Subword Complexity and k-Synchronization

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 Daniel Goč and Luke Schaeffer.

Representations of numbers

- Represent elements of $\mathbb{N} = \{0, 1, 2, \ldots\}$ as words in base k over the alphabet $\Sigma_k = \{0, 1, \ldots, k-1\}$
- Canonical representation $(n)_k$ has no leading zeros
- Represent pairs of integers (m, n) as words over the alphabet $\Sigma_k \times \Sigma_k$
  - For example,

$$(43, 17)_2 = \left[\begin{array}{c} 1 \\ 0 \end{array}\right]\left[\begin{array}{c} 0 \\ 1 \end{array}\right]\left[\begin{array}{c} 1 \\ 0 \end{array}\right]\left[\begin{array}{c} 0 \\ 0 \end{array}\right]\left[\begin{array}{c} 1 \\ 0 \end{array}\right]\left[\begin{array}{c} 1 \\ 1 \end{array}\right].$$

  - Canonical representation $(m, n)_k$ has no leading $\left[\begin{array}{c} 0 \\ 0 \end{array}\right]$'s

Synchronized sequences

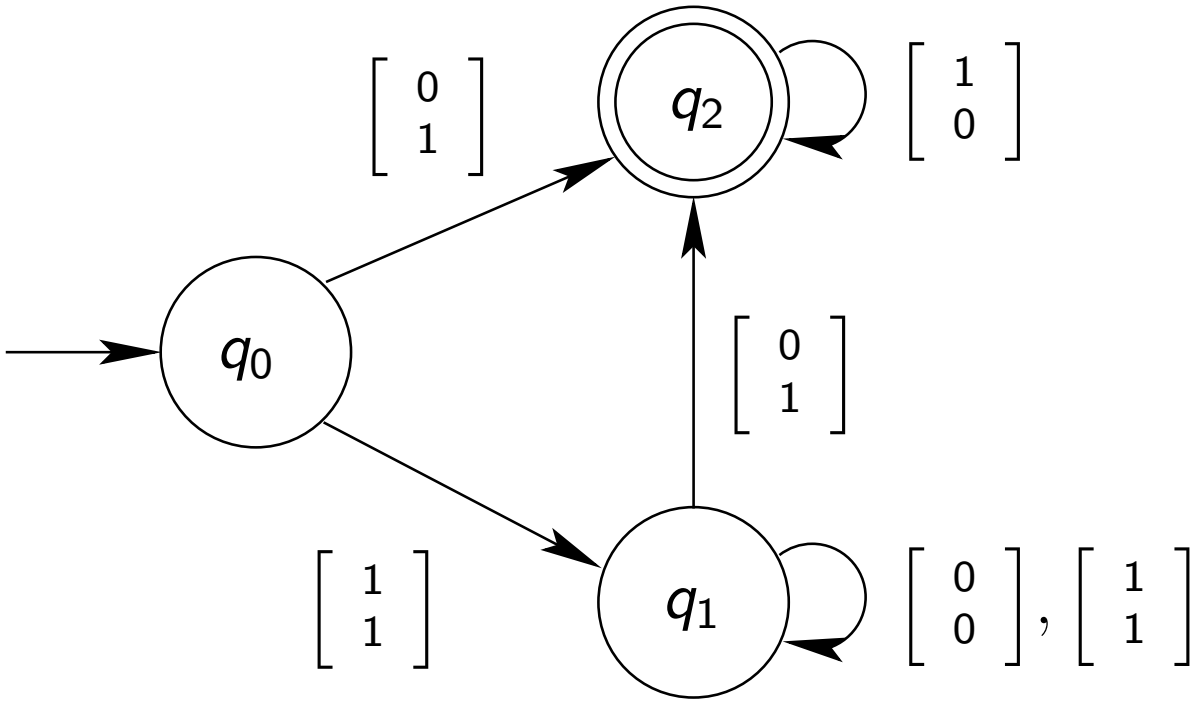
(Carpi) A sequence $f : \mathbb{N} \to \mathbb{N}$ is k-synchronized if its graph

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

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 $f(n) = O(n)$
- If $f(n)$ is non-decreasing and k-synchronized, then either
  $f(n) = \Theta(1)$ or $f(n) = \Theta(n)$

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

What is an automatic sequence?

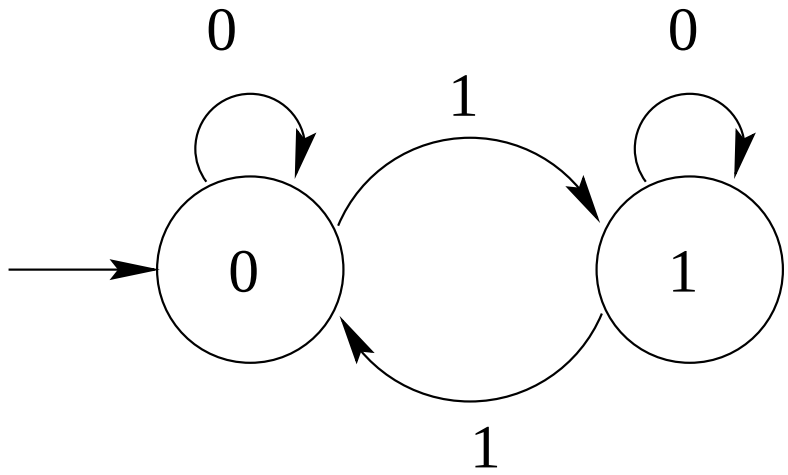
- An infinite sequence

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

over a finite alphabet of letters, generated by a finite 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

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

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

$$
\begin{aligned}
L = \{(n,t)_k \quad : \quad & \forall\, i \geq 0\ \exists j \leq t-n \\
& \text{such that } \mathbf{x}[i..i+n-1] = \mathbf{x}[j..j+n-1] \\
& \text{and } \forall\, \ell < t-n \\
& \mathbf{x}[\ell..\ell+n-1] \neq \mathbf{x}[t-n..t-1]\}.
\end{aligned}
$$

The general technique

Theorem. If a subset S of $\mathbb{N}^r$ is defined using a predicate involving

- $\forall$, $\exists$
- indexing into a k-automatic sequence $\mathbf{x}$
- addition or subtraction of indices, or multiplication by constant
- comparison of integers or symbols of $\mathbf{x}$

then the set of base-k representations of elements of S is regular. Furthermore, given the predicate, it can be mechanically translated into an automaton accepting the corresponding language L.

Corollary.
The appearance function is k-synchronized.

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
- this 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:

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

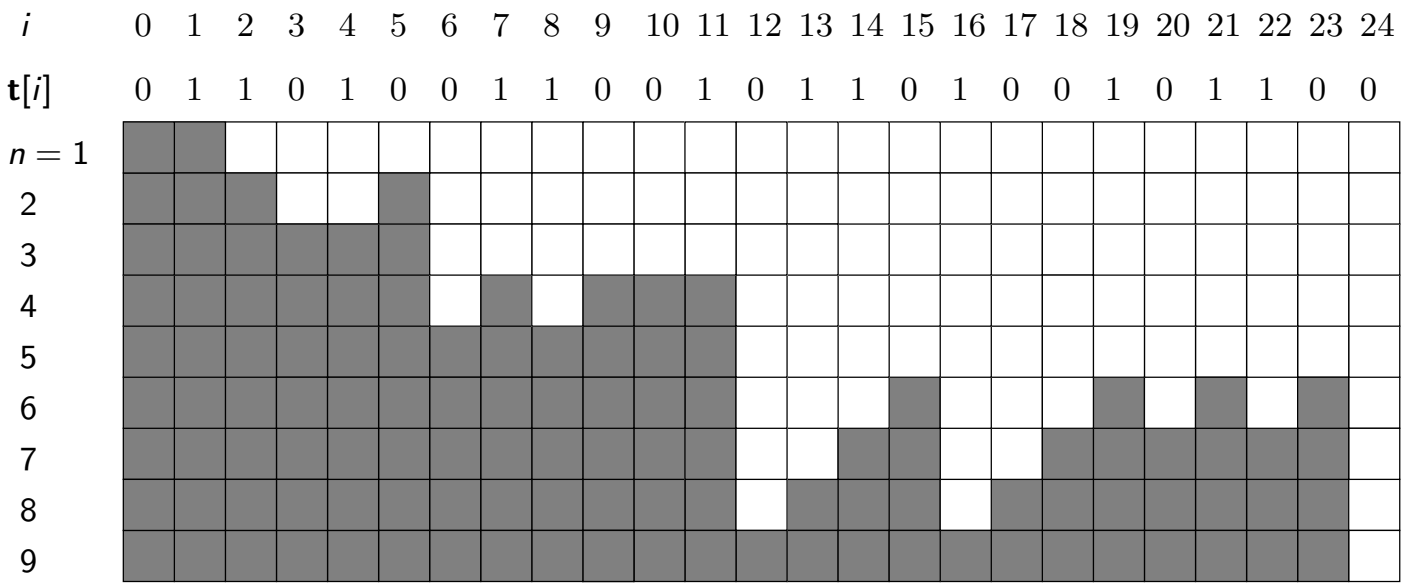
Theorem. In any sequence of linear complexity, the starting positions of novel occurrences of factors are "clumped together" in a bounded number of contiguous blocks.

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.

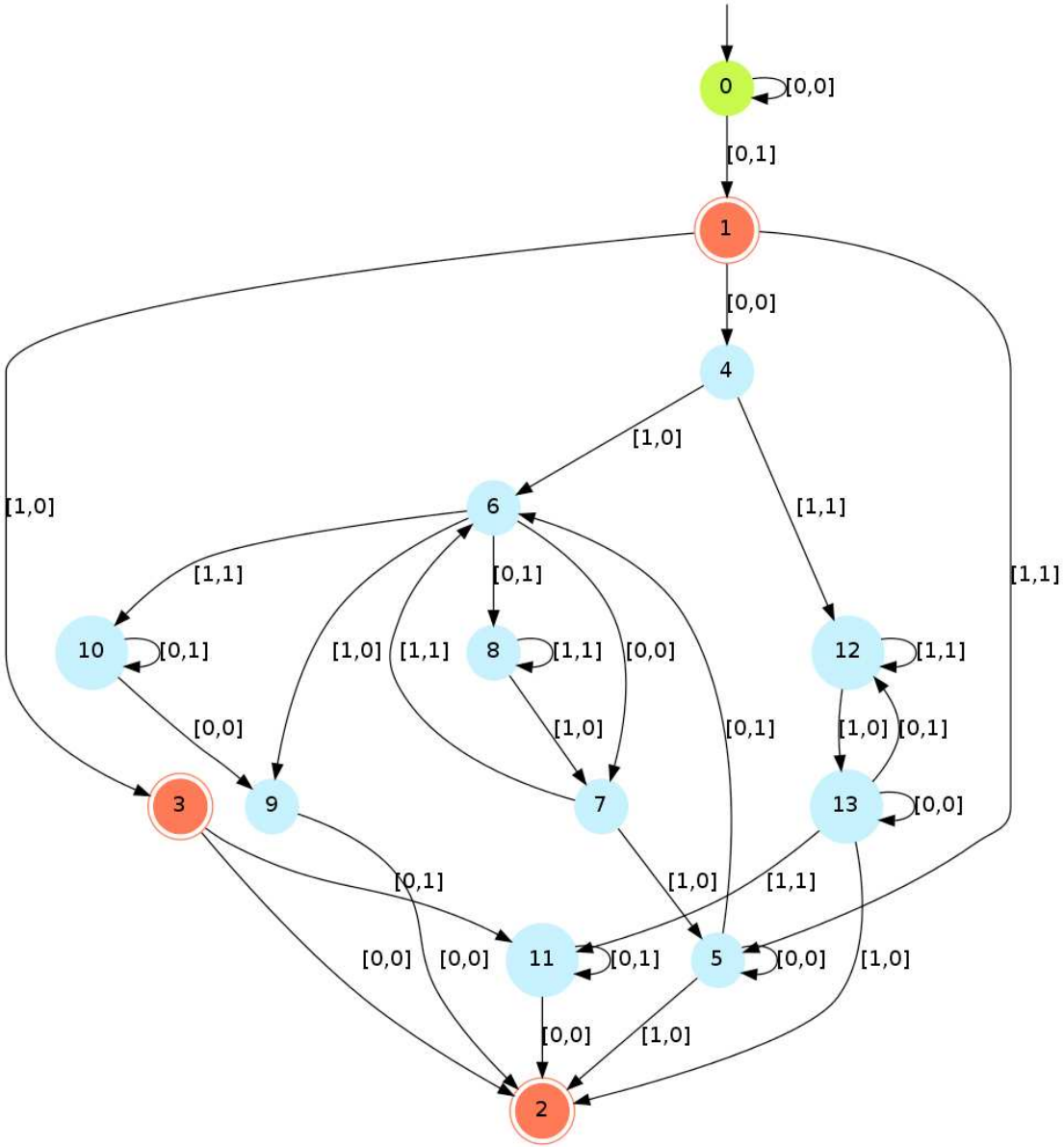
Finally, verify that m is the sum of the sizes of the blocks. □

A corollary

Corollary

Given a k-automatic sequence $\mathbf{x}$, there is an algorithm that, on input n in base k, will compute the subword complexity $\rho_{\mathbf{x}}(n)$ expressed in base k in time $O(\log n)$.

Example: subword complexity for Thue-Morse



Applications: an improvement on Goldstein

Corollary

There is an algorithm, that, given a k-automatic sequence $\mathbf{x}$*, will compute*

- $\sup_{n \geq 1} \rho_{\mathbf{x}}(n)/n$,
- $\limsup_{n \geq 1} \rho_{\mathbf{x}}(n)/n$,
- $\inf_{n \geq 1} \rho_{\mathbf{x}}(n)/n$, *and*
- $\liminf_{n \geq 1} \rho_{\mathbf{x}}(n)/n$.

Proof.

We already showed how to construct an automaton accepting $\{(n, \rho_{\mathbf{x}}(n))_k \ : \ n \geq 1\}$. Using Schaeffer & S (2012), we can compute the sup, lim sup etc. □

Powers

- A nonempty word w is a *power* if $w = x^k$ for some word x and integer $k \geq 2$.
- Example in French: `chercher`.
- Otherwise w is *primitive*.

Conjugates

- Two finite words x, y are conjugates if one is a cyclic shift of the other; in other words, if there exist words u, v such that $x = uv$ and $y = vu$.
- For example, `draper` is a conjugate of `perdra`.

Results

Theorem

If $\mathbf{x}$ *is* k*-automatic, then the following are* k*-synchronized:*

- *the function counting the number of distinct length-*n *factors that are powers;*
- *the function counting the number of distinct length-*n *factors that are primitive words.*

Sketch of proof

Main ideas:

- A word x is a power if and only if there exist nonempty words y, z such that $x = yz = zy$.
- Thus, we can express the predicate $P(i, j) :=$ "$\mathbf{x}[i..j]$ is a power" as follows: "there exists d, $0 < d < j - i + 1$, such that $\mathbf{x}[i..j - d] = \mathbf{x}[i + d..j]$ and $\mathbf{x}[j - d + 1..j] = \mathbf{x}[i..i + d - 1]$".
- Furthermore, we can express the predicate $P'(i, n) :=$ "$\mathbf{x}[i..i + n - 1]$ is a length-n power and is a novel occurrence of that factor in $\mathbf{x}$".
- We show that once again the novel occurrences of length-n powers are clustered into a finite number of blocks.

Sketch of proof

- Then we can nondeterministically guess the endpoints of these blocks, and verify that the length-n factors beginning at the positions inside the blocks are novel occurrences of powers, while those outside are not, and sum the lengths of the blocks, using a finite automaton built from M.
- So the counting function for powers is k-synchronized.
- The number of length-n primitive words in $\mathbf{x}$ is then also k-synchronized, since it is expressible as the total number of words of length n, minus the number of length-n powers.

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.*