Decidability 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

What is an automatic sequence?

- An infinite sequence

\[ a = a_0a_1a_2 \ldots \]

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

\[ t = (t_n)_{n \geq 0} = 0110100110010110 \cdots. \]
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 English: alfalfa.)
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 $n \geq 0$.
4. $t$ has infinitely many distinct palindromic factors (A palindrome is a word equal to its reverse, like radar.)
5. The number $p(n)$ of distinct palindromic factors of length $n$ in $t$ is given by

\[
p(n) = \begin{cases} 
0, & \text{if } n \text{ odd and } n \geq 5; \\
1, & \text{if } n = 0; \\
2, & \text{if } 1 \leq n \leq 4, \text{ or } n \text{ even and } 3 \cdot 4^k + 2 \leq n \leq 4^{k+1}; \\
4, & \text{if } n \text{ even and } 4^k + 2 \leq n \leq 3 \cdot 4^k. 
\end{cases}
\]
6. \( t \) is \textit{mirror-invariant}: if \( x \) is a finite factor of \( t \), then so is its reverse \( x^R \).

7. \( t \) is \textit{recurrent}, that is, every factor that occurs, occurs infinitely often.

8. \( t \) is \textit{uniformly recurrent}, that is, for all factors \( x \) occurring in \( t \), there is a constant \( c(x) \) such that two consecutive occurrences of \( x \) are separated by at most \( c(x) \) symbols.

9. \( t \) is \textit{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 \( e = \lfloor \log_2(n - 2) \rfloor \).
10. The lexicographically least sequence in the shift orbit closure of $t$ is $\overline{t_1} \overline{t_2} \overline{t_3} \cdots$, 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

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

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...
Let \( Th(\mathbb{N}, +, 0, 1, <) \) 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:

\[
(\forall n > 43 \ \exists x, y, z \geq 0 \text{ such that } n = 6x + 9y + 20z) \land \\
\neg(\exists x, y, z \geq 0 \text{ such that } 43 = 6x + 9y + 20z).
\]

Here, of course, “6x” is shorthand for the expression “\( x + x + x + x + x + x \)”, and similarly for 9y and 20z.

Presburger proved that \( Th(\mathbb{N}, +, 0, 1, <) \) is decidable: that is, there exists an algorithm that, given a sentence in the theory, will decide its truth.
represent integers in an integer base $k \geq 2$ using the alphabet $\Sigma_k = \{0, 1, \ldots, k - 1\}$.

represent $n$-tuples of integers as words over the alphabet $\Sigma_k^n$, padding with leading zeroes, if necessary.

For example, the pair $(21, 7)$ can be represented in base 2 by the word

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

$$\{[a, b, c] : a + b = c\}$$
$$\{[a, b, c] : a + b + 1 = c + k\}$$
$$\{[a, b, c] : a + b + 1 = c\}$$
$$\{[a, b, c] : a + b = c + k\}$$

**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.
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.

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

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

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

- Nevertheless, an implementation often succeeds in verifying statements in the theory
- More information in the talk of Daniel Goč
Deciding periodicity

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

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

- When we run this on the Thue-Morse sequence, we discover (as expected) that \( \mathbf{t} \) is not ultimately periodic.
Repetitions

Thue (1912) proved that $t$ contains no overlaps; that is, $t$ 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: $a[N..N + p] = a[N + p..N + 2p]$.

So the corresponding automatic predicate for $t$ is

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

or, in other words,

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

We programmed up our decision procedure and verified that indeed $t$ is overlap-free.
We can define more general repetitions as follows: a word $x$ is an $\alpha$-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 $\alpha$-powers, using the following predicate:

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

However, this observation alone does not suffice to compute the so-called critical exponent of $a$, which is the supremum over all rational $\alpha$ such that $a$ has $\alpha$-power factors.

It turns out that the critical exponent is also computable for automatic sequences.
We can express the property that $a$ is mirror-invariant as follows:

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

which is the same as

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

which can be easily checked by our method.
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:

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

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:

$$\forall N \geq 0, \ell \geq 1 \exists M \neq N \ a[N..N+\ell-1] = a[M..M+\ell-1].$$
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:

$$\forall \ell \geq 1 \ \exists C \geq 1 \ \forall N \geq 0 \ \exists M \ \text{with} \ N < M \leq N + C \ \ a[N..N+\ell-1] = a[M..M+\ell-1].$$
The **shift orbit** of a sequence \( a = a_0a_1a_2 \cdots \) is the set of all sequences under the shift, that is, the set

\[
S = \{a_ia_{i+1}a_{i+2} \cdots : i \geq 0\}.
\]

The **orbit closure** is the topological closure \( \overline{S} \) under the usual topology.

In other words, a sequence \( b = b_0b_1b_2 \cdots \) is in \( \overline{S} \) if and only if, for each \( j \geq 0 \), the prefix \( b_0 \cdots b_j \) is a factor of \( a \).

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.

More in the talk of Narad Rampersad.
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 $t$ 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,

\[ 00110100101100101100101100101 \]

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

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

- Using our technique, we were able to prove **Theorem**

*There is an unbordered factor of length $\ell$ in $t$ if and only if* \((\ell)_2 \not\in 1(01*0)^*10*1.*
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.
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

$$\{(n, \ell)_k : a[n..n + \ell - 1] \text{ is the first occurrence of the given factor}\}$$

$$= \{(n, \ell)_k : \forall n' < n \ a[n..n + \ell - 1] \neq a[n'..n' + \ell - 1]\}.$$  

- the number of $n$ corresponding to a given $\ell$ is just the number of distinct subwords of length $\ell$
- this number can be expressed as the product

$$vM_{a_1} \cdots M_{a_i}w$$

for suitable vectors $v, w$ and matrices $M_0, \ldots, M_{k-1}$, where $a_1 \cdots a_i$ is the base-$k$ representation of $\ell$, thus giving an efficient algorithm to compute it.
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.
What other properties of automatic sequences are decidable?

- 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 shows that the predicate for abelian squarefreeness is indeed inexpressible in $\text{Th}(\mathbb{N}, +, 0, 1, <, V_k)$
- Nevertheless, some abelian properties are decidable by other means — see the talk of James Currie
The other talks

- Daniel Goč: Automatic theorem-proving in automatic sequences
- Luke Schaeffer: Abelian powers in automatic sequences are not always automatic
- Narad Rampersad: Extremal words in the shift orbit closure of a morphic sequence
- James Currie: Abelian powers and patterns in words: problems and perspectives