Automatically Proving Theorems in Combinatorics on Words Using a Computer

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

Seven Points of the Talk

1. A decision procedure for answering questions about a large class of interesting sequences exists (and handles famous sequences such as Thue-Morse, Rudin-Shapiro, etc.), based on first-order logic
2. Many properties that have been studied in the literature can be phrased in first-order logic (including some for which this is not obvious!)
3. The decision procedure is relatively easy to implement and often runs remarkably quickly, despite its formidable worst-case complexity — and we have an implementation that is publicly available
4. The method can also be used to not simply decide, but also *enumerate*, many aspects of sequences

Seven Points of the Talk

5. Many results already in the literature (in dozens of papers and Ph. D. theses) can be reproved by our program in a matter of seconds (including fixing at least one that was wrong!)
6. Many new results can be proved
7. There are some well-defined limits to what we can do because either
   - the property is not expressible in first-order logic; or
   - the underlying sequence leads to undecidability

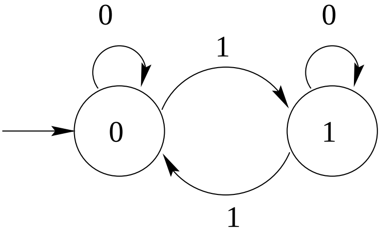
For which sequences does it work?

- One large class: the class of k-automatic sequences
- These are infinite sequences

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

  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*

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

What kind of properties can we handle?

1. Ultimate periodicity: $\mathbf{t}$ is not ultimately periodic.
2. Repetitions: $\mathbf{t}$ 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. $\mathbf{t}$ 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. Palindromes: $\mathbf{t}$ 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 $\mathbf{t}$ 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 **t**

6. **t** is *mirror-invariant*: if x is a finite factor of **t**, then so is its reverse x^R.
7. Recurrence: **t** is *recurrent*, that is, every factor that occurs, occurs infinitely often.
8. **t** is *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 *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 $\mathbf{t}$

10. Dynamical systems: the lexicographically least sequence in the shift orbit closure of $\mathbf{t}$ is $\overline{t_1}\ \overline{t_2}\ \overline{t_3}\ \cdots$, which is also 2-automatic.
11. The *subword complexity* $\rho(n)$ of $\mathbf{t}$, which is the function counting the number of distinct factors of $\mathbf{t}$, 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. $\mathbf{t}$ 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.)

Hilbert's dreams

- To show that every true statement is provable (killed by Gödel)
- To provide an algorithm to decide if an input statement is provable (killed by Turing)
- Nevertheless, some subclasses of problems are decidable — i.e., an algorithm exists guaranteed to prove or disprove any statement

First-order 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.
- This is sometimes called *Presburger arithmetic*.
- Here we are allowed to use any number of variables, logical connectives like "and", "or", "not", etc., and quantifiers like $\exists$ and $\forall$.

Example: The Chicken McNuggets Problem

A famous problem in elementary arithmetic books in the US:

At McDonald's, Chicken McNuggets are available in packs of either 6, 9, or 20 nuggets. What is the largest number of McNuggets that one cannot purchase?

Presburger arithmetic

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

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

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

Presburger's theorem

Figure: Mojżesz Presburger (1904–1943)

Presburger 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. He used quantifier elimination.

Decidability of Presburger arithmetic: Rabin's proof

Rabin found a much simpler proof of Presburger's result, based on automata.

Ideas:

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

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

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".

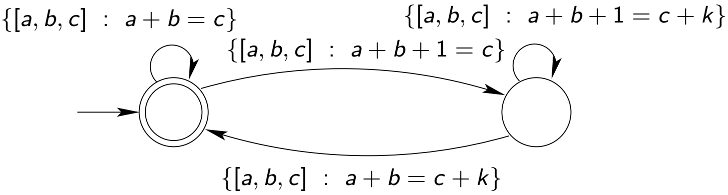


Figure: Checking addition in base k

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.

The bad news

- The worst-case running time of the algorithm above is bounded above by
$$2^{2^{\cdot^{\cdot^{\cdot^{2^{p(N)}}}}}},$$
where the number of 2's in the exponent is equal to the number of quantifier alternations, p is a polynomial, and N is the number of states needed to describe the underlying automatic sequence.
- This bound can be improved to double-exponential.

The good news

- With a small extension to Presburger's logical theory — adding the function $V_k(n)$, the largest power of k dividing n — one can also verify many more interesting statements (examples to follow). But then the worst-case time bound returns to

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

- Beautiful theory due to Büchi, Bruyère, Hansel, Michaux, Villemaire, etc.
- Despite the awful worst-case bound on running time, an implementation often succeeds in verifying statements in the theory in a reasonable amount of time and space.
- Many old results from the literature can been verified with this technique, and many new ones can be proved.

Deciding periodicity

First example:

- 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^ω for a (possibly empty) finite word y.
- Honkala (1986) proved that ultimate periodicity is decidable for automatic sequences.
- Using this 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 $\mathbf{t}$ is not ultimately periodic.

Repetitions

- Thue (1912) proved that $\mathbf{t}$ contains no overlaps; that is, $\mathbf{t}$ is overlap-free.
- Using this 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,

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

- Our program easily verifies that indeed $\mathbf{t}$ is overlap-free.

Mirror invariance

We can express the property that **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 the 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:

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

- 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:

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

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:

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

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 the method to show that two distinguished sequences, the lexicographically least and lexicographically greatest sequences in the orbit closure, are indeed k-automatic.

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 this technique, we were able to prove

Theorem

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

Balance

- Let $|w|_a$ denote the number of occurrences of the letter a in the word w.
- A word z is said to be *balanced* if for all finite factors w, x of the same length and all alphabet symbols a we have

$$\left| |w|_a - |x|_a \right| \leq 1.$$

- At first glance it is not obvious how to express this property in first-order arithmetic. (How do we count symbols?)

Balance

- Luckily, for *binary* words over $\{0, 1\}$ there is an equivalent characterization: a word z is *unbalanced* if and only if there exists a palindrome u such that both $0u0$ and $1u1$ are factors of z.
- Now *that* is a first-order statement!
- So we can, for example, write a predicate for all the balanced factors of Thue-Morse.
- The result: there are exactly 41 balanced factors of the Thue-Morse word, and the longest is of length 8.

Enumeration

- 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

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

- the number of n corresponding to a given ℓ is just the number of distinct subwords of length ℓ
- 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 ℓ, thus giving an efficient algorithm to compute it.

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.

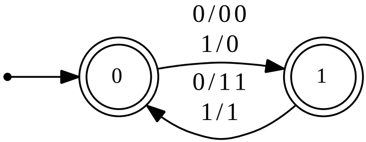
Getting some new results

- We are interested in binary words avoiding the pattern xxx^R.
- An example of this pattern in English is contained in the word

  `bepepper`.
- Are there infinite binary words avoiding this pattern?
- Of course: $(01)^\omega = 010101\cdots$.
- But there are other periodic examples, like $(0010011011)^\omega$, that also avoid the pattern
- Are there aperiodic examples?

Avoidability

Yes! Take the infinite Fibonacci word $\mathbf{f}$ (generated by iterating the morphism $0 \to 01$, $1 \to 0$)and run it through the following transducer:



obtaining the infinite word

$$\mathbf{r} = 001001101101100100110110110010010010011011001001001101100 \cdots .$$

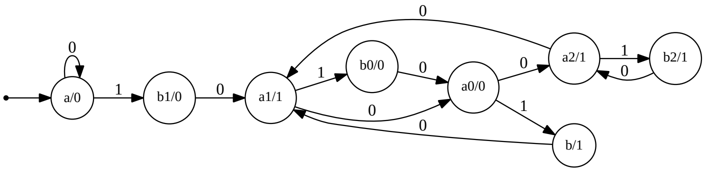
Claim: it avoids the patterns xxx^R and also xx^Rx^R.

Avoiding xxx^R

To prove this we use the predicate

$$\exists i \geq 0 \forall t,\ 0 \leq t < n\, (\mathbf{r}[i+t] = \mathbf{r}[i+t+n]) \wedge (\mathbf{r}[i+t] = \mathbf{r}[i+3n-1-t]),$$

which says that the first block of length n equals the second block, and the first block equals the reverse of the the third block.
The word $\mathbf{r}$ itself is generated by an 8-state automaton:



When we run this predicate on the automaton, we get that only length $n = 0$ is accepted. So the pattern xxx^R doesn't occur. This takes about 8 seconds on a laptop and the largest intermediate automaton has 8304 states.

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 showed that the predicate for abelian squarefreeness is indeed inexpressible in $\mathrm{Th}(\mathbb{N}, +, 0, 1, V_k)$
- However, for some sequences (e.g., Thue-Morse, Fibonacci) many abelian properties are decidable

Other limits to the approach

- Consider the morphism $a \to abcc$, $b \to bcc$, $c \to c$.
- The fixed point of this morphism is

$$\mathbf{s} = abccbccccbccccccbccccccccb\cdots$$

- It encodes, in the positions of the b's, the characteristic sequence of the squares.
- So the first-order theory $\mathrm{Th}(\mathbb{N}, +, 0, 1, n \to \mathbf{s}[n])$ is powerful enough to express the assertion that "n is a square"
- With that, one can express multiplication, and so it is undecidable.

Three Open Problems

- Is there a first-order characterization of the balance property for alphabets of more than two symbols?
- Let p denote the characteristic sequence of the prime numbers. Is the logical theory $\mathrm{Th}(\mathbb{N}, +, 0, 1, n \to p(n))$ decidable?
- Is the following problem decidable? Given two k-automatic sequences $(a(n))_{n\geq 0}$ and $(b(n))_{n\geq 0}$, are there integers $c \geq 1$ and $d \geq 0$ such that $a(n) = b(cn + d)$ for all n?

The Walnut Prover

Our publicly-available prover, written by Hamoon Mousavi, is called `Walnut` and can be downloaded from

`www.cs.uwaterloo.ca/~shallit/papers.html` .