Decidability and Enumeration in Automatic Sequences

Jeffrey Shallit
School of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada
shallit@cs.uwaterloo.ca
http://www.cs.uwaterloo.ca/~shallit

Joint work with Jean-Paul Allouche, Émilie Charlier, Narad Rampersad, Dane Henshall, Luke Schaeffer, Eric Rowland, Daniel Goč, and Hamoon Mousavi.

In Memory of My Grandparents

Зиппора Левинтан
(1877–1964)

Моише Шалит
(1871–1949)

What is an automatic sequence?

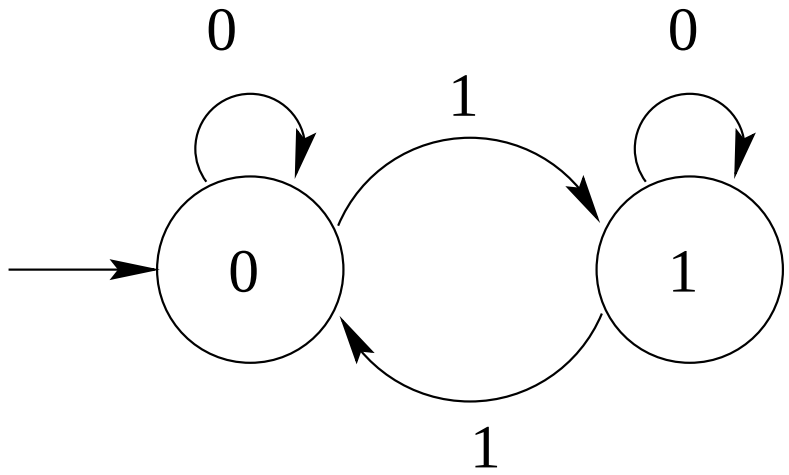
- An infinite sequence

$$\mathbf{a} = a_0 a_1 a_2 \cdots$$

over a finite alphabet of letters, generated by a finite-state machine (automaton)

- The automaton, given n as input, computes a_n as follows:
  - n is represented in some fixed integer base $k \geq 2$
  - The automaton moves from state to state according to this input
  - Each state has an output letter associated with it
  - The output on input n is the output associated with the last state reached

The canonical example: the Thue-Morse automaton



This automaton generates the Thue-Morse sequence

$$\mathbf{t} = (t_n)_{n \geq 0} = 0110100110010110 \cdots .$$

Why automatic sequences?

- A nontrivial class of self-similar sequences
- Many "naturally-occurring" sequences are automatic
- Halfway between periodic and chaotic
- Provide canonical examples for various kinds of avoidance problems

Historically interesting properties of **t**

1. **t** is not ultimately periodic.
2. **t** contains no factor that is an *overlap*, that is, a word of the form *axaxa*, where *a* is a single letter and *x* is an arbitrary finite word. (Example in Russian: мамам.)
3. **t** contains infinitely many distinct square factors *xx*, but for each such factor we have $|x| = 2^n$ or $3 \cdot 2^n$, for $n \geq 0$. Examples of squares in Russian include дядя ("uncle") and кускус ("couscous").
4. **t** has infinitely many distinct palindromic factors (A *palindrome* is a word equal to its reverse, like доход.)
5. The number $p(n)$ of distinct palindromic factors of length n in **t** is given by

$$p(n) = \begin{cases} 0, & \text{if } n \text{ odd and } n \geq 5; \\ 1, & \text{if } n = 0; \\ 2, & \text{if } 1 \leq n \leq 4, \text{ or } n \text{ even and } 3 \cdot 4^k + 2 \leq n \leq 4^{k+1}; \\ 4, & \text{if } n \text{ even and } 4^k + 2 \leq n \leq 3 \cdot 4^k. \end{cases}$$

Historically interesting properties of $\mathbf{t}$

6. $\mathbf{t}$ is *mirror-invariant*: if x is a finite factor of $\mathbf{t}$, then so is its reverse x^R.
7. $\mathbf{t}$ is *recurrent*, that is, every factor that occurs, occurs infinitely often.
8. $\mathbf{t}$ is *uniformly recurrent*, that is, for all factors x occurring in $\mathbf{t}$, there is a constant $c(x)$ such that two consecutive occurrences of x are separated by at most $c(x)$ symbols.
9. $\mathbf{t}$ is *linearly recurrent*, that is, it is uniformly recurrent and furthermore there is a constant C such that $c(x) \leq C|x|$ for all factors x. In fact, the optimal bound is given by $c(1) = 3$, $c(2) = 8$, and $c(n) = 9 \cdot 2^e$ for $n \geq 3$, where $e = \lfloor \log_2(n-2) \rfloor$.

Historically interesting properties of **t**

10. The lexicographically least sequence in the shift orbit closure of **t** is
$$\overline{t_1}\ \overline{t_2}\ \overline{t_3}\ \cdots = 0010110\cdots,$$
which is also 2-automatic.

11. The *subword complexity* $\rho(n)$ of **t**, which is the function counting the number of distinct factors of **t**, is given by
$$\rho(n) = \begin{cases} 2^n, & \text{if } 0 \le n \le 2; \\ 2n + 2^{t+2} - 2, & \text{if } 3\cdot 2^t \le n \le 2^{t+2}+1; \\ 4n - 2^t - 4, & \text{if } 2^t + 1 \le n \le 3\cdot 2^{t-1}; \end{cases}.$$

12. **t** has an unbordered factor of length n if $n \not\equiv 1 \pmod 6$ (Here by an *unbordered* word y we mean one with no expression in the form $y = uvu$ for words u, v with u nonempty.)

The claim

Claim. All of these properties can be verified (and in some case, even obtained) purely mechanically, by a machine computation.

To see how, we need to digress into...

Logic

- Let $\mathrm{Th}(\mathbb{N}, +, 0, 1, <)$ denote the set of all true first-order sentences in the logical theory of the natural numbers with addition.
- Example: in this theory we can express the so-called "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:

$$(\forall n > 43\ \exists x, y, z \geq 0 \text{ such that } n = 6x + 9y + 20z) \wedge \neg(\exists x, y, z \geq 0 \text{ such that } 43 = 6x + 9y + 20z). \quad (1)$$

  Here, of course, "$6x$" is shorthand for the expression "$x + x + x + x + x + x$", and similarly for $9y$ and $20z$.
- Presburger (1929) proved that $\mathrm{Th}(\mathbb{N}, +, 0, 1, <)$ is *decidable*: that is, there exists an algorithm that, given a sentence in the theory, will decide its truth.

Decidability of Presburger arithmetic: proof sketch

- represent integers in a 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.
- For example, the pair $(21, 7)$ can be represented in base 2 by the word

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

Decidability of Presburger arithmetic

- Then 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".

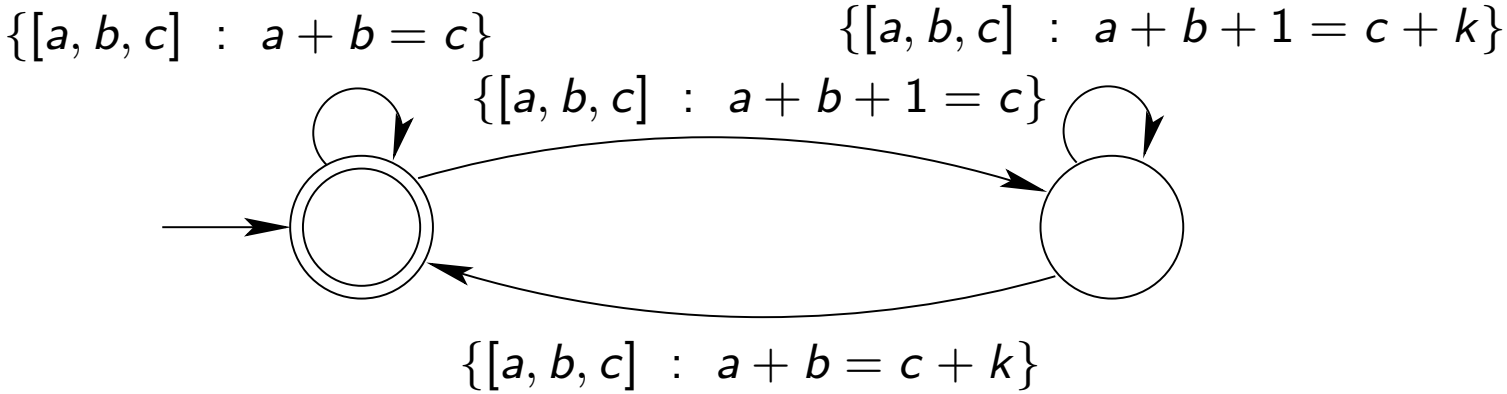


Figure: Checking addition in base k

Decidability of Presburger arithmetic: proof sketch

- Relations like $x = y$ and $x < y$ can be checked similarly.
- 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.
- Finally, the truth of a formula can be checked by using the usual depth-first search techniques to see if any final state is reachable from the start state.

Even more

- If we add the function $V_k : \mathbb{N} \to \mathbb{N}$ to our logical theory, where $V_k(x) = k^n$, and k^n is the largest power of k dividing x, it is still decidable by a similar automaton-based technique (cf. Bruyère, Villemaire, Hodgson, ...).
- By doing so, we gain the capability of deciding many questions about automatic sequences.

Theorem

There is an algorithm that, given a predicate phrased using only the universal and existential quantifiers, indexing into a given automatic sequence **a**, *addition, subtraction, logical operations, and comparisons, will decide the truth of that proposition.*

We call such a predicate an *automatic predicate*.

The bad news

- The worst-case running time of our algorithm 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 quantifiers, p is a polynomial, and N is the number of states needed to describe the underlying automatic sequence.

The good news

- That's just the worst-case upper bound!
- An implementation often succeeds in verifying statements in the theory
- It has been implemented by Dane Henshall and Daniel Goč, independently

Deciding periodicity

- An infinite word **a** is *periodic* if it is of the form $x^\omega = xxx\cdots$ for a finite nonempty word x (e.g., $\texttt{nananana}\cdots$).
- It is *ultimately periodic* if it is of the form yx^ω for a (possibly empty) finite word y (e.g., $\texttt{bananananana}\cdots$).
- Honkala (1986) proved that ultimate periodicity is decidable for automatic sequences.
- Using our approach: it suffices to express ultimately periodicity as an automatic predicate:

$$\exists p \geq 1, N \geq 0\ \forall i \geq N\ \mathbf{a}[i] = \mathbf{a}[i+p].$$

- When we run this on the Thue-Morse sequence, we discover (as expected) that **t** is not ultimately periodic.
- Can be implemented in polynomial time.

Repetitions

- Thue (1912) proved that $\mathbf{t}$ contains no overlaps; that is, $\mathbf{t}$ is overlap-free.
- Using our technique, we can express the property of having an overlap *axaxa* beginning at position N with $|ax| = p$, as follows: $\mathbf{a}[N..N+p] = \mathbf{a}[N+p..N+2p]$.
- So the corresponding automatic predicate for $\mathbf{t}$ is

$$\exists p \geq 1, N \geq 0 \ \mathbf{t}[N..N+p] = \mathbf{t}[N+p..N+2p],$$

or, in other words,

$$\exists p \geq 1, N \geq 0 \ \forall i, 0 \leq i \leq p \ \mathbf{t}[N+i] = \mathbf{t}[N+p+i].$$

- We programmed up our decision procedure and verified that indeed $\mathbf{t}$ is overlap-free.

Critical exponent

- We can define more general repetitions as follows: a word x is an α-power for $\alpha \geq 1$ if we can write $x = y^e y'$ where $e = \lfloor \alpha \rfloor$ and y' is a prefix of y and $|x| = \alpha |y|$.
- For example, `abracadabra` is an $\frac{11}{7}$-power.
- The techniques above suffice to check if a k-automatic sequence has α-powers, using the following predicate:

$$\exists N \geq 0, p, q \geq 1 \; \mathbf{a}[N..N+p-q-1] = \mathbf{a}[N+q..N+p-1] \text{ and } p = \alpha q.$$

- However, this observation alone does not suffice to compute the so-called *critical exponent* of $\mathbf{a}$, which is the supremum over all rational α such that $\mathbf{a}$ has α-power factors.
- It turns out that the critical exponent is also computable for automatic sequences....

Representing rational numbers

- Represent rational number $\alpha = p/q$ by pair of integers (p, q), represented in base k; pad shorter with leading zeroes
- So representations of rationals are over the alphabet $\Sigma_k \times \Sigma_k$
- For example, if $w = [3, 0][5, 0][2, 4][6, 1]$ then $[w]_{10} = (3526, 41)$.
- Define $\mathrm{quo}_k(x) = [\pi_1(x)]_k / [\pi_2(x)]_k$, where π_i is the projection onto the i'th coordinate
- So $\mathrm{quo}_{10}(w) = 3526/41 = 86$.
- Canonical representations lack leading $[0, 0]$'s
- Every rational has infinitely many canonical representations, e.g., as $(1, 2), (2, 4), (3, 6), \ldots$, etc.

Automatic sets over $\mathbb{Q}^{\geq 0}$

- $\mathrm{quo}_k(L) = \bigcup_{x \in L} \{\mathrm{quo}_k(x)\}$
- $A \subseteq \mathbb{Q}^{\geq 0}$ is a k-automatic set of rationals if $A = \mathrm{quo}_k(L)$ for some regular language $L \subseteq (\Sigma_k \times \Sigma_k)^*$.

Examples

Example 1. Let $k = 2$, $B = \{[0,0],[0,1],[1,0],[1,1]\}$, and consider

$$L_1 := B^*\{[0,1],[1,1]\}B^*.$$

Then L_1 consists of all pairs of integers where the second component has at least one nonzero digit — the point being to avoid division by 0. Then $\mathrm{quo}_k(L) = \mathbb{Q}^{\geq 0}$, the set of all non-negative rational numbers.

Example 2. Consider

$$L_2 = \{w \in (\Sigma_k^2)^* \,:\, \pi_1(w) \in 0^* C_k \text{ and } \pi_2(w) \in 0^*1\}.$$

Then $\mathrm{quo}_k(L_2) = \mathbb{N}$.

Examples

Example 3. Let $k = 3$, and consider the language

$$L_3 := [0,1]\{[0,0],[2,0]\}^*.$$

Then $\mathrm{quo}_k(L_3)$ is the 3-*adic Cantor set*, the set of all rational numbers in the "middle-thirds" Cantor set with denominators a power of 3.

Example 4. Let $k = 2$, and consider

$$L_4 := [0,1]\{[0,0],[0,1]\}^*\{[1,0],[1,1]\}.$$

Then the numerator encodes the integer 1, while the denominator encodes all positive integers that start with 1. Hence

$$\mathrm{quo}_k(L_4) = \{\frac{1}{n} \,:\, n \geq 1\}.$$

Examples

Example 5. Let $k = 4$, and consider

$$S := \{0, 1, 3, 4, 5, 11, 12, 13, \ldots\}$$

of all non-negative integers that can be represented using only the digits $0, 1, -1$ in base 4. Consider the language

$$L_5 = \{(p, q)_4 \ : \ p, q \in S\}.$$

It is not hard to see that L_5 is $(\mathbb{Q}, 4)$-automatic.
The main result in Loxton & van der Poorten [1987] can be rephrased as follows: $\mathrm{quo}_4(L_5)$ contains every odd integer.
In fact, an integer t is in $\mathrm{quo}_4(L_5)$ if and only if the exponent of the largest power of 2 dividing t is even.

Examples

Example 6. Consider

$$L_6 = \{w \in (\Sigma_k^2)^* \ : \ \pi_2(w) \in 0^*1^+0^*\}.$$

An easy exercise using the Fermat-Euler theorem shows that that $\mathrm{quo}_k(L_6) = \mathbb{Q}^{\geq 0}$.

Examples

Example 7. For a word x and letter a let $|x|_a$ denote the number of occurrences of a in x. Consider the regular language

$$L_7 = \{w \in (\Sigma_2^2) \ : \ |\pi_1(w)|_1 \text{ is even and } |\pi_2(w)|_1 \text{ is odd}\}.$$

Then it follows from a result of Schmid [1984] that

$$\text{quo}_2(L_7) = \mathbb{Q}^{\geq 0} - \{2^n \ : \ n \in \mathbb{Z}\}.$$

Basic decidability properties

Given a DFA M accepting a language L representing a set of rationals S, can decide

- if $S = \emptyset$
- given $\alpha \in \mathbb{Q}^{\geq 0}$, whether there exists $x \in S$ with $x = \alpha$ (resp., $x < \alpha$, $x \leq \alpha$, $x > \alpha$, $x \geq \alpha$, $x \neq \alpha$, etc.)
- if $|S| = \infty$
- given a finite set $F \subseteq \mathbb{Q}^{\geq 0}$, if $F \subseteq S$ or if $S \subseteq F$
- given $\alpha \in \mathbb{Q}^{\geq 0}$, if α is an accumulation point of S

$\sup A$ is rational or infinite

Given a DFA M accepting $L \subseteq (\Sigma_k \times \Sigma_k)^*$ representing a set of rationals $A \subseteq \mathbb{Q}^{\geq 0}$, what can we say about $\sup A$?

Theorem. $\sup A$ is rational or infinite.

Proof ideas: $\mathrm{quo}_k(uv^i w)$ forms a monotonic sequence. Defining

$$\gamma(u, v) := \frac{[\pi_1(uv)]_k - [\pi_1(u)]_k}{[\pi_2(uv)]_k - [\pi_2(u)]_k}$$

one of the following three cases must hold:

(i) $\mathrm{quo}_k(uw) < \mathrm{quo}_k(uvw) < \mathrm{quo}_k(uv^2w) < \cdots < U$;
(ii) $\mathrm{quo}_k(uw) = \mathrm{quo}_k(uvw) = \mathrm{quo}_k(uv^2w) = \cdots = U$;
(iii) $\mathrm{quo}_k(uw) > \mathrm{quo}_k(uvw) > \mathrm{quo}_k(uv^2w) > \cdots > U$.

Furthermore, $\lim_{i\to\infty} \mathrm{quo}_k(uv^i w) = U$.

sup A is rational or infinite

It follows that if $\sup A$ is finite, and the DFA M has n states, then $\sup A = \max T$, where

$$T = T_1 \cup T_2$$

and

$$\begin{aligned} T_1 &= \{\mathrm{quo}_k(x) : |x| < n \text{ and } x \in L\}; \\ T_2 &= \{\gamma(u,v) : |uv| \leq n,\ |v| \geq 1,\ \delta(q_0,u) = \delta(q_0,uv), \\ &\quad \text{and there exists } w \text{ such that } uvw \in L\}. \end{aligned}$$

$\sup A$ is computable

We know that $\sup A$ lies in the finite computable set T.

For each of $t \in T$, we can check to see if $t \geq \sup A$ by checking if $A \cap (t, \infty)$ is empty.

Then $\sup A$ is the least such t.

Computing the critical exponent

- Previously known to be computable for fixed points of uniform morphisms (Krieger)

Theorem. If $\mathbf{w}$ is a k-automatic sequence, then its critical exponent is rational or infinite. Furthermore, it is computable from the DFAO M generating w.

Proof sketch. Given M, we can transform it into another automaton M' accepting

$\{(m,n) : \text{there exists } i \geq 0 \text{ such that } \mathbf{w}[i..i+m-1] \text{ has period } n\}$.

We then apply our algorithm for computing $\sup(\mathrm{quo}_k(L))$ to $L(M')$.

The Leech word

Leech [1957] showed that the fixed point **l** of the morphism

$$\begin{aligned} 0 &\to 0121021201210 \\ 1 &\to 1202102012021 \\ 2 &\to 2010210120102 \end{aligned}$$

is squarefree.

We used our method to compute the critical exponent of this word. It is $15/8$.

Furthermore, if x is a $15/8$-power occurring in **l**, then $|x| = 15 \cdot 13^i$ for some $i \geq 0$.

Applications 2: Diophantine exponent

The *Diophantine exponent* of an infinite word w is defined to be the supremum of the real numbers β for which there exist arbitrarily long prefixes of **w** that can be expressed as uv^e for finite words u, v and rationals e such that $|uv^e|/|uv| \geq \beta$.

(concept due to Adamczewski, Bugeaud)

Theorem. The Diophantine exponent of a k-automatic sequence is either rational or infinite. Furthermore, it is computable.

Mirror invariance

We can express the property that $\mathbf{a}$ is mirror-invariant as follows:

$$\forall N \geq 0, \ell \geq 1 \ \exists N' \geq 0 \ \mathbf{a}[N..N+\ell-1] = \mathbf{a}[N'..N'+\ell-1]^R,$$

which is the same as

$$\forall N \geq 0, \ell \geq 1 \ \exists N' \geq 0 \ \forall i, \ 0 \leq i < \ell \ \mathbf{a}[N+i] = \mathbf{a}[N'+\ell-i-1],$$

which can be easily checked by our method.

Recurrence

- We can express the property that **a** is recurrent by saying that for each factor, and each integer M there exists a copy of that factor occurring at a position after M in **a**.
- This corresponds to the following predicate:

$$\forall N, M \geq 0, \ell \geq 1 \; \exists M' \geq M \; \mathbf{a}[N..N+\ell-1] = \mathbf{a}[M'..M'+\ell-1].$$

- An easy argument shows that an infinite word **a** is recurrent if and only if each finite factor occurs at least twice. This means that the following simpler predicate suffices:

$$\forall N \geq 0, \ell \geq 1 \; \exists M \neq N \; \mathbf{a}[N..N+\ell-1] = \mathbf{a}[M..M+\ell-1].$$

Uniform recurrence

- For uniform recurrence, we need to express the fact that two consecutive occurrences of each factor are separated by no more than C positions.
- Since there are only finitely many factors of each length, we can take C to be the maximum of the constants corresponding to each factor of that length.
- Thus uniform recurrence corresponds to the following predicate:

$$\forall \ell \geq 1 \; \exists C \geq 1 \; \forall N \geq 0 \; \exists M \text{ with } N < M \leq N + C$$
$$\mathbf{a}[N..N + \ell - 1] = \mathbf{a}[M..M + \ell - 1].$$

Sequences with grouped factors

Cassaigne (1997) said a sequence $\mathbf{a}(a_i)_{i\geq 0}$ has *grouped factors* if for all $n \geq 1$, there exists some position $m = m(n)$ such that

$$\mathbf{a}[m..m + \rho(n) + n - 2]$$

contains all of the $\rho(n)$ length-n factors of $\mathbf{a}$, each factor occurring exactly once.

For example, the Fibonacci word has grouped factors.

Sequences with grouped factors

We can write a predicate for the property of having grouped factors, as follows:

$$\forall n \geq 1 \quad \exists m, s \geq 0 \quad \forall i \geq 0$$
$$\exists j,\ m \leq j \leq m+s \quad \mathbf{a}[i..i+n-1] = \mathbf{a}[j..j+n-1] \text{ and}$$
$$\forall j',\ m \leq j' \leq m+s, \quad j \neq j' \text{ we have } \mathbf{a}[i..i+n-1] \neq \mathbf{a}[j'..j'+n-1].$$

The first part of the predicate says that any length-n factor appears somewhere in the desired window, and the second says that it appears exactly once.

"...in what sense would a problem that required at least three alternating quantifiers to describe be natural?" – Homer and Selman, *Computability and Complexity Theory*, 2011.

Sequences with grouped factors

Open problem: is there an aperiodic automatic sequence with grouped factors?

Weaker property: having grouped factors for *infinitely many* n.

We can use our method to check this. The results are:

- The Thue-Morse sequence has grouped factors exactly for $n = 1$ and $n = 2^j + 1$, $j \geq 0$
- The period-doubling sequence has grouped factors exactly for $n = 2^j$, $j \geq 0$, and when $(n)_2$ starts with 11.

Orbit closure

- The *shift orbit* of a sequence $\mathbf{a} = a_0 a_1 a_2 \cdots$ is the set of all sequences under the shift, that is, the set

$$\mathcal{S} = \{a_i a_{i+1} a_{i+2} \cdots \ : \ i \geq 0\}.$$

- The *orbit closure* is the topological closure $\overline{\mathcal{S}}$ under the usual topology.
- In other words, a sequence $\mathbf{b} = b_0 b_1 b_2 \cdots$ is in $\overline{\mathcal{S}}$ if and only if, for each $j \geq 0$, the prefix $b_0 \cdots b_j$ is a factor of $\mathbf{a}$.
- Most sequences in the orbit closure of a k-automatic sequence are not automatic themselves.
- However, we can use our method to show that two distinguished sequences, the lexicographically least and lexicographically greatest sequences in the orbit closure, are indeed k-automatic.
- We were able to verify, for example, a recent result of Currie that the lexicographically least sequence in the orbit closure of the Rudin-Shapiro sequence $\mathbf{r}$ is $0\mathbf{r}$.

Unbordered factors

- A word is *bordered* if it can be expressed as uvu for words u, v with u nonempty, and otherwise it is unbordered.
- Currie and Saari proved that $\mathbf{t}$ has an unbordered factor of length n if $n \not\equiv 1 \pmod 6$.
- However, these are not the only lengths with an unbordered factor; for example,

  $$0011010010110100110010110100101$$

  is an unbordered factor of length 31.
- We can express the property that $\mathbf{t}$ has an unbordered factor of length ℓ as follows:

  $$\exists N \geq 0\ \forall j, 1 \leq j \leq \ell/2\ \mathbf{t}[N..N+j-1] \neq \mathbf{t}[N+\ell-j..N+\ell-1].$$

- Using our technique, we were able to prove

Theorem

There is an unbordered factor of length ℓ in $\mathbf{t}$ if and only iff $(\ell)_2 \notin 1(01^*0)^*10^*1$.

Enumeration

- What if we want to know the *number* of unbordered factors of length n, not just whether they exist or not?
- In many cases we can *count* the number $T(n)$ of length-n factors of an 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.

Enumeration

The basic idea is to create an automaton A accepting words of the form $(n, i)_k$, where each i corresponds to one of the distinct length-n factors we are trying to count.

The easiest way to do this is to let i be the position of the first occurrence of this factor.

Now the number of different paths in A whose first component spells out $(n)_k$ gives us the number of different i, and hence the number of distinct factors.

From A we can determine vectors v and x and matrices $M_0, M_1, \ldots, M_{k-1}$ such that if $[c_{i-1}c_{i-2}\cdots c_1 c_0]_k = n$, then

$$T(n) = v \cdots M_{c_{i-1}} M_{c_{i-2}} \cdots M_{c_1} M_{c_0} x^T.$$

Enumeration example: Subword complexity

- Subword complexity counts the number of distinct length-n factors of a sequence.
- To count these factors in an automatic sequence, we create a DFA M accepting the language

$$\begin{aligned} & \{ (\ell, i)_k \ : \ \mathbf{a}[i..i+\ell-1] \text{ is the first occurrence} \\ & \quad \text{of the given factor} \} \\ = \ & \{ (\ell, i)_k \ : \ \forall i' < i \ \mathbf{a}[i..i+\ell-1] \neq \mathbf{a}[i'..i'+\ell-1] \}. \end{aligned}$$

- the number of i corresponding to a given ℓ is just the number of subwords of length ℓ.

Enumeration: subword complexity

From this, for example, we can recover the well-known result on the subword complexity of the Thue-Morse sequence:

$$\rho(n) = \begin{cases} 2^n, & \text{if } 0 \leq n \leq 2; \\ 2n + 2^{t+2} - 2, & \text{if } 3 \cdot 2^t \leq n \leq 2^{t+2} + 1; \\ 4n - 2^t - 4, & \text{if } 2^t + 1 \leq n \leq 3 \cdot 2^{t-1}; \end{cases}.$$

Enumeration: unbordered factors

A little numerical experimention suggests the following:

Conjecture

Let $f(n)$ denote the number of unbordered factors of length n in $\mathbf{t}$, the Thue-Morse sequence. Then f is given by $f(0) = 1$, $f(1) = 2$, $f(2) = 2$, and the system of recurrences

$$\begin{aligned}
f(4n) &= 2f(2n), && (n \geq 2)\\
f(4n+1) &= f(2n+1), && (n \geq 0)\\
f(8n+2) &= f(2n+1) + f(4n+3), && (n \geq 1)\\
f(8n+3) &= -f(2n+1) + f(4n+2) && (n \geq 2)\\
f(8n+6) &= -f(2n+1) + f(4n+2) + f(4n+3) && (n \geq 2)\\
f(8n+7) &= 2f(2n+1) + f(4n+3) && (n \geq 3)
\end{aligned}$$

for $n \geq 0$.

Proving the conjecture on number of unbordered factors

- Express "first occurrence of unbordered factor $\mathbf{t}[i..i+n-1]$ is at position i" as a predicate $P(n,i)$
- Translate the predicate to a DFA accepting the language $\{(n,i)_2 \ : \ P(n,i)\}$
- Translate the DFA to vectors v, x and matrices M_0, M_1 such that if $n = [c_{j-1} \cdots c_1 c_0]_2$ then

$$f(n) = vM_{c_{j-1}} \cdots M_{c_1} M_{c_0} x^T$$

- Assertions about f correspond to certain assertions about M, a matrix that could occur as a product of the M_0 and M_1
- For example, the identity $f(8n+2) = f(2n+1) + f(4n+3)$

$$vMM_0M_1M_0x^T = vMM_1x^T + vMM_1M_1x^T, \qquad (2)$$

where M is the matrix product corresponding to the base-2 expansion of n.

Proving the conjecture on number of unbordered factors

- We have now reduced the problem to a finite verification.
- To reduce the amount of work even further, we can work with vM instead of M.
- With our identities in hand, we can now easily prove results such as
  - $f(n) \leq n$ for $n \geq 4$ and
  - $f(n) = n$ infinitely often.

Enumeration

In a similar way, we can handle

- palindrome complexity (the number of distinct length-n palindromic factors)
- the number of words whose reversals are also factors;
- the number of squares of a given length;
- the number of unbordered factors

and so forth.

Are there some properties that are not expressible?

- A difficult candidate: abelian properties
- We say that a nonempty word x is an *abelian square* if it of the form ww' with $|w| = |w'|$ and w' a permutation of w. (An example in English is the word `reappear`.)
- Luke Schaeffer has recently shown that the predicate for abelian squarefreeness of the paperfolding sequence is indeed *inexpressible* in $\mathrm{Th}(\mathbb{N}, +, 0, 1, <, V_k)$
- Nevertheless, some abelian properties are decidable by other means (Currie & Rampersad)

Synchronized sequences

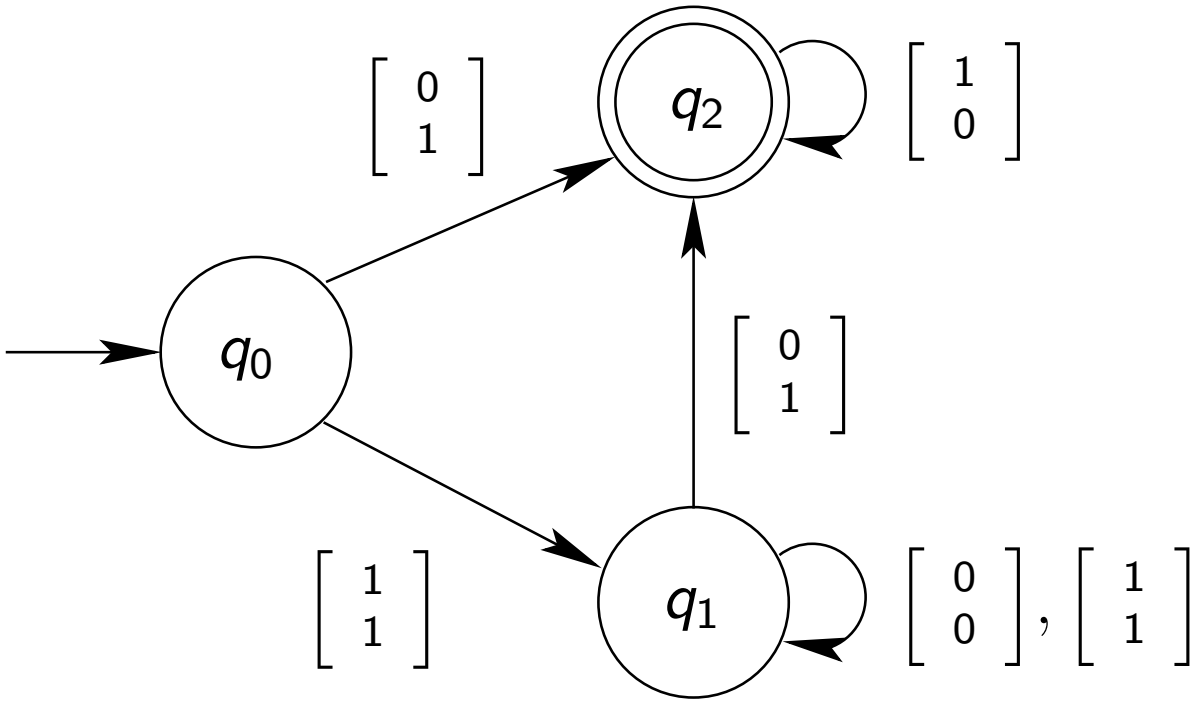
(Carpi) A sequence $f : \mathbb{N} \to \mathbb{N}$ is *k*-synchronized if its graph

$$\{ (n, f(n))_k \ : \ n \geq 0\}$$

is a regular language.

Synchronized sequences

Example. The function $f(n) = n + 1$ is k-synchronized. For example, for $k = 2$, the following automaton suffices:



Why synchronized sequences?

- If $f(n)$ is k-synchronized, then
  we can compute $f(n)$ in $O(\log n)$ time

- If $f(n)$ is k-synchronized, then $f(n) = O(n)$

Efficient computation of synchronized sequences

To compute $f(n)$ in $O(\log n)$ time:

- On input n, construct the $O(\log n)$-state machine M' accepting those words with first component of the form $0^*(n)_k$ and second component anything
- Intersect, using the familiar direct product construction, with the DFA M accepting $\{ (n, f(n))_k \ : \ n \geq 0\}$
  - Resulting automaton accepts exactly one word
  - Find accepting path using depth-first search
  - Label of accepting path gives $f(n)$ in base k

Growth bounds

Theorem. If $f(n)$ is k-synchronized, then $f(n) = O(n)$.

Proof.

- Suppose f is k-synchronized and accepted by a DFA with t states.
- If $f(n) \neq O(n)$, then there exists n such that $f(n) > k^t n$.
- So the base-k representation of $(n, f(n))$ starts with at least t zeros in the first component, and a nonzero symbol in the second component.
- Apply the pumping lemma to $z = (n, f(n))_k$
- We see that "pumping" gives a new word in the language with n unchanged, but $f(n)$ increased.
- But f is a function, a contradiction. ■

Closure properties of synchronized sequences

The class of k-synchronized sequences is closed under

- sum
- $\mathbb{N}$-linear combination
- $f(n) \to \lfloor \alpha f(n) \rfloor$ for α rational
- term-wise maximum and minimum
- running maximum: $g(n) = \max_{0 \le i < n} f(i)$
- discrete inverse: $g(n) = \min\{i \ : f(i) \ge n\}$
- composition

Many aspects of k-automatic sequences are k-synchronized

Example: the appearance function.

$A_{\mathbf{x}}(n) =$ 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 $\mathbf{x}[0..t-1]$

Appearance function predicate

$$
\begin{aligned}
L = \{(n,t)_k \quad : \quad & \forall\, i \geq 0\ \exists j \leq t-n \\
& \text{such that } \mathbf{x}[i..i+n-1] = \mathbf{x}[j..j+n-1] \\
& \text{and } \forall\, j' < t-n \\
& \mathbf{x}[j'..j'+n-1] \neq \mathbf{x}[t-n..t-1]\}.
\end{aligned}
$$

Here $t = A_{\mathbf{x}}(n)$.

Other synchronized functions

- **separator function**: length of the shortest factor of **x** beginning at position n that never appeared previously in **x** (Carpi & Maggi, 2001)
- **repetitivity index**: the minimal distance between two consecutive occurrences of the same length-n factor in **x** (Carpi & D'Alonzo, 2009)
- **recurrence function**: size of the smallest "window" always guaranteed to contain all length-n factors in **x** (Charlier & Rampersad & S, 2011)

Subword complexity

$\rho_{\mathbf{x}}(n) =$ number of distinct length-n factors of $\mathbf{x}$

- known to be k-regular
- known to be $O(n)$ for k-automatic sequences
- suggests it could be k-synchronized

Novel occurrences

Call a length-n factor *novel* at position i if it occurs there but in no earlier location.

Here is a predicate for novel factors:

$$\{(n,i)_k \ : \ \forall j, 0 \leq j < i \quad \mathbf{x}[i..i+n-1] \neq \mathbf{x}[j..j+n-1]\}$$

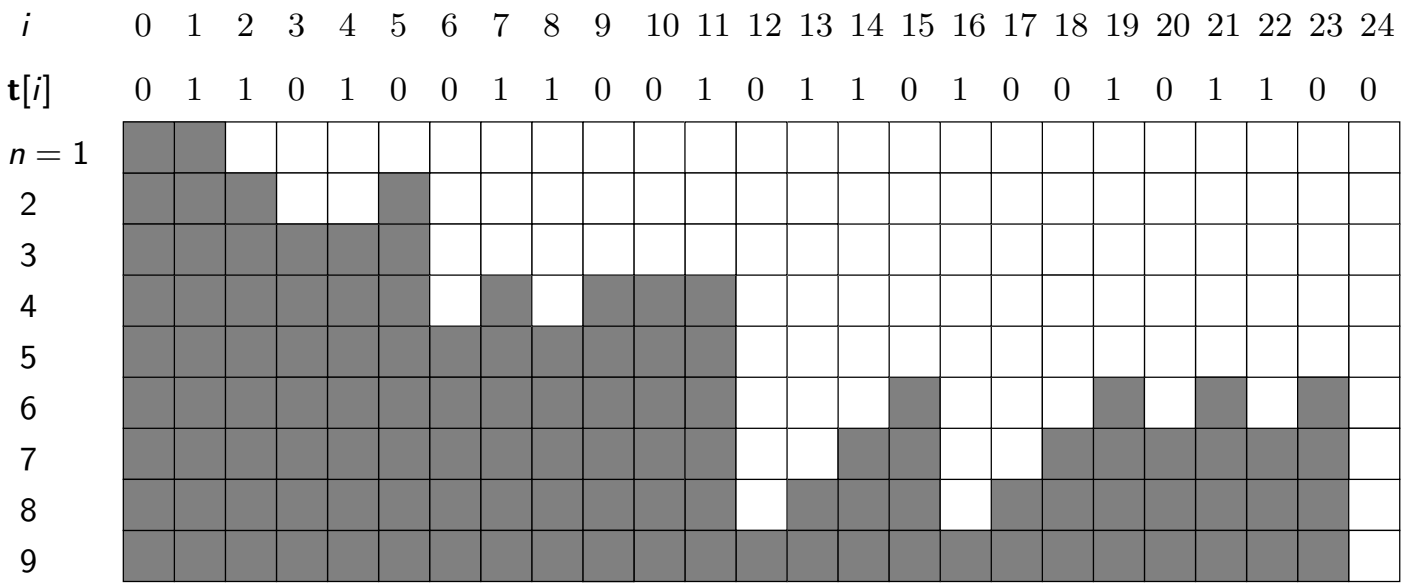
Theorem. In any sequence of linear complexity, the starting positions of novel occurrences of factors are "clumped together" in a bounded number of contiguous blocks.

Novel factors for Thue-Morse

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 $1 \leq n \leq 9$.



Bound on number of contiguous blocks

Theorem

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. Base case easy.
Assume true for $n-1$. We prove 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.

Bound for Thue-Morse

For Thue-Morse example, it is well-known 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 $n = 6$.

Linear complexity

Corollary

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 (1996) we know that 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. □

Subword complexity of automatic sequences is k-synchronized

Theorem

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. □

Unsynchronized sequences

Are other aspects of k-automatic sequences always k-synchronized?

No.

We say 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

The characteristic sequence $\mathbf{c} = 0110100010\cdots$ *is 2-automatic, but the function* $u_{\mathbf{c}}(n)$ *counting the number of unbordered factors is not 2-synchronized.*

Open Problems

- Extend these ideas to morphic sequences (fixed points of possibly non-uniform morphisms, followed by a coding)
- Is $\sup\{x/y : (x,y)_k \in L\}$ computable for context-free languages L?
- Given a regular language $L \subseteq (\Sigma_k \times \Sigma_k)^*$ representing a a set $S \subseteq \mathbb{N} \times \mathbb{N}$ of pairs of natural numbers, is it decidable if S contains a pair (p, q) with $p \mid q$?