continue along the different subsequences $a(2n+1), a(4n)$, etc.

Example 4.7. Let $p_1 := 0$ and $p_{n+1} := p_n 0 \overline{p_n^R}$ (where $\overline{x}$ denotes the complement of x, so $\overline{0} = 1$ and $\overline{1} = 0$). (This is known as the *paperfolding sequence*.) Then

$$p_1 = 0, p_2 = 001, p_3 = 0010011, \text{ etc.}$$

Let $p := \lim_{n \to \infty} p_n = p_1 p_2 p_3 \ldots$. Let $p_0 := 2$ as well. Then

$$\begin{aligned} (p_n) = p &= 2001001100011011 \ldots, \\ (p_{2n}) &= 20010011 \cdots = (p_n), \\ (p_{2n+1}) &= 01010101 \ldots, \\ (p_{4n+1}) &= 00000 \ldots, \\ (p_{4n+3}) &= 11111 \ldots, \end{aligned}$$

and additionally we have $p_{8n+1} = p_{4n+1} = p_{8n+5}$ and $p_{6n+3} = p_{4n+3} = p_{8n+7}$.

So then we can make the following conjectured automaton for this sequence. Its start state is $q_0 = (p_n)_{n \geq 0}$ with output 0, and its other states are $q_1 := (p_{2n+1})_{n \geq 0}$ with output 0,

$q_2 = (p_{4n+1})_{n\geq 0}$ with output 0, and $q_3 = (p_{4n+3})_{n\geq 0}$ with output 1. There is an arrow labelled 1 from q_0 to q_1, an arrow labelled 0 from q_1 to q_2, an arrow labelled 1 from q_1 to q_3, an arrow labelled $0, 1$ from q_2 to q_2, and an arrow labelled $0, 1$ from q_3 to q_3.

From this automaton, we can get the following heuristic formula for the paperfolding sequence. Let $n = 2^i(2j + 1)$ for $n > 0$. Then

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

When you repeatedly fold a piece of paper in half, then unfold, you get a sequence of valleys and hills in the paper, which obeys the paperfolding sequence; that's where it gets its name. We now do this as a class.

We will now prove our conjectural formula for the paperfolding sequence by induction on n.

Proof. We give a sketch. Recall that $p_1 = 0$ and $p_{n+1} = p_n 0 \overline{p_n^R}$. Note that $|p_n| = 2^n - 1$. In the base case, $n = 1$. Then $i = 0, j = 0$, and $p_1 = 0$. If $n = 2^i$, then $j = 0$ and $p_{2^i} = 0$ for all i. Now assume the result holds for all $n' < n$. We will prove it for n.

If $2^{m-1} < n < 2^m$, then $p_m = p_{m-1} 0 \overline{p_{m-1}^R}$. Note that $p_n = \overline{p_{2^m - n}}$. We calculate that

$$\begin{aligned} 2^m - n &= 2^m - 2^i(2j + 1) \\ &= 2^i(2^{m-i} - 2j - 1) \\ &= 2^i(2r + 1) \end{aligned}$$

where $r = 2^{m-i-1} - j - 1$. Then $r \pmod 2 \neq j \pmod 2$. Comparing lengths, this completes the sketch. □

The paperfolding sequence discussed above is known in the literature as the *regular paperfolding sequence*. In the general case, we choose to introduce a hill or valley at each fold, and we have a sequence $(f_i)_{i\geq 0}$ of unfolding instructions. Then $p_0 := \epsilon$ and $p_{n+1} := p_n f_n \overline{p_n^R}$. There are uncountably many paperfolding sequences, but there are only countably many automata, so most paperfolding sequences are not automatic. The ones that are automatic are precisely the ones where the sequence of unfolding instructions is eventually periodic.

Professor Shallit mentions the following caution for our heuristic for finding whether sequences are automatic. Start with the Thue–Morse sequence

$$t = 0110100110010110100101100110100 1 \ldots$$

Look at the length of blockos of 0's and 1's: the first block has length 1, the second has length 2, the third length 1, then 1, then 2, 2, 2, 1, 1, 2, 1, 1, etc. Group these like this:

$$(121)(12221)(121)1 \ldots$$

Call this sequence $s = (s_n)_{n \geq 0}$. We can make a morphism $1 \mapsto 121$, $2 \mapsto 12221$, etc. Then (s_{16n+1}) agrees with (s_{64n+1}) on the first 1864135 terms but fails at $n = 1864135$. An open problem is to explain this. Professor Shallit has been suggesting this for like 25 years, but no one has worked on it. It would be nice to find some reasonably good bound on how many terms two elements in the 2-kernel of (s_n) can agree on before disagreeing.

5 Jan. 21, 2020.

We need to hand a 1-page description of our final project topic by February 6. There are a bunch of options on the website, and independent research is also allowed.

We discuss number theory today. We will demonstrate a connection between automatic sequences and continued fractions.

We first discuss continued fractions for real numbers. A *(simple) continued fraction* is an expression of the form

$$\alpha = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \ddots}}}$$

where $\alpha \in \mathbb{R}$, $a_i \in \mathbb{Z}$ for every i, and $a_i \geq 1$ for $i \geq 1$. We abbreviate the above continued fraction by $[a_0, a_1, a_2, \dots]$. This expression can be finite or infinite in length.

For example,

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

If $a_n \neq 1$ (except if $a_0 = 1$ and this is the only term, i.e., the case of the expansion for 1), then the continued fraction expansion is unique for rational numbers. (To see why this is required, consider $2 = [2] = [1, 1]$.) Expansions are unique for irrational numbers. "Here's where mathematicians have messed up a little bit," says Professor Shallit. By the bracket notation, we mean the rational function of the a_i's, evaluated at the values in the brackets, gives the constant α. There are other interpretations of the bracket notation, and this creates an issue for the rational function case later.

The continued fraction expansion is ultimately periodic if and only if α is a quadratic irrational. For example, $\sqrt{2} = [1, 2, 2, 2, \dots]$. What happens when you take a continued fraction and truncate it? Take, for example, $\alpha = [a_0, a_1, a_2, \dots]$ and truncate at $[a_0, a_1, \dots, a_n]$. This would be an approximation to α. How do we evaluate $[a_0, a_1, \dots, a_n]$ from "left to right"

rather than "right to left" (by just calculating the fraction)? We discuss this now.

Write $[a_0, a_1, \ldots, a_n] = p_n/q_n$. This fraction is known as a *convergent*, and the a_i's are called *partial quotients*. We can calculate p_n and q_n by the following recurrence:

$$\begin{aligned} p_{-2} &= 0, q_{-2} = 1, \\ p_{-1} &= 1, q_{-1} = 0, \\ p_n &= a_n p_{n-1} + p_{n-2}, \\ q_n &= a_n q_{n-1} + q_{n-2}, \end{aligned}$$

where $n \geq 0$. For example, we have the following approximation to π.

n	-2	-1	0	1	2	3	4
a_n			3	7	15	1	292
p_n	0	1	3	22	333	355	
q_n	1	0	1	7	106	113	

The following identity is called the Hurwitz–Frame–Kolden representation:

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

Taking the determinant of both sides gives the identity

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

Taking the transpose of both sides gives *Galois's identity* (so named because it was his first published paper):

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

The following lemma is known as the *folding lemma* for continued fractions. This is because it gives a relationship between the paperfolding sequence from last class and continued fractions.

Lemma 5.1. *Suppose*

$$p_n/q_n = [c_0, c_1, c_2, \ldots, c_n].$$

Let $w := (c_1, c_2, \ldots, c_n)$*, viewed as a word over the alphabet of integers. Then*

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

What this says is that if you start with a continued fraction and you do this "fold" to it by inserting a t in the middle, what you get is a number that is an extremely close approximation to the original number (differing only quadratically from it).

Proof. One can prove by induction that

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

and

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

Taking the transpose of the second equation gives

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

We multiply on the right of this equation by the inverse of the rightmost matrix on the left-hand side to get

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

We calculate

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

This completes the proof. □

We now discuss formal power series. If $P(X) := \sum_{i \geq 0} p_i X^i$ and $Q(X) := \sum_{i \geq 0} q_i X^i$, then

$$(P + Q)(X) = \sum_{i \geq 0} (p_i + q_i) X^i$$

and

$$(PQ)(X) = \sum_{n > 0} \left(\sum_{i+j=n} p_i q_j \right) X^n.$$

They form a ring. We can just as well consider power series in X^{-1} as in X. We now talk about formal Laurent series in X^{-1}. These are of the form

$$A(X) := \sum_{i \geq -c} a_i X^{-1}$$

where $c \in \mathbb{Z}$. An example is

$$x^2 + x + 7 + 3x^{-1} + 12x^{-2} + \dots .$$

There are a finite number of *positive exponents* but there might be infinitely many negative exponents.

We now step back to describe the algorithm for finding a continued fraction for a real number α. Set $x_0 := \alpha$. Set $\alpha_0 := \lfloor \alpha \rfloor$. Set

$$x_1 := \frac{1}{x_0 - a_0}.$$

Set $a_1 := \lfloor x_1 \rfloor$. Set

$$x_2 := \frac{1}{x_1 - a_1}.$$

Continue in this way to obtain as many a_i's as you want.

The same procedure works for formal Laurent series in X^{-1}. We first define a notion of floor. The floor is just the stuff before the decimal point of your real number. This is the stuff with a positive power of 10 in its decimal expansion. So we can do the same thing for Laurent series. Given

$$A(X) = \sum_{i \geq -c} a_i X^{-i},$$

we define

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

Then we can use the exact same algorithm to find a continued fraction representation of our Laurent series. We now do an example. Start with

$$f(X) := X^{-1} + X^{-2} + X^{-4} + X^{-8} + X^{-16} + \cdots .$$

Then $\lfloor f \rfloor = 0$. We get

$$f_1 := \frac{1}{f} = X - 1 - X^{-1} - 2X^{-1} + 3X^{-3} - 4X^{-4} + 6X^{-5} + \cdots .$$

Then

$$a_1 := \lfloor f_1 \rfloor = X - 1$$

and

$$f_2 := \frac{1}{f_1 - a_1} = X + 2 + X^{-1} + X^{-3} + 2X^{-5} + \cdots .$$

So far our continued fraction representation for f is $[0, X-1, X+2, \dots]$. We will not calculate more of it, but it goes on $[0, X-1, X+2, X, X, X-2, X, X+2, \dots]$. One can prove that all the terms are linear integer polynomials in X, which is unusual. On the course

website there is some Maple code for calculating these continued fractions. (The notation is cfps(f, X) in Maple.)

Consider the power series of the form

$$X \sum_{i \geq 0} e_i X^{-2^i}$$

where $a_0 = 0$ and $e_i = \pm 1$ for all $i \geq 1$.

Theorem 5.2. *The continued fraction expansion of a power series $f(X)$ of the above form is*

$$f = [1, \mathit{Fold}(e_1 X, -e_2 X, -e_3 X, \dots]$$

where $F_a : \{-1, 1\}^* \to \{-1, 1\}^*$ (which is not a morphism) is defined by

$$F_a(w) := (w, a, -w^R)$$

and

$$\mathit{Fold}(a_n, a_{n-1}, \dots, a_1) := F_{a_1}(F_{a_2}(F_{a_3}(\cdots(\epsilon)))).$$

Recall the identity

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

from the folding lemma. The analogous folding operation for Laurent series in X^{-1} is $[1, X] \mapsto [1, X, e_2 X, -X]$. So $e_2 X$ plays the role of t.

For example, we get

$$\sum_{i \geq 0} X^{1-2^i} = 1 + X^{-1} + X^{-3} + X^{-7} + X^{-15} + \cdots$$

$$= [1, X, -X, -X, -X, X, X, -X, -X, X, -X, -X, -X, X, X, X, -X, \dots].$$

Subbing in $X := 2$, we get

$$2 \sum_{i \geq 0} 2^{-2^i} = [1, 2, -2, -2, -2, 2, 2, -2, -2, 2, -2, -2, \dots].$$

One might object that we are not allowed to have negatives (aside from the leading term) in our real continued fractions. We can fix this using the rules $[a, 0, b] = [a + b]$ and $[a, -b, c] =$

$[a-1, 1, b-2, 1, c-1]$. Applying them we successively get

$$\begin{aligned}
[1, 2, -2, -2, -2, 2, 2, -2, -2, \dots] & \\
&= [1, 1, 1, 0, 1, -3, -2, 2, 2, -2, -2, \dots] \\
&= [1, 1, 2, -3, -2, 2, 2, -2, -2, \dots] \\
&= [1, 1, 1, 1, 1, 1, -3, 2, 2, -2, -2, \dots] \\
&= [1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 2, -2, 2, \dots] \\
&= [1, 1, 1, 1, 2, 1, 1, 1, 2, -2, 2, \dots].
\end{aligned}$$

One number with a very good approximation by rationals is *Liouville's constant*

$$\alpha = \sum_{n \geq 1} 10^{-n!}.$$

For Liouville's constant α and every t, there exist integers p, q with

$$\left| \alpha - \frac{p}{q} \right| < q^{-t}.$$

For general β, for all integers p, q there exists C with

$$\left| \beta - \frac{p}{q} \right| > \frac{C}{p^2}.$$

The earlier Laurent series example gives uncountably many (though they still form a set of measure zero) numbers having bounded partial quotients.

6 Jan. 23, 2020.

In the tower of Hanoi problem, you have three pegs and a bunch of disks that start on peg 1. The disks are sorted from smallest to largest on the peg, with the smallest at the top and the largest at the bottom. You can never put a larger disk on top of a smaller one. The goal is to get all of the disks on the last peg (peg 3). If there are N disks, the optimal solution requires $2^N - 1$ moves and can be done in that many.

The argument for the lower bound is as follows. To move N disks, you must move the N^{th} one eventually. To move the N^{th} disk, one peg (of the two pegs you're moving between) must hold $N-1$ and the other must hold none. So the optimal sequence of moves is to (i) move $N-1$ disks to one peg somehow, then (ii) move disk N, and then (iii) move $N-1$ disks to that peg somehow. Part (i) requires $\geq 2^N - 1$ moves, part (ii) requires 1 move, and part (iii) requires $\geq 2^N - 1$ moves, so in total we need $\geq 2^N - 1$ moves. This gives the lower bound.

Here is a goal. Given t, determine what the t^{th} move in the optimal solution is. Then one can ask two questions to specify the answer. From which peg to which other peg are they moving? Which disk is being moved?

"The" optimal solution is defined to move peg 1 to peg 2 if N Is odd and from peg 1 to peg 3 if N is even. Then the solution for N moves is a prefix of the solution for $N-1$ moves.

If the moves are numbered $i = 0, 1, 2,$ and so forth, then on move t, the disk moved is $\nu_2(t+1)+1$ where $\nu_2(n)$ is the exponent of the highest power of 2 dividing n. (This sequence is sometimes called the *ruler function.*) This answers the second question.

The first question is harder to answer. Let's use the following notation a denotes the move $1 \to 2$, b the move $2 \to 3$, c the move $3 \to 1$. Let $\bar{a}, \bar{b}, \bar{c}$ be their inverses (so $\bar{a} : 2 \to 1$, etc.). For 1 disk the optimal solution is a. For 2 disks, it is $a\bar{c}b$. For 3 disks, it is $a\bar{c}ba\bar{c}\bar{b}a$.

Let H_n be the optimal solution for n disks. Then $H_{2i} = H_{2i-1}\bar{c}\sigma(H_{2i-1})$ for all $i \geq 1$ and $H_{2i+1} = H_{2i}a\sigma^{-1}(H_{2i})$ for all $i \geq 0$. Here σ is defined by $\sigma(a) := b$, $\sigma(b) := c$, $\sigma(c) := a$ and $\sigma(\bar{a}) := \bar{b}$, $\sigma(\bar{b}) = \bar{c}$, $\sigma(\bar{c}) = \bar{a}$. Let

$$H := \lim_{n\to\infty} H_n.$$

Then H is 2-automatic. Define the morphism φ by $\varphi(a) := a\bar{c}$, $\varphi(b) := c\bar{b}$, $\varphi(c) := b\bar{a}$, $\varphi(\bar{a}) := ac$, $\varphi(\bar{b}) := cb$, and $\varphi(\bar{c}) := ba$. Our goal is to prove that $\varphi(H) = H$.

Lemma 6.1. *We have*

$$\varphi(\sigma(w)) = \sigma^{-1}(\varphi(w))$$

and

$$\varphi(\sigma^{-1}(w)) = \sigma(\varphi(w))$$

for all w.

Proof. We can check it holds for $w = a, b, c, \bar{a}, \bar{b}, \bar{c}$. Then since all these maps are morphisms, it holds in general. □

Lemma 6.2. *We have*

$$H_{2i+1} = \varphi(H_{2i})a$$

and

$$H_{2i+2} = \varphi(H_{2i+1})b$$

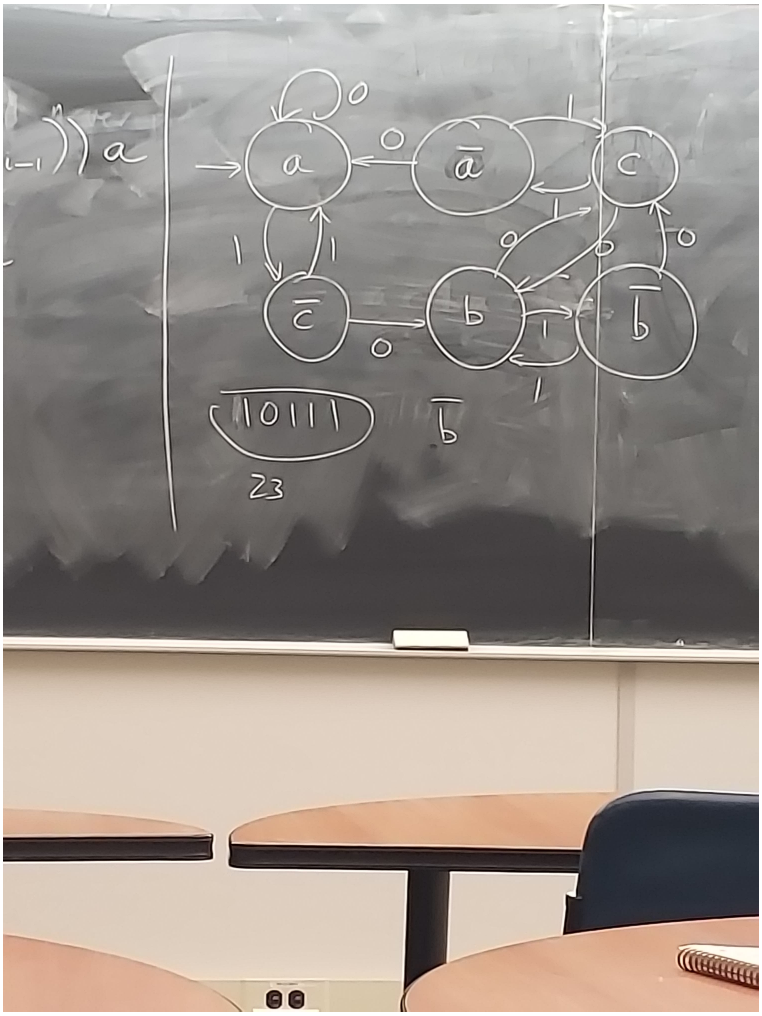
for all $i \geq 0$.

Proof. By induction on i, we get

$$\begin{aligned} H_{2i+1} &= H_{2i}a\sigma^{-1}(H_{2i}) \\ &= \varphi(H_{2i-1})ba\sigma^{-1}(\varphi(H_{2i-1})b) \\ &= \varphi(H_{2i-1})\varphi(\bar{c})\varphi(\sigma(H_{2i-1}))a \\ &= \varphi(H_{2i})a. \end{aligned}$$

□

Here is the automaton recognizing the optimal move sequences if you input the number of the move you want as a binary sequence. (For example, for move 23 in the optimal sequence, you would input 10111.)



Now we consider a different problem. Let $s_2(n)$ be the sum of the bits of n in base 2. Newman proved in the 1960s that $s_2(3n)$ is "usually" even. This can be done using the following general result about automatic sequences.

Theorem 6.3. *If $(u(n))_{n\geq 0}$ is k-automatic, then so is $(u(an+b))_{n\geq 0}$ for any $a, b \geq 0$.*

We give two proofs. The first is "automata-theoretic", the second "language-theoretic".

Proof. We use the k-kernel characterization of automaticity. Assume without loss of generality that $a \geq 1$. Let

$$K_k(u) = \{(u_1(n)), (u_2(n)), \ldots, (u_r(n))\}.$$

where $(u_1(n)), \ldots, (u_r(n))$ are some sequences. Let

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

Let $v(n) := u(an + b)$. We claim that $K_k(v) \subseteq S$.

Consider $v(k^e n + j))_{n \geq 0}$ for $e \geq 0$ and $0 \leq j < k^e$. Then $K_k(v)$ consists precisely of the sequences of this form. We divide $ja + b$ by k^e, obtaining a quotient d and a remainder f. From the properties of division, we have

$$ja + b = dk^e + f$$

where $0 \leq f < k^e$ and $0 \leq d < a + b$. We have

$$v(k^e n + j) = u(a(k^e n + j) + b)$$

$$= u(k^e(an + d) + f).$$

There exists i such that $(u(k^e n + f))_{n \geq 0} = (u_i(m))_{m \geq 0}$. Then

$$v(k^e n + j) = u_i(an + d)$$

and

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

This proves that $K_k(v) \subseteq S$, so we are done. □

We now introduce the notion of transducer before giving the second proof. A *finite-state transducer* is non-deterministic automaton with inputs and outputs on transitions. The inputs and outputs can be arbitrary words. Each transition is labelled with a label of the form x/w where x is the input and w is the output. Also, the input must end in a final state of the transducer.

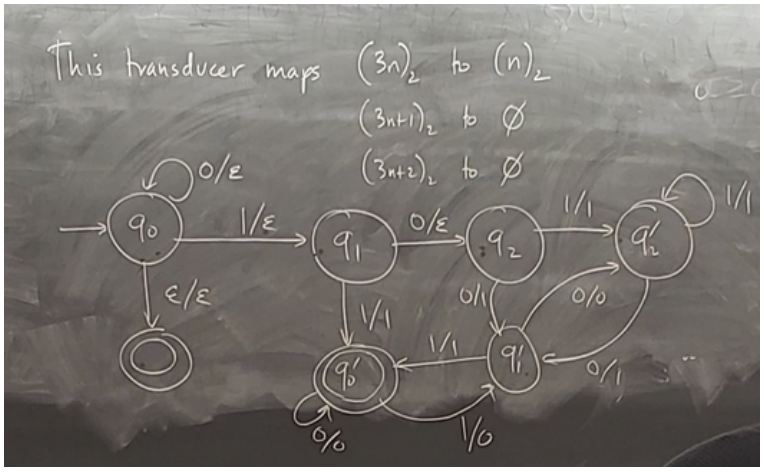
The following result is known as *Nivat's theorem.*

Theorem 6.4. *If T is a finite-state transducer and L is a regular language, then $T(L)$ and $T^{-1}(L)$ are regular. (Here T^{-1} means what you would expect it to mean.)*

Let $u := (u(n))_{n \geq 0}$ be a sequence. For each $d \in \Delta$, the d^{th} *fibre* is defined by

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

The following transducer maps $(3n)_2$ to $(n)_2$, $(3n + 1)_2$ to $\emptyset$, and $(3n + 2)_2$ to $\emptyset$. In this picture, $q - i$ is the carry of i, and a prime next to the state (e.g., q_1') indicates that a 1 has been output.



Theorem 6.5. *Let $a \geq 1$. Let $(u(n))_{n\geq 0}$ be a sequence, and suppose that each $(u(an+b))_{n\geq 0}$ is k-automatic for $0 \leq b < a$. Then $(u(n))_{n\geq 0}$ is k-automatic.*

In what follows, we will apply a transducer to an automatic sequence u. We require the transducer to be *functional*, i.e., 1 input gives 1 output and there is no notion of final states.

Let $t := 0110100110010110\ldots$. Consider the *period-doubling sequence* of t, $010001010100010\ldots$. Its i^{th} entry is defined to be 1 if $t_i = t_{i+1}$ and 0 otherwise. The morphism which, when applied to t, generates its period-doubling sequence is $0 \mapsto 01$, $1 \mapsto 00$.

Theorem 6.6. *If T is a uniform transducer, i.e., 1 letter of input always gives t letters of output for some t that works for every letter, then $T(u)$ is k-automatic if u is k-automatic.*

Consider the characteristic sequence of the powers of 2: $01101000100\ldots$. Call this sequence a and apply the morphism h given by $1 \mapsto 10$, $0 \mapsto 0$ to get the the sequence $h(a)$ given by $01010010000100\ldots$. The sequence $h(a)$ turns out not to be 2-automatic! (We will prove this next time.) This means that we cannot relax the requirement of uniformity on our transducer in the previous theorem because any morphism f (including, in particular, our morphism h) can be viewed as a transducer with a single state that on input a gives $f(a)$.

7 Jan. 28, 2020.

Let (a_n) be the characteristic sequence of the powers of 2, starting with $a_0 = 0$. Let $a := (a_n)_{n\geq 0}$. Let $h : 1 \mapsto 10, 0 \mapsto 0$, and consider

$$b := h(a) = 01010^2 10^4 10^8 \ldots.$$

Then b is the characteristic sequence of $(2^r + r)_{r\geq 0}$. We will show that b is not 2-automatic using the *pumping lemma* for regular languages. This is the following.

Lemma 7.1. *If L is a regular language, then there exitsts $n = n(L)$ such that for every $z \in L$ such that $|z| \geq n$, there exists a decomposition $z = uvw$ where $|uv| \leq n$ and $|v| \geq 1$ such that for every $i \geq 0$, $uv^i w \in L$.*

There is a nice proof of this in Sipser's book on automata, which Professor Shallit now states. Now, the fibre $I_1(b)$ is

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

where the subscript 2 refers to base 2. If b is 2-automatic, then $I_1(b)$ is a regular language. Now, $(2^r + r)_2$ is of the form

$$(2^r + r)_2 = 10^{r-\lfloor \log_2 r \rfloor - 1}(r)_2.$$

The length of $(r)_2$ is

$$|(r)_2| = \lfloor \log_2 r \rfloor + 1$$

for $r \geq 1$. Now, if $I_1(b)$ is regular, then so is

$$1^{-1}I_1(b) = \{0^{r-\lfloor \log_2 r \rfloor - 1}(r)_2 \mid r \geq 1\}.$$

Let n be the pumping length for $1^{-1}I_1(b)$ (if it exists), and let $r \geq 2n$. Decompose the word $0^{r-\lfloor \log_2 r \rfloor - 1}(r_2)$ as uvw where $|uv| \leq n$ and $|v| \geq 1$. Then $uv^2w \notin 1^{-1}I_1(b)$, so $1^{-1}I_1(b)$ is not a regular language, so $I_1(b)$ is not a regular language, so b is not 2-automatic.

We now consider deterministic transducers operating on infinite words (so all states are final). Transducers can basically only remove information (like a transducer that turns every 1 into a 0 and preserves all 0's) or add a finite amount of information (like prepending each 0 with a certain finite prefix). This motivates the following definition.

Definition 7.2. Let u and v be infinite words. We say $u \geq v$ if there exists a transducer T such that $T(u) = v$.

We claim this gives a partial order on infinite words. (This requires us to check transitivity.) One can also prove that under this partial order, every sequence is above every eventually periodic sequence. (It is not hard: just use the same prefix for each period, and delete the rest of the information in the word.)

Endrullis and Hendriks gave the following definition.

Definition 7.3. An infinite word u is an *atom* if whenever $u \geq v$, then either v is eventually periodic or $v \geq u$.

They proved the following result.

Theorem 7.4. *The sequence*

$$0101^201^301^4\ldots$$

is an atom.

Endrullis and Hendriks then posed the following open problem, which Professor Shallit does not believe is that hard: Is the Thue–Morse word t an atom? Endrullis and Hendriks suggest the answer is no. Their proferred example, which no one has been able to disprove, is you start with the Thue–Morse sequence, apply the periodic doubling map that sends t_i to 1 if

$t_i = t_{i+1}$ and sends it to 0 otherwise (to get the period-doubling sequence), and then remove every third symbol in the resulting word, which begins $0100\ldots$. Can you get back to the Thue–Morse word from this sequence using a transducer? This is an open problem, and it seems like it should be easy (or maybe some other transformation works), but we do not know the answer.

Theorem 7.5. *If $a = (a_n)_{n\geq 0}$ is a sequence and $k \geq 2$, then a is k^i-automatic for $i \geq 1$ if and only if a is k^j-automatic for $j \geq 1$.*

Proof. It suffices to take $j = 1$. First, suppose a is k^i-automatic. Then the fibre

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

is regular. If $|w| = i$ and $[w]_k = c$ where $0 \leq c < k^i$, let the morphism h be defined by $h : c \mapsto w$. Let rlz mean "remove leading zeroes". Then

$$\mathrm{rlz}(h(I_d)) = \{(n)_k \mid a_n = d\}$$

is regular, so a is k-automatic.

Conversely, suppose a is k-automatic. By Cobham's little theorem, there exist a k-uniform morphism g, a coding τ, and a letter c such that

$$a = \tau(g^\omega(c)).$$

Thus, $a = \tau(h^\omega(c))$ where $h(d) := g^i(d)$ for all d. Then h is k^i-uniform, so by Cobham's little theorem, a is k^i-automatic. □

Example 7.6. Let μ be the Thue–Morse morphism $0 \mapsto 01$, $1 \mapsto 10$. Then $t = \mu(t)$ where t is the Thue–Morse word. But also we have the morphism μ^2 given by $0 \mapsto 0110$ and $1 \mapsto 1001$, and we have that $t = \mu^2(t)$.

Recall that the notation $K[X]$ denotes the ring of polynomials over a field K, $K[[X]]$ denotes the ring of formal power series over K, and $K((X))$ denotes the field of formal Laurent series over K.

In Maple, the command for formal Laurent series is series$(p/q, x = 0, 10)$;. For formal Laurent series in X^{-1}, use $x =$ infinity instead of $x = 0$.

We say $\alpha \in \mathbb{R}$ is *algebraic* over $\mathbb{Q}$ if there exist rationals $a_0, a_1, \ldots, a_n$, not all zero, such that

$$\sum_{0\leq i\leq n} a_i \alpha^i = 0.$$

If it is not algebraic, α is said to be *transcendental*. (For example, $3/5$ and $\sqrt{2}$ are algebraic while π is transcendental.)

Similarly, we say a formal power series

$$g(X) = \sum_{i \geq 0} g_i X^i$$

is *algebraic* over $K(X)$ if there exists $n \geq 0$ and $a_0(X), \ldots, a_n(X)$, not all zero, such that

$$\sum_{0 \leq i \leq n} a_i(X) g(X)^i = 0.$$

Example 7.7. Let

$$f(x) = 1 + x + 2x^2 + 5x^3 + 14x^4 + 42x^5 + \cdots$$

$$= \sum_{n \geq 0} c_n x^n$$

where

$$c_n := \frac{\binom{2n}{n}}{n+1}$$

is the n^{th} Catalan number. Then $f(x)$ satisfies

$$xf^2 - f + 1 = 0.$$

Thus,

$$f(x) = \frac{1 - \sqrt{1 - 4x}}{2x}$$

as a formal power series.

Example 7.8. The power series

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

is transcendental in $\mathbb{Q}[[x]]$. One can prove that the coefficients of an algebraic power series cannot decrease as $1/n!$, so that is why. This by itself does not imply the transcendence of e, nor is it implied by it.

Professor Shallit remarks that these notions are where the term "transcendental function" comes from.

Let $K := \mathrm{GF}(q)$ where $q = p^n$, p is prime, and $n \geq 1$. (The "GF" stands for "Galois field". This is just the finite field of order q, and it is often denoted $\mathbb{F}_q$ instead.)

Of course, $\mathrm{GF}(p)$ is arithmetic modulo p, but $\mathrm{GF}(p^n)$ is not arithmetic mod p^n for $n > 1$. Instead, in the latter case we find an irreducible polynomial $f(x)$ of degree n over $\mathrm{GF}(p)$ and then do arithmetic mod p and mod f to get $\mathrm{GF}(p^n)$.

Example 7.9. For $p = 2$, $n = 2$, we can take $f(x) = x^2 + x + 1$. By calculating the multiplication table of $\{0, 1, x, x+1\}$, we can see that $\mathrm{GF}(4) \simeq \{0, 1, x, x+1\}$.

Note that in $\mathrm{GF}(p)$, we have $a^p = a$ for any constant a, by Fermat's little theorem, but we also have $X^p \neq X$ for the indeterminate X. Also, the "freshman's dream" $(a+b)^p = a^p + b^p$ is true.

Example 7.10. Fix $q = p^n$ where p is prime and $n \geq 1$. We consider examples of algebraic formal power series in $\mathrm{GF}(q)[[x]]$. For $q = 2$, an example is

$$f(x) = x + x^2 + x^4 + x^8 + \cdots$$

because

$$f^2 = x^2 + x^4 + x^8 + \cdots,$$

so

$$f^2 + f - x = 0$$

(or equivalently, $f^2 + f + x = 0$).

Example 7.11. Consider the Rudin–Shapiro sequence. It is defined by letting r_n be the number of 11's in $(n)_2$ mod 2. Then $r_{2n} = r_n$, $r_{4n+1} = r_n$, and $r_{4n+3} = 1 - r_{2n+1}$. We define the power series

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

where

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

Before we had $R = R^2 + XS^2$, so $XS^2 = R - R^2 = R + R^2$. Thus,

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

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

Thus,

$$R = R^2 + X\left(\frac{X}{(1+X)^2} + R\right)^2,$$

and finally we obtain

$$(1+X)^5R^2 + (1+X)^4R + X^3 = 0.$$

8 Jan. 30, 2020.

Today we discuss the proof of Christol's theorem. Apparently the proof in the course textbook is slightly wrong.

Christol's theorem says (roughly) that a sequence $(a_n)_{n\geq 0}$ is q-automatic (for $q = p^n$, p prime, $n \geq 1$) if and only if the formal power series

$$\sum_{n\geq 0} a_n X^n$$

is algebraic over $\mathrm{GF}(q)(X)$.

This statement is not quite correct because the way we defined automatic sequences, they can be over any alphabet, whereas in $\mathrm{GF}(q)(X)$, the coefficients have some actual meaning. We get around this by enlarging the finite field to include the entire alphabet of our automatic sequence. We now give a precise statement.

Theorem 8.1. *Let Δ be a finite set. Let $\underline{a} = (a_i)_{i\geq 0}$ be a sequence taking values in Δ. Let $p \geq 2$ be prime. Then $\underline{a}$ is p-automatic if and only if there exists $n \geq 1$ and an injective map $\beta : \Delta \to GF(p^n)$ such that*

$$\sum_{i\geq 0} \beta(a_i) X^i$$

is algebraic over $GF(p^n)(X)$.

We will use the *Cartier operators*, which are maps from Laurent series to Laurent series. Let

$$A(X) := \sum_{i\geq -n} a_i X^i$$

Then the Cartier operator Λ_r is defined by

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

We will mostly omit the subscript of this summation because if i is too small, the corresponding a_i's will vanish anyway. Now we have a couple of lemmas.

Lemma 8.2. *With the above notation,*

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

Proof. We have

$$\begin{aligned} A(X) &= \sum_i a_i X^i \\ &= \sum_{0 \leq r < q} \sum_i a_{qi+r} X^{qi+r} \\ &= \sum_{0 \leq r < q} X^r \sum_i a_{qi+r} X^{qi} \\ &= \sum_{0 \leq r < q} X^r \left(\sum_i a_{qi+r} X^i \right)^q \\ &= \sum_{0 \leq r < q} X^r \Lambda_r(A)^q. \end{aligned}$$

□

Lemma 8.3. *Let G, H be Laurent series over GF(q). Then $\Lambda_r(G^q H) = G\Lambda_r(H)$.*

Proof. Let

$$G(X) := \sum_k g_k X^k, H(X) := \sum_j h_j X^j.$$

Then

$$\begin{aligned} \Lambda_r(G^q H) &= \Lambda_r \left(\left(\sum_k g_k X^k \right)^q \left(\sum_j h_j X^j \right) \right) \\ &= \Lambda_r \left(\left(\sum_k g_k X^{kq} \right) \left(\sum_j h_j X^j \right) \right) \\ &= \Lambda_r \left(\sum_i X^i \sum_{i = qk+j, k, j} g_k h_j \right) \\ &= \sum_i X^i \sum_{qi+r = qk+j, k, j} g_k h_j \end{aligned}$$

$$= \sum_k X^{k+t} \sum_t g_k h_{qt+r}$$
$$= \sum_k g_k X^k \sum_t h_{qt+r} X^t$$
$$= \left(\sum_k g_k X^k \right) \left(\sum_t h_{qt+r} X^t \right)$$
$$= G\Lambda_r(H).$$

□

We think of $\mathrm{GF}(q)((X))$ as a vector space over $\mathrm{GF}(q)(X)$. This is an infinite-dimensional vector space in general, but we will pick out some finite-dimensional subspace of it.

We now prove the forward direction of Christol's theorem, the one that goes from the automatic case to the algebraic case.

Proof. Assume, without loss of generality, that $a = (a_n)_{n \geq 0}$ is defined over $\Delta \subseteq \mathrm{GF}(p^n)$. Now a is p-automatic, so it is q-automatic with $q = p^n$. The k-kernel is

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

where $a = a^{(1)}$ and $d \geq 1$. Let's write $a^{(i)} =: (a_n^{(i)})_{n \geq 0}$. Let's define

$$A_i(X) := \sum_{n \geq 0} a_n^{(i)} X^n.$$

Then

$$A_j(X) = \sum_{0 \leq r < q} \sum_{m \geq 0} a_{qm+r}^{(j)} X^{qm+r}$$
$$= \sum_{0 \leq r < q} X^r \sum_{m \geq 0} a_{qm+r}^{(j)} X^{qm}.$$

Now $\sum_{m \geq 0} a_m^{(j')} X^{qm} = A_{j'}(X^q)$ for any $1 \leq j' \leq d$. Thus,

$$A_j(x) \in \langle A_1(X^q), A_2(X^q), \ldots, A_d(X^q) \rangle \quad (*)$$

for $1 \leq j \leq d$, where the angled brackets indicate the vector space span. Substituting X^q for X in (*) gives

$$A_j(X^q) \in \langle A_1(X^{q^2}), A_2(X^{q^2}), \ldots, A_d(X^{q^2}) \rangle.$$

Doing this d times gives

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

Thus, there exist rational functions $B_0, B_1, \ldots, B_d$ such that

$$\sum_{0 \le i \le d} B_i A_j(X^{q^i}) = 0.$$

(In fact, by clearing denominators we can assume the B_i's are polynomials.) This implies that $A_j(X)$ is the root of an algebraic equation of degree at most q^d. This completes the proof of the first direction of Christol's theorem. □

Soon we will prove the "algebraic $\implies$ automatic" direction of the theorem. First, we need a couple lemmas.

Lemma 8.4. *If $A(X) = \sum_{i \ge 0} a_i X^i$ is algebraic over GF(q)(X) with $q = p^n$, then there exist $t+1$ polynomials $B_0(X), \ldots, B_t(X)$, not all zero, such that*

$$B_0 A + B_1 A^q + B_2 A^{q^2} + \cdots + B_t A^{q^t} = 0.$$

Furthermore, such a relation exists with $B_0 \neq 0$.

Proof. Consider $A^q, A^{q^2}, A^{q^3}, \ldots$. These cannot all be linearly independent, so there is a relation between them. Assume $B_0 A + B_1 A^q + \cdots + B_t A^{q^t} = 0$. Let $j := \min\{i \mid B_i \neq 0\}$. Our goal is to show that $j = 0$.

We have

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

Therefore, there exists some r such that $\Lambda_r(B_j) \neq 0$. Now,

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

because $B_0 = B_1 = \cdots = B_{j-1} = 0$. We have

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

so if $j \neq 0$, then

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

This contradicts the minimality of t, so $j = 0$. It follows that $B_0 \neq 0$. □

Lemma 8.5. *Let $a = (a_n)_{n \ge 0}$ be a sequence over GF(q). Then a is q-automatic if and only if there exists a finite collection $\mathcal{F}$ of power series such that (i) $\sum_{n \ge 0} a_n X^n \in \mathcal{F}$ and (ii) $\Lambda_r(g) \in \mathcal{F}$ for all $g \in \mathcal{F}$ and all $0 \le r < q$.*

This last lemma just follows because the given statement is equivalent to finiteness of the k-kernel. Now we are ready to prove the "algebraic $\implies$ automatic" direction of Christol's theorem.

Proof. Suppose $A(X) = \sum_{i \geq 0} a_i X^i$ is algebraic. Then there exist $B_0, B_1, \ldots, B_t$ such that

$$\sum_{0 \leq i \leq t} B_i A^{q^i} = 0$$

and $B_0 \neq 0$. Let

$$G(X) := A(X)/B_0(X),$$

so $G \in \mathrm{GF}(q)((X))$ and $A = GB_0$. Then

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

which implies that

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

Then

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

where

$$C_i := -B_i B_0^{q^i - 2},$$

so $C_i \in \mathrm{GF}(q)[X]$. Set

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

Let

$$S := \{H \in \mathrm{GF}(q)((X)) \mid H = \sum_{0 \leq i \leq t} D_i G^{q^i}, D_i \in \mathrm{GF}(q)[X], \deg D_i \leq N\}.$$

Note that $A \in S$. We claim that $\Lambda_r(S) \subseteq S$ and prove this now.

Proof. Let $H \in S$. Then

$$\begin{aligned} \Lambda_r(H) &= \Lambda_r(D_0 G + \sum_{1 \leq i \leq t} D_i G^{q^i}) \\ &= \sum_{1 \leq i \leq t} \Lambda_r(D_0 C_i + D_i) G^{q^{i-1}}. \end{aligned}$$

Now $\deg \Lambda_r(D_0 C_i + D_i) \leq 2N/q \leq N$, so $\Lambda_r(H) \in S$. □

This completes the proof of Christol's theorem. □

The *Hadamard product* of formal series $G(X)$ and $H(X)$ is written $G \odot H$ and defined by

$$(G \odot H)(X) := \sum_n g_n h_n X^n$$

if $G(X) = \sum_n g_n X^n$ and $H(X) = \sum_n h_n X^n$.

Theorem 8.6. *If G, H are two algebraic formal series over $GF(q)(X)$ with $q = p^n$, p prime, $n \geq 1$, then so is $G \odot H$.*

Proof. If G, H are algebraic, then $(g_n)_{n\geq 0}$, $(h_n)_{n\geq 0}$ are q-automatic. So $(g_n h_n)_{n\geq 0}$ is q-automatic, which implies that $\sum_n g_n h_n X^n$ is algebraic over $\mathrm{GF}(q)(X)$. By Christol's theorem, we are done. □

Carlitz and Wade worked, back in the 1930s and '40s, on proving power series over finite fields are not algebraic. They developed many complicated arithmetic techniques for this, but with automata theory, this becomes much easier to do.

9 Feb. 4, 2020.

Christol's theorem states that for $q = p^n$, p prime, $n \geq 1$, a sequence is automatic if and only if its corresponding power series is algebraic. Now we discuss the Riemann zeta function. The values of $\zeta(2k)$ for $k \geq 1$ have nice formulas. The irrationality of $\zeta(3)$, proved in 1979 by Apéry, came as such a shock that many mathematicians did not originally believe the proof. Alf van der Poorten wrote an article for the Mathematical Intelligencer defending it.

Professor Shallit recommends the book *Function Field Arithmetic* by Thakur for information about power series. Setting $q := p^n$ we can define a zeta function for $\mathrm{GF}(q)[X]$ by

$$\zeta_q(n) := \prod_{P \text{ monic, irreducible in } \mathrm{GF}(q)[X]} (1 - P^{-n})^{-1} = \sum_{P \text{ monic in } \mathrm{GF}(q)[X]} P^{-n} \in \mathrm{GF}(q)[X^{-1}].$$

Carlitz proved that if $q - 1 \mid n$, then

$$\zeta_q(n) = \Pi_q^n \cdot R$$

where R is a rational function and

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

Denoting by X^* the invertible elements of X, we have

$$|\mathrm{GF}(q)^*| = q - 1.$$

Wade proved (c. 1941) that Π_q is transcendental over $\mathrm{GF}(q)(X)$.

Example 9.1. We calculate

$$\zeta_2(1) = 1 + \frac{1}{X} + \frac{1}{X+1} + \frac{1}{X^2} + \frac{1}{X^2+1} + \frac{1}{X^2+X} + \frac{1}{X^2+X+1} + \cdots$$

$$= 1 + X^{-2} + X^{-3} + X^{-4} + X^{-5} + X^{-9} + X^{-10} + \cdots .$$

Definition 9.2. Let f be a formal power series. The *logarithmic derivative* of f is defined to be f'/f.

The formal derivative of $f(x) = \sum_{i\geq 0} a_i X^{-i}$ is $f'(x) = \sum_{i\geq 1}(-i)a_i X^{-(i+1)}$. If f is algebraic, then so is f' because by Christol's theorem, (a_i) is automatic, thus $(-i)a_i$ is also automatic. It follows that f'/f is algebraic because algebraic elements form a field.

The logarithmic derivative is useful because it turns products into sums. For example,

$$\frac{(xy)'}{xy} = \frac{x'}{x} + \frac{y'}{y}.$$

We now give Wade's proof.

Proof. Recall that

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

After a short calculation, we obtain

$$\frac{\Pi_q'}{\Pi_q} = \sum_{k\geq 1} \frac{1}{X^{q^{k+1}} - x}$$

$$= \left(\sum_{k\geq 1} \frac{1}{X^{q^k} - X}\right) - \frac{1}{X^{q^k} - X},$$

which implies that

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

is algebraic. (In the literature, sometimes the abbreviation $[k] := X^{q^k} - X$ is used. Professor Shallit remarks that the typesetters of the 1940s were probably grateful for this.)

Let

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

$$= \sum_{k\geq 1} \frac{1}{X^{q^k}(1 - (1/x)^{q^k - 1})}$$

$$= \sum_{k\geq 1} \frac{1}{X^{q^k}} \sum_{n\geq 0} (1/X)^{n(q^k-1)}$$

$$= \frac{1}{X} \sum_{k \geq 1} \frac{1}{X^{q^k-1}} \sum_{n \geq 0} \left(\frac{1}{X}\right)^{n(q^k-1)}$$

$$= \frac{1}{X} \sum_{k \geq 1, n \geq 0} \left(\frac{1}{X}\right)^{(n+1)(q^k-1)}$$

$$= \frac{1}{X} \sum_{k \geq 1, m \geq 1} \left(\frac{1}{X}\right)^{m(q^k-1)}$$

$$= \frac{1}{X} \sum_{r \geq 1} X^{-r} \left(\sum_{k,m \geq 1, m(q^k-1)=r} 1 \right)$$

$$= \frac{1}{X} \sum_{r \geq 1} X^{-r} \left(\sum_{k \geq 1, q^k-1 | r} 1 \right)$$

$$= \frac{1}{X} \sum_{r \geq 1} c(r) X^{-r}$$

where

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

.

Note that $(c(r) \bmod p)_{r \geq 1}$ is a q-automatic sequence. We use the following trick: if $(c(r))_{r \geq 0}$ is k-automatic, then $c(k^r - 1))_{r \geq 0}$ is ultimately periodic. To prove this, consider outputs of the DFAO generating $(c(r))$ when it is given inputs of the form $(k-1, k-1, k-1, \dots)$, r times.

Therefore, $(c(q^n - 1) \bmod p)_{n \geq 1}$ is ultimately periodic. We have

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

Note that $q^k - 1 \mid q^n - 1$ if and only if $k \mid n$. Thus,

$$c(q^n - 1) = \sum_{k \geq 1, k | n} 1 = d(n),$$

the sum of the divisors of n. Thus, $(d(n) \bmod p)_{n \geq 1}$ is ultimately periodic, i.e., there exist t and n_0 such that

$$d(n + it) \equiv d(n) \pmod{p}$$

for all $i \geq 1$ and $n \geq n_0$. Choose $i := ni'$, $i' \geq 1$. We get

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

By Dirichlet's theorem, there exist infinitely many primes p' of the form $1 + i't$, and so there

exists one that is greater than or equal to n_0. Choose $n := p'$. Then

$$d((p')^2) \equiv d(p') \pmod{p},$$

so $3 \equiv 2 \pmod{p}$, which is a contradiction. □

We now discuss how to prove series in $\mathbb{Q}[[X^{-1}]]$ or $\mathbb{Q}[[X]]$ are transcendental. The rough idea is to (i) assume the series is algebraic: (ii) reduce its coefficients mod p, obtaining a series in $\mathrm{GF}(p)[X]$, for example; (iii) prove the new series is not algebraic by some means; and (iv) conclude that the original series is transcendental. What could go wrong is if all the coefficients of the equation $\beta_0 + \beta_1 F + \cdots + \beta_t F^t = 0$ witnessing the algebraicity of the original polynomial vanish mod p. But if this happens, then all those coefficients are divisible by p, so you could just divide out by p to begin with.

Consider

$$\Theta_3(X) := \sum_{-\infty<n<\infty} X^{n^2}.$$

We wish to prove that $\Theta_3(X)$ is transcendental over $\mathbb{Q}(X)$. Assume it is algebraic. Then since

$$\Theta_3(X) = 1 + 2\sum_{n\geq 1} X^{n^2}$$

we have that $\sum_{n\geq 1} X^{n^2}$ is algebraic. So $\sum_{n\geq 1} X^{n^2}$ is algebraic over $\mathrm{GF}(2)(X)$. By Christol's theorem, the characteristic sequence $(a_n)_{n\geq 0}$ of the squares is 2-automatic. We now enjoy a historical diversion.

Büchi was interested in logic circa 1960. Alan Cobham (not the pilot), circa 1968, was working at IBM and wrote a paper called "uniform tag sequences" (1972). Today these are called k-automatic sequences. In his paper, Cobham proved several cool theorems about k-automatic sequences, including his "little theorem". Some of the other theorems are about gaps that can occur between the elements of a k-automatic sequence. One of these theorems is the following.

Theorem 9.3. *Let $x = (x_n)_{n\geq 0}$ be a k-automatic sequence taking values in Δ. Let $d \in \Delta$. Let α_j be the position of the j^{th} occurrence of d in x. Then either*

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

or else

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

(The notation $|x[0..n-1]|_d$ means the number of occurrences of d in x from position 0 to position $n-1$.)

Applying this theorem, the characteristic sequence of the squares is not 2-automatic, so $\Theta_3(X)$ is transcendental. We now state a corollary to Cobham's theorem.

Corollary 9.4. *Let p be a polynomial such that $p(\mathbb{N}) \subseteq \mathbb{N}$. Then the characteristic sequence of $\{p(0), p(1), \dots\}$ is k-automatic if and only if* $\deg(p) < 2$.

10 Feb. 6, 2020.

I was headed to Michigan this class. The notes are available on Professor Shallit's website.

11 Feb. 11, 2020.

I will give the first presentation on March 24. Presentations should be about 20 to 25 minutes each, which leaves some time for questions.

Professor Shallit points out that the number of quantifiers tends to be quite low (at most around five) for the properties people are interested in. So in practice, Walnut tends not to time out when checking statements about automatic sequences. The other day, Jason Bell asked him a question about automatic sequences, and it is now one of the assignment questions.

I will not take notes on the material from the slideshow being presented today. In the slide with "eval tmup", "eval" means "evaluate" and "tmup" is the file name you save it to. (It stands for "Thue–Morse ultimate periodicity".) Be careful with the use of "&" and "=>". The code "Ep (p >= 1) &" means "There exists p, and p >= 1". The final "&" binds "Ep" to "(p >= 1)". Note that Walnut knows that "T" refers to the Thue–Morse sequence. On the slide that says "orders of squares", the word "squares" means substrings of the form xx where x is some non-empty substring; it does not refer to squares of integers.

A *fractional power* n/p is a word of length n and period p. For example, "alfalfa" is a (7/3)-power. We say a word w *avoids* α-powers if it contains, as factors, no word of fractional power β with $\beta \geq \alpha$.

The notation "?msd=13" means "evaluate the string in base 13, starting with the most significant digit".

If you don't quantify over a variable in Walnut, it gives you the automaton that accepts the values of that variable that make the formula true.

The property of being balanced (on one of the slides) is one that we do not know how to state in first-order logic. How can you count the number of letters in a word in first-order logic? But then we have a nice alternative characterization that we can conver to a statement of first-order logic instead.

12 Feb. 13, 2020.

We continue going through the slideshow from last class. I will only write down things that aren't in the slideshow.

Definition 12.1. Let $(s_n)_{n\geq 0}$ be a sequence. Its *critical exponent* is

$$\sup\{e \mid \exists \text{ a finite block } x \text{ in } (s_n)_{n\geq 0} \text{ such that } \exp(x) = e\}.$$

Here

$$\exp(x) := \frac{|x|}{\text{per}(x)}$$

where $\text{per}(x)$ is the period of x.

Professor Shallit asks the following open problem. Consider the *Cantor numbers*, those base 3 numbers that can be written using only the digits 0 and 2. Call this set

$$C := \{0, 2, 6, 8, \dots\}.$$

Which non-negative integers can be written as the quotient of two Cantor numbers? In other words, what are the elements of

$$\{\frac{p}{q} \mid p, q \in C, q \neq 0\} \cap \mathbb{Z}_{\geq 0}.$$

The Rudin–Shapiro sequence is an example of an automatic sequence that is not the fixed point of any uniform morphism.

Definition 12.2. Let $s := (s_n)_{n\geq 0}$ be a sequence. The *subword complexity* (sometimes called the *factor complexity* or just the *complexity*) of s, denoted by $\rho_s(n)$, is defined to be the number of distinct length-n blocks appearing in s.

If s is written in base k, we have the bound $1 \leq \rho_s(n) \leq k^n$. The subword complexity can be calculated explicitly when s is k-automatic. There is a theorem saying that for automatic sequences, subword complexity is small. More precisely, if s is k-automatic, then $\rho_s(n) = O(n)$. Given an automaton generating s, we can produce a formula for $\rho_s(n)$. These formulas can be quite complicated. Even in the case of the Thue–Morse sequence, the formula for $\rho_s(n)$ is somewhat complicated and is given by a piecewise function depending on when n is bounded by various powers of 2. But the problem with giving this type of formula is what exactly do you allow as your operations in the function?

Instead, we will give a different type of formula. We will have $t \times t$ matrices $M_0, M_1, \dots, M_{k-1}$ and vectors u, v, of respective dimensions $1 \times t$ and $t \times 1$. Then we will write

$$\rho_s(n) = vM_{a_1} \cdots M_{a_r} v$$

where

$$(n)_k =: a_1 a_2 \dots a_r.$$

This is not entirely satisfactory because we will be multiplying matrices with integer entries, which already causes undecidability issues. Mike Patterson proved in the 1970s that the problem of whether there is a product of finitely many given 3×3 matrices that is equal to the 3×3 zero matrix is undecidable. There is a notion of joint spectral radius of matrices, though, that gives an upper bound on how fast such a product can grow and gives a polynomial time algorithm for computing the subword complexity here. This is the way around the undecidability issues. The representation in base k has only $\log n$ bits, so you are multiplying some matrices $\log n$ times, which is quite efficient for large n.

In our next assignment, we will be asked how many different equivalence classes there are for words in which all conjugates appear in Thue–Morse. You can get an automaton and get a formula for how many there are. This is related to the previous problem.

The trick on Slide 9/56 of Slideshow 4 involving counting the first occurrence of each palindrome rather than just counting each palindrome is apparently a useful hint about Assignment 2.

To prove a quantity is finite, often you can find a k-regular sequence, you produce the matrix representation, you use the "semigroup trick" from Slide 14 or so of Slideshow 4, and thus you get a bound.

Recall that a sequence s is k-automatic if and only if the k-kernel $K_k(s)$ is finite. It turns out that s is k-regular if and only if $K_k(s)$ is contained in a finitely-generated $\mathbb{Z}$-module. For example, recall that $s_2(n)$ is the sum of the bits in n's base 2 representation. Then $s_2(2n) = s_2(n)$ and $s_2(2n+1) = s_2(n) + 1$. Thus, the k-kernel of $s_2(n)$ will be contained in $\langle s_2(n), 1\rangle$ where 1 is the sequence of all 1's.

13 Feb. 25, 2020.

A sequence f is *k-regular* if there exist finitely many sequences $(f_1(n)), \ldots, (f_r(n))$ such that every sequence in the k-kernel of $(f(n))$ is a $\mathbb{Q}$-linear combination of the $(f_i(n))$'s.

Equivalently, in the above definition one can assume that the finitely many sequences can be taken to be elements of the k-kernel. Equivalently, there exists a $1 \times r$ vector v, an $r \times 1$ vector w, and $r \times r$ matrices $\mu(a)$, $0 \le a < k$, such that

$$f(n) = v \cdot \mu((n)_k) \cdot w.$$

Prof. Shallit: "With greater generalization comes greater power and undecidability." Unlike the case of automata, there is typically no general procedure for solving decision problems about k-regular sequences.

Here is an example. Let $s_2(n)$ be the sum of the bits in $(n)_2$. Take $k = 2$. Then the k-kernel

consists of elements of the form

$$s_2(2^e n + i) = s_2(n) + s_2(i), 0 \leq i < 2^e.$$

Thus, the k-kernel is contained in

$$\langle (s_2(n))_{n \geq 0}, 1 \rangle.$$

Here we have used the first definition of 2-regularity. Alternatively, using the second definition, we observe that

$$s_2(2n) = s_2(n), s_2(4n+1) = s_2(2n+1), s_2(4n+3) = -s_2(n) + 2s_2(2n+1).$$

Thus, the 2-kernel of s_2 is contained in

$$\langle s_2(n), s_2(2n+1) \rangle.$$

Alternatively, using the third definition of 2-regularity, we observe that

$$[s_2(n), 1]\mu(0) = [s_2(2n), 1]$$

and

$$[s_2(n), 1]\mu(1) = [s_2(2n+1), 1].$$

Here we take

$$v := [0, 1], w := [1, 0]^T, \mu(0) = I_2, \mu(1) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

Also, we have

$$[s_2(n), s_2(2n+1)]\mu_0 = [s_2(2n), s_2(4n+1)],$$
$$[s_2(n), s_2(2n+1)]\mu_1 = [s_2(2n+1), s_2(4n+3)].$$

Here we take

$$v := [0, 1], w := [1, 0]^T, \mu_0 := I_2, \mu_1 := \begin{bmatrix} 0 & -1 \\ 1 & 2 \end{bmatrix}.$$

We now discuss closure properties of k-regular sequences. Let $s := (s(n))$, $t := (t(n))$ be k-regular. Then (i) $s + t = (s(n) + t(n))$, (ii) $st = (s(n)t(n))$, and (iii) $ct = (ct(n))$ are k-regular for any constant c.

Proof. By k-regularity, there exist $(s_1(n)), \ldots, (s_r(n))$ such that the k-kernel of s is contained in $\langle s_1, s_2, \ldots, s_r \rangle$ and there exist $(t_1(n)), \ldots, (t_{r'}(n))$ such that the k-kernel of t is contained in $\langle t_1, t_2, \ldots, t_{r'} \rangle$. Then the k-kernel of $s + t$ is contained in $\langle s_1, s_2, \ldots, s_r, t_1, t_2, \ldots, t_{r'} \rangle$, the k-kernel of st is contained in $\{s_i t_j \mid 1 \leq i \leq r, 1 \leq j \leq r'\}$, and the k-kernel of ct is contained in $\langle t_1, t_2, \ldots, t_{r'} \rangle$. (For a ring, you need to change this to $\langle ct_1, ct_2, \ldots, ct_{r'} \rangle$, but since we are working over $\mathbb{Q}$, we do not need the constants.) □

Suppose s and t are k-regular and $t(n) \neq 0$ for all n. Define $s/t := (s(n)/t(n))$. Is s/t k-regular? It turns out that, in general, it isn't. For example, take $s(n) := 1$, $t(0) := 1$,

$t(2n) := n+1$, $t(2n+1) := t(n)+1$. Then

$$t(4n) = t(n) + 2t(2n) - t(2n+1),$$

$$t(4n+1) = -t(n) + t(2n) + t(2n+1),$$

$$t(4n+2) = 2t(2n),$$

$$t(4n+3) = -t(n) + 2t(2n+1).$$

We claim that $(1/t(n))$ is not 2-regular. We need to produce arbitrarily many linearly independent sequences in its 2-kernel. Define

$$t_j(n) := t(2^j n + 2^{j-1} - 1), j \geq 1.$$

One can check that $t_j(n) = n + j$. Then $(1/t(n))$ has the sequence $(1/(n+j))$ in its 2-kernel for $j \geq 1$. It remains to prove that these are all linearly independent. We put the first r values of $1/(n+1)$ into the first row of a matrix, the first r values of $1/(n+2)$ into the second row, and so on until the first r values of $1/(n+r)$ are placed in the r^{th} row. The resulting matrix

$$H_r = \begin{bmatrix} 1/1 & 1/2 & 1/3 & 1/4 & \dots \\ 1/2 & 1/3 & 1/4 & 1/5 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

is known as the $r \times r$ *Hlbert matrix*. Its determinant is known to be

$$\det(H_r) = \left(\pm \prod_{1 \leq k < r} (2k+1) \binom{2k}{k}^2 \right)^{-1}.$$

This is really small in absolute value but non-zero. This proves the $1/(n+j)$'s are linearly independent.

We now discuss the composer Per Nørgård's sequence

$$s(0) := 0, s(2n) := -s(n), s(2n+1) := s(n) + 1.$$

The sequence begins $(0, 1, -1, 2, 1, 0, -2, 3, -1, \dots)$. Nørgård used this to compose music by assigning the note G to 0, $G\#$ to 1, $F\#$ to -1, etc.

Let $e_0(n)$ be the number of 0's in $(n)_2$, $e_1(n) := s_2(n)$ (the number of 1's in base 2). Then

$$f(n) := e_0(n) - e_1(n)$$

is 2-regular. However, $|f(n)|$ is not even though $f(n)^2$ is.

Proof. Let $g(n) := |f(n)|$. Then

$$g(2^j n) = |e_0(n) - e_1(n) + j|, n \geq 1, j \geq 0.$$

Suppose $\{(g(2^j n))_{n\geq 1} \mid j \geq 0\}$ were linearly dependent. Then $|m+j|$, $j \geq 0$, would be linearly dependent for all $m \in \mathbb{Z}$. Thus, for some a, we would have

$$|m+a| = \sum_{a+1\leq j\leq b} c_j |m+j|$$

for some constants c_j. Look at the right-hand side for $m \geq -(a+1)$. It is monotonic, but evaluating the left-hand side at $m := -(a+1)$ gives 1, at $m := -a$ gives 0, and at $m := 1-a$ gives 1, so it is not monotonic. This is a contradiction, so $|f(n)|$ is not 2-regular. □

We showed that any linearly indexed subsequence of an automatic sequence is automatic. We would now like to prove the same result for k-regular sequences. To do so, we discuss transducers of k-regular sequences.

Let T be a deterministic finite-state transducer with transitions on single letters in $\Sigma_k = \{0, 1, \ldots, k\}$ and outputs in Σ_k^*. If $f(n)$ is k-regular, we claim that so is $f((T((n)_k)))_{n\geq 0}$. Since we can make a transducer that on input n, outputs $an+b$, this claim, once proved, will complete the argument.

Proof. Combine T with the linear representation for f. Suppose the linear representation of f has rank s with parameters v,w, and $\mu(a)$ for $0 \leq a < k$. The transducer has r states, say. We can think of the transducer as itself having a linear representation, with a vector $[1, 0, \ldots, 0]$ corresponding to the initial state, matrices corresponding to the transitions, and an output vector $[1, 1, \ldots, 1]^T$. The transition matrix has a 1 in position (i, j) if and only if $\delta(q_i, -) = q_j$, and a 0 otherwise. We will create a linear representation for the sequence

$$g(n) := f(T((n)_k))$$

using $(rs) \times (rs)$ matrices (really the tensor product). We think of these matrices as $r \times r$ matrices where the entries are $\mu(a)$'s. We build such a matrix by placing $\mu(t)$ in position (i, j) if and only if $\delta(q_i, t) = q_j$. □

We now prove a theorem.

Theorem 13.1. *The following problem is not recursively solvable (i.e., is undecidable): Given a k-regular sequence over $\mathbb{Z}$, say $f(n)$, does there exist n_0 such that $f(n_0) = 0$?*

Proof. We reduce this to Hilbert's 10^{th} problem: Given a multivariate polynomial $p(x_1, x_2, \ldots, x_r)$ with integer coefficients, decide whether there exist $a_1, a_2, \ldots, a_r \in \mathbb{N}$ with $p(a_1, a_2, \ldots, a_r) = 0$. This problem is known to be undecidable.

Let p be an input to Hilbert's 10^{th} problem. We need to make a k-regular sequence out of it. The easiest thing to do is to take n, express it in base k, and count the number of occurrences of each digit. Let $k := r + 1$ where r is the number of variables in our Hilbert's problem instance. Create a sequence

$$f(n) := p(|(n)_k|_1, |(n)_k|_2, \ldots, |(n)_k|_r).$$

(Here $|(n)_k|_1$ denotes the number of occurrences of the digit 1 in the base-k expansion of n.) Since k-regular sequences are closed under addition and multiplication, any polynomial in them is going to be k-regular. (Also, everything done so far is effective: given a polynomial, we could explicitly find the linear representation of f.) Then $f(n_0) = 0$ if and only if $p(a_1, a_2, \ldots, a_r) = 0$ where $a_i := |(n_0)_k|_i$. This proves the claim. □

In certain special cases, questions about k-regularity are decidable, but not in general, as the above theorem shows.

14 Feb. 27, 2020.