Decidable Properties of Automatic Sequences

Narad Rampersad

Department of Mathematics and Statistics
University of Winnipeg

Goal: exploit the decidability of certain logical theories to provide entirely computer generated proofs of certain results in combinatorics on words.

- The first-order theory of $\langle \mathbb{N}, +, \times, = \rangle$ is undecidable (Tarski and Mostowski 1949).
- The first-order theory of $\langle \mathbb{R}, +, \times, = \rangle$ is decidable (Tarski 1949).
- The first-order theory of $\mathfrak{N}_A = \langle \mathbb{N}, +, = \rangle$ is decidable (Presburger 1929).

- We call the theory of $\mathfrak{N}_A$ Presburger arithmetic.
- Presburger's proof used elimination of quantifiers.
- The decidability of Presburger arithmetic can also be proved using automata.
- Stronger result: we can prove decidability of certain extensions of Presburger arithmetic.

- A corollary of Presburger's proof is that $S \subseteq \mathbb{N}$ is definable in Presburger arithmetic if and only if S is a finite union of arithmetic progressions.
- Recall: A set $S \subseteq \mathbb{N}^d$ is definable in $\mathfrak{N}_A$ if there is a formula ϕ with d free variables such that

$$S = \{(n_1, \ldots, n_d) \in \mathbb{N} : \mathfrak{N}_A \models \phi(n_1, \ldots, n_d)\}.$$

- Let's extend Presburger arithmetic as follows.
- Let $V_k(x)$ denote the largest power of k that divides x.
- e.g., $V_2(80) = 16$.
- by convention $V_k(0) = 1$
- consider the structure $\mathfrak{N}_k = \langle \mathbb{N}, +, =, V_k \rangle$

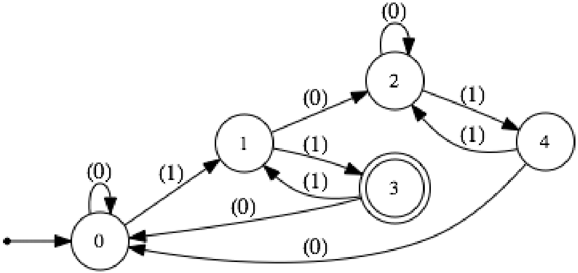
Theorem (Bruyère 1985)

The first order theory of $\mathfrak{N}_k = \langle \mathbb{N}, +, =, V_k \rangle$ is decidable.

- Recall that the only subsets of $\mathbb{N}$ definable in $\mathfrak{N}_A$ are finite unions of arithmetic progressions.
- $\mathfrak{N}_k$ is richer
- e.g., we can define the powers of 2 in $\mathfrak{N}_2$ by the formula

$$x = V_2(x).$$

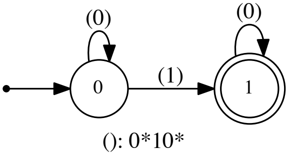
- We now consider a completely different way to define subsets of $\mathbb{N}$.



informally, a finite automaton is a directed, edge-labelled multigraph

- we call the vertices states and the edges transitions
- for us each state will have k outgoing transitions labeled with the digits $0, 1, \ldots, k-1$ (this is the alphabet)
- there is an initial state and a set of final states
- an automaton accepts a string of digits $b_0 b_1 \cdots b_{m-1}$ if there is a path of length m from the initial state to a final state such that for $i = 0, \ldots, m-1$, the i-th transition in the path is labeled b_i

An automaton for the powers of 2



(Transitions not shown go to an implied "sink" state.)

A subset $S \subseteq \mathbb{N}$ is k-automatic if there is some finite automaton that accepts exactly the base-k representations of elements of S.

- How do we define relations?
- To recognize an element $(x, y) \in \mathbb{N} \times \mathbb{N}$ we extend the alphabet of the automaton to $\{0, \ldots, k-1\} \times \{0, \ldots, k-1\}$.
- Then (x, y) is in the set defined by the automaton if the automaton accepts

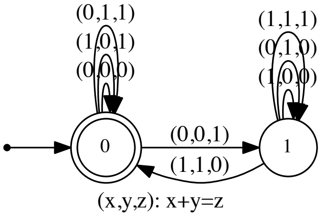
$$(a_0, b_0)(a_1, b_1) \cdots (a_{m-1}, b_{m-1}),$$

where $a_0 a_1 \cdots a_{m-1}$ and $b_0 b_1 \cdots b_{m-1}$ are the base-k representations for x and y respectively (possibly padded with leading zeros).

For example, we represent $(11, 5, 16) \in \mathbb{N} \times \mathbb{N} \times \mathbb{N}$ in binary by

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix},$$

where we have written triples as columns.



This automaton defines the relation $x + y = z$ in base 2.

Base-k addition is definable using automata!

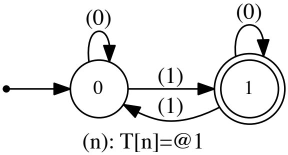
Let S, T be k-automatic sets. There are standard constructions to obtain automata that define

$$S \cup T, \quad S \cap T, \quad \overline{S}.$$

Theorem (Büchi–Bruyère)

A set $S \subseteq \mathbb{N}^d$ is k-automatic if and only if it is definable in $\mathfrak{N}_k$.

- Idea for $\Leftarrow$: logical operations $\vee, \wedge, \neg$ correspond to $\cup, \cap, \overline{}$.
- Quantifiers $\exists, \forall$ are trickier (we use non-determinism).
- Addition is done as shown previously.
- Proof is by structural induction on the formula.
- Let's do an example of $\Rightarrow$.



This automaton accepts any number whose binary representation contains an odd number of 1's.

- let's try to define the same set in $\mathfrak{N}_2$
- consider some number n written in binary
- let m be the number obtained from the binary representation of n by turning every second 1 into a 0 (say, from right to left)
- e.g.,

$$\begin{array}{rcccccccccccc} (n)_2 & = & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ (m)_2 & = & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}$$

Then n has an odd number of 1's if and only if there exists m satisfying:

- the smallest powers of 2 appearing in $(m)_2$ and $(n)_2$ are equal
- the largest powers of 2 appearing in $(m)_2$ and $(n)_2$ are equal
- for every pair of consecutive powers of 2 occurring in $(n)_2$, one occurs in $(m)_2$ and the other does not
- minor technicality: What if n or m is 0?

- the relations $\leq$ and $<$, as well as any given constant, can be defined in $\langle \mathbb{N}, +, =, V_2 \rangle$
- when building our formula in $\mathfrak{N}_2$ we use these symbols as shortcuts for their defining formulas

- Checking the smallest powers of 2 is easy: $V_2(n) = V_2(m)$
- Let $\lambda_2(x) = y$ denote the largest power of 2 appearing in $(x)_2$ (by convention $\lambda_2(0) = 1$).
- Then $\lambda_2(x) = y$ is defined by

$$
\begin{aligned}
&[(V_2(y) = y) \wedge (y \leq x) \\
&\quad \wedge ((\forall z)((V_2(z) = z) \wedge (y < z)) \rightarrow (x < z))] \\
\vee&[(x = 0) \wedge (y = 1)]
\end{aligned}
$$

- Checking the largest powers of 2 becomes $\lambda_2(n) = \lambda_2(m)$.
- To check the "internal 1's" we start by defining a predicate $\phi_2(x, y)$ which indicates that y is a power of 2 occurring in the binary expansion of x.

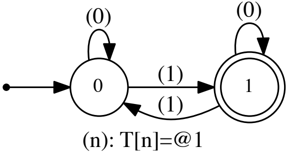
- $\phi_2(x, y)$ is defined by

$$(V_2(y) = y) \wedge [(\exists z)(\exists t)(x = z + y + t) \wedge (z < y) \\ \wedge ((y < V_2(t)) \vee (t = 0))]$$

- Using this we can verify the last condition (we omit the details).
- Summary: the property that $(n)_2$ has an odd number of 1's can be defined in $\mathfrak{N}_2$.

- This was an example of automaton $\Rightarrow$ formula conversion.
- To show decidability of $\mathfrak{N}_k$ we convert from formula $\Rightarrow$ automaton.
- Determining if the set defined by an automaton is empty is decidable.
- We can decide if a given formula is satisfiable in $\mathfrak{N}_k$ by building an automaton accepting all satisfying assignments and then checking if this set is non-empty.
- Hence, the theory $\mathfrak{N}_k$ is decidable.
- A fortiori, we see that $\mathfrak{N}_A$ is decidable.

Recall: goal was to apply the decidability of $\mathfrak{N}_k$ to prove combinatorial properties of certain sequences.



Let

$$\mathbf{t} = 0110100110010110\cdots$$

be the sequence with a 1 in position n exactly when the above automaton accepts $(n)_2$.

- The Thue–Morse sequence

$$\mathbf{t} = 0110100110010110\cdots$$

  has the remarkable combinatorial property that it does not ever contain a repetition of the same block X three times in succession (i.e., XXX).
- Is there an algorithm that can verify for any given k-automatic sequence if the sequence has this property?
- If the property can be expressed in $\mathfrak{N}_k$, then by the earlier decidability result, the answer is yes.

Theorem (Charlier, R., Shallit 2011)

If we can express a property of a k-automatic sequence $\mathbf{x}$ using quantifiers, logical operations, integer variables, the operations of addition, subtraction, indexing into $\mathbf{x}$, and comparison of integers or elements of $\mathbf{x}$, then this property is decidable.

- A sequence $\mathbf{a}$ contains an occurrence of the pattern XXX if and only if there exist integers $p \geq 1$ and $0 \leq m_1 < m_2 < m_3$ such that $\mathbf{a}(m_1 + i) = \mathbf{a}(m_2 + i) = \mathbf{a}(m_3 + i)$ for all $0 \leq i < p$.
- If $\mathbf{a}$ is k-automatic then this property can be defined in $\mathfrak{N}_k$.
- Hence there is an algorithm to decide if a given k-automatic sequence avoids XXX.

- Hamoon Mousavi has implemented this method as a Java application called `Walnut`.
- `Walnut` can be found on Jeffrey Shallit's webpage `https://cs.uwaterloo.ca/~shallit/papers.html` along with many examples of applications of the method.

- previously proving results like this involved an ad hoc argument for each situation
- this method allows for quick, routine verifications of properties of automatic sequences in a wide variety of contexts
- time complexity: theoretically the worst case is a tower of exponentials as high as the number of quantifier alternations in the formula
- in practice, runs quickly if formula not too complicated.

The End