Automatic Theorem-Proving in Automatic Sequences

Daniel Goč
School of Computer Science, University of Waterloo
Waterloo, Ontario N2L 3G1, Canada
dgoc@cs.uwaterloo.ca

(Joint work with Luke Schaeffer and Jeffrey Shallit)

What are k-automatic sequences?

Let $\mathbf{x} = (a(n))_{n\geq 0}$ be an infinite sequence over a finite alphabet Δ.

- $\mathbf{x}$ is said to be *k-automatic* if there is a deterministic finite automaton M taking as input the base-k representation of n, and having $a(n)$ as the output associated with the last state encountered.
- In this case, we say that M *generates* the sequence $\mathbf{x}$.

Some notation:

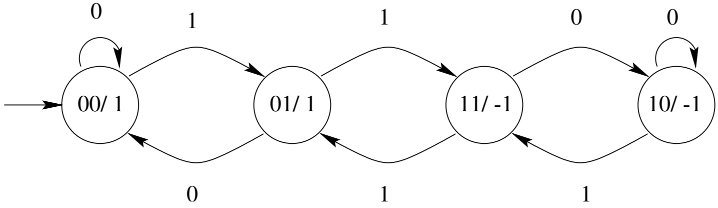
- $\mathbf{x}[i..j]$ denotes the factor of $\mathbf{x}$ starting at position i and ending at position j
- $(n)_k$ is the k-ary expansion of n without leading zeroes.
- For example: $(13)_2 = 1101$

The Rudin-Shapiro sequence

The Rudin-Shapiro sequence is the count, modulo 2, of the number of (possibly overlapping) occurrences of 11 in $(n)_2$.

$\mathbf{r} = r(0)r(1)r(2)\cdots = 00010010000111010001001011 1000\cdots$

The sequence is generated by the following base-2 DFAO:



The input is n, expressed in base 2, and the output is the number contained in the state last reached.

Basic Idea

The basic idea is:

- given an automaton M for a k-automatic sequence for which we have a query
- we convert our query into first order logic predicate $P(n)$
- we parse $P(n)$ and we carefully alter M by a series of transformations to get a new automaton M'
- M' accepts the base-k representations of those integers n for which $P(n)$ is true
- we then interpret M' to characterize the predicate $P(n)$ (we can check if M' accepts a finite language, everything, nothing, etc. . .)

Building blocks

The types of questions we can ask correspond to formal logic predicates built from the following building blocks:

- **comparison**(i, j) which accepts iff $i < j$, (or $i \leq j$, or $i = j$)
- **addition** and **multiplication by constants** of the input numbers
- **match**(i, j) which accepts input (i, j) if $\mathbf{x}[i] = \mathbf{x}[j]$ (alternatively $\mathbf{x}[i] < \mathbf{x}[j]$) where $\mathbf{x}$ is the given k-automatic sequence.
- the normal logical connectives: **and** $(\vee)$, **or** $(\wedge)$, **implies** $(\rightarrow)$
- the complement operator **not** $(\neg)$
- quantifiers (over variables): **for all** $(\forall i)$ and **there exists** $(\exists i)$

Theory

Jeff already mentioned the decidability of *Presburger arithmetic*, i.e., the result that the logical theory $Th(\mathbb{N}, +, 0, 1, <)$ is decidable

Similarly, so is our extension of the arithmetic to deal with positions of k-automatic sequences.

Least Periods

Definition

The factor u is said to be a *period* of w if $w = uu \cdots uu'$ where u' is a prefix of u.
We say u is the *least period of* w if u is the shortest such factor of w.

- For example, `alfalfa` has period 3 and `entanglement` has period 9.
- The factors of a *periodic infinite word* such as $(012)^\omega = 012012012012\cdots$ only have one shortest period, in this case 3.

Least Periods

- Given an infinite word $\mathbf{x}$, we are interested in the set of integers that are the least period of some factor w of $\mathbf{x}$.
- The set of least periods of a k-automatic word is itself k-automatic.
- Specifically, the *characteristic sequence* of the set of least periods is k-automatic.
- (For example, the characteristic sequence of the even integers is $(01)^\omega = 010101010 \cdots$)

Least Periods Query

- First, the predicate P that n is a period of the factor $\mathbf{x}[i..j]$:

$$\begin{aligned} P(n,i,j) \quad \text{means} \quad & \mathbf{x}[i..j-n] = \mathbf{x}[i+n..j] \\ = \quad & \forall\ t \text{ with } i \le t \le j-n \text{ we have } \mathbf{x}[t] = \mathbf{x}[t+n]. \end{aligned}$$

- Using this, we express LP that n is the least period of $\mathbf{x}[i..j]$:

$$LP(n,i,j) = P(n,i,j) \wedge \forall n' < n\ \neg P(n',i,j).$$

Least Periods Query

- Finally, we express the predicate that n is a least period:

$$L(n) = \exists i, j : (j \geq 0) \wedge (0 \leq i + n \leq j - 1) \wedge LP(n, i, j).$$

- In the Thue-Morse sequence, the set of least periods includes every positive integer.
- For example, the factor `1010` starting at position 2 has least period 2 and the factor `011` starting at position 0 has least period 3.
- The same is true for the Rudin-Shapiro sequence.

Powers

- A word w is called a *square* if it's of the form $w = uu$
- A word w of the form $w = uuu$ is called a *cube*.
- The exponent need not be integer; a word is $\frac{a}{b}$*-power* if w has period p and
$$\frac{|w|}{|p|} = \frac{a}{b}.$$
- For example, the English word `ionization` is a $\frac{10}{7}$-power.
- A word is called *square-free* if none of its factors are squares.
- Similarly, a word is $\frac{a}{b}$*-power free* if none of its factors are $\frac{a}{b}$*-powers*.

Leech Word

- It is well known that the Thue-Morse word avoids cubes,
- and that only *square-free* words over 2 letters are $\epsilon, 0, 1, 01, 10, 010,$ and 101.

In 1957 John Leech found an infinite *square-free* word over 3 letters. It happens to be 13-automatic.

The Leech word is defined by the following morphism:

$$0 \Rightarrow 0121021201210$$
$$1 \Rightarrow 1202102012021$$
$$2 \Rightarrow 2010210120102$$

Leech Word 15/8+

But is square-free the best we can do?

Theorem

The Leech sequence is $\frac{15}{8}^+$*-free, and this exponent is optimal.*

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

The exponent is optimal because, for example, the factor $\mathbf{l}[25..39] = 120102101201021$ is easily seen to be a $\frac{15}{8}$ power.

- We verified that there are no powers $> \frac{15}{8}$.

$$\exists p : (15p < 8n) \wedge (\exists i, j :(i + n - 1 = j) \wedge P(p, i, j))$$

- (This took 9 minutes to compute.)
- We also computed the pairs (i, n) for which a $\frac{15}{8}$ power of length n begins at position i.
- The set of all accepting paths can be represented as: $[*, 0]^*\{[1, 1], [9, 1]\}[12, 2][0, 0]^*$,
- This corresponds to lengths of the form $15 \cdot 13^i$.
- (This took 19 minutes to compute.)

Condensation

- The *appearance* and *recurrence* are well-studied properties of infinite words.
- The *appearance function* gives the size of the smallest *prefix* ‘window’ of a word such that every factor of length n is contained in the window.
- The *recurrence function* gives the size of the smallest ‘window’ *starting anywhere* of a word such that every factor of length n is contained in the window.
- The *condensation function* gives the size of the smallest ‘window’ *at some starting point* of a word such that every factor of length n is contained in the window.

Condensation examples

Formally, the **condensation function** $C(n)$ of a word is the smallest integer m such that there exists a factor of the word of length m that contains all the factors of length n.

Here is the *Thue-Morse* sequence:

0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0	1	0	0	1	0	1	1	$\cdots$	

Here the *condensation function* for *Thue-Morse* evaluates to at most 5 for $n = 2$.
(In fact it is exactly 5.)

Condensation query

We can create a machine that accepts pairs $[n, m]$ such that $m = C(n)$ for any particular k-automatic sequence:

- For a k-automatic sequence $\mathbf{x}$, we evaluate the following expression:

$$
\begin{aligned}
[n, m] = [n, \min(m : \forall k\, (\exists j\, (\exists l\, (x[i + l \ldots i + l + n - 1] \\
= x[i + j \ldots i + j + n - 1] \\
\wedge (m + k \geq n + l) \\
\wedge (l \geq k)))))]
\end{aligned}
$$

Condensation: Thue-Morse

Theorem

For the Thue-Morse sequence, we have

$$C_{\mathbf{t}}(n) = \begin{cases} 2, & \text{if } n = 1; \\ 5, & \text{if } n = 2; \\ 2^{t+1} + 2n - 2, & \text{if } n \geq 3 \text{ and } t = \lceil \log_2(n-1) \rceil. \end{cases}$$

This result was computed in in 2.959 s.

Condensation: Rudin-Shapiro

Theorem

For the Rudin-Shapiro sequence, we have

$$C_{\mathbf{r}}(n) = \begin{cases} 2, & \textit{if } n = 1; \\ 6, & \textit{if } n = 2; \\ 10, & \textit{if } n = 3; \\ 36, & \textit{if } n = 4; \\ 38, & \textit{if } n = 5; \\ 70, & \textit{if } n = 6; \\ 75, & \textit{if } n = 7; \\ 2^{t+3} + 2n - 2, & \textit{if } n \geq 8 \textit{ and } t = \lceil \log_2(n-1) \rceil. \end{cases}$$

This result was computed in 59.208 s.

Recurrence

The **recurrence quotient** Q is $\sup_{n\to\infty} R(n)/n$; it could be infinite.

- For the Rudin-Shapiro sequence, Allouche and Bousquet-Mélou gave the estimate $R_{\mathbf{r}}(n+1) < 172n$ for $n \geq 1$. (in other words: $Q_{\mathbf{r}} < 172$)
- We computed a new explicit expression for the recurrence function $R_{\mathbf{r}}(n)$ and recurrence quotient for the Rudin-Shapiro sequence $\mathbf{r}$.

Recurrence

Theorem

Let $\mathbf{r} = (r(n))_{n\geq 0}$ *be the Rudin-Shapiro sequence. Then*

$$R_{\mathbf{r}}(n) = \begin{cases} 5, & \text{if } n = 1; \\ 19, & \text{if } n = 2; \\ 25, & \text{if } n = 3; \\ 20 \cdot 2^t + n - 1, & \text{if } n \geq 4 \text{ and } t = \lceil \log_2(n-1) \rceil. \end{cases}$$

Furthermore, the recurrence quotient

$$\sup_{n\geq 1} \frac{R_{\mathbf{r}}(n)}{n}$$

is equal to 41*; it is not attained.*

Recurrence

Proof.

We created a DFA to accept

$$\{(m,n)_2 \;:\; (m - 20\cdot 2^t - n + 1, n) \;:\; n \geq 4 \text{ and } m = R(n) \text{ and } t = \lceil \log_2(n-1) \rceil\}.$$

We then verified that the resulting DFA accepted exactly pairs of the form $(0,n)_2$ for $n \geq 4$.

The local maximum of the **recurrence quotient** is evidently achieved when $n = 2^r + 2$ for some $r \geq 1$; here it is equal to $(41 \cdot 2^r + 2)/(2^r + 2)$.

As $r \to \infty$, this approaches 41 from below.

□

computed in 77.2 s

Conclusion

- We have a feasible implementation of the first order theory on k-automatic sequences.
- We can express and evaluate many commonly sought properties these words.
- We improve hand-made approximations.
- We propose a *condensation function* and describe it.
- We show that the set of least periods of a k-automatic sequence is also k-automatic (in some representation.)
- Thank you!