Automata and Numeration Systems

Authors: Jack Gentile, Erik Joan Hernandez, Dun "Eric" Ma, Stephen O'Brien, Haozhe "Howard" Wang
Graduate Mentors: Eion Blanchard and Alexi Block Gorman
Faculty Advisors: Philipp Hieronymi and Erik Walsberg

Introduction

Automata

Definition. *A* ***nondeterministic finite automaton****, or briefly an nfa, over alphabet* Σ *is a quadruple* $A = (S, I, T, F)$*, where*

- S *is a finite nonempty set called the set of* ***states****.*
- I *is a subset of* S *called the set of* ***initial states****.*
- $T \subset S \times \Sigma \times S$ *is a nonempty set called the* ***transition table*** *or* ***transition diagram****.*
- F *is a subset of* S *called the set of* ***final states****.*

Definition. *A* ***run*** *of A is a sequence* $s_1 ... s_{n+1}$ *on* $u = \delta_1 ... \delta_n$ *so that* $s_1 \in I$ *and* $(s_i, \delta_i, s_{i+1}) \in T$*.*
Definition. *An input* $u = \delta_1 ... \delta_n$ *is* ***accepted*** *by* A *if the last state of the run of* A *on* u *is in* F*.*

Numeration Systems

Definition. *A* ***numeration system*** *is a method used to represent numbers in which each digit represents a unique base value. Some examples of common systems are binary, decimal, and Ostrowski numerations.*
Definition. *An* ***Ostrowski-***α ***numeration*** *is a numeration system where the base values are calculated from the continued fraction of* α*, denoted* $[a_0; a_1, a_2, ...]$*.*
The base values q_n *of an Ostrowski numeration are defined recursively by*

$$q_n = a_n q_{n-1} + q_{n-2} \text{ when } n \geq 2, \text{ where } q_1 = a_1 \text{ and } q_0 = 1.$$

If a number x *written as* $b_n \dots b_3 b_2 b_1$ *is in Ostrowski numeration, then*

- ***constraint 1.*** *For all* $n \geq 2$*,* $b_n \leq a_n$*, and* $b_1 < a_1$
- ***constraint 2.*** *For all* $n \geq 2$*, if* $b_n = a_n$*, then* $b_{n-1} = 0$*.*

For example, when $\alpha = \sqrt{2} = [1; 2, 2, 2 \dots]$, the base values starting with $q_0, q_1, q_2, \dots$ are $1, 2, 5, 12, 29, 70, 169, \dots$.
112100 violates constraint 2 by having a 2 followed by a 1, 112030 violates constraint 1 by having a 3 on the fourth digit. Thus, 112100 and 112030 are not valid numbers in Ostrowski-$\sqrt{2}$ representation.
112020 would be a valid number in Ostrowski-$\sqrt{2}$ representation and would have value equal to 127 in base 10.

Walnut Software

Walnut is a theorem-prover software developed by Hamoon Mousavi in 2016. It takes any *first-order logic with addition and comparison* then outputs an automaton associated with it. In addition, Walnut can use any numeration system, as long as the following three automata are provided:

- A *recognition automaton* that only accepts valid numbers in that numeration system.
- An *addition automaton* that only accepts triples (a, b, c) such that $a + b = c$.
- A *comparison automaton* that only accepts pairs (a, b) such that $a < b$.

For the Ostrowski-$\sqrt{2}$ numeration system, the comparison automaton is auto-generated, and one can check that the automaton in *Figure 1* accepts input that meets constraints 1 and 2. We managed to produce the addition automaton as described in the next section.

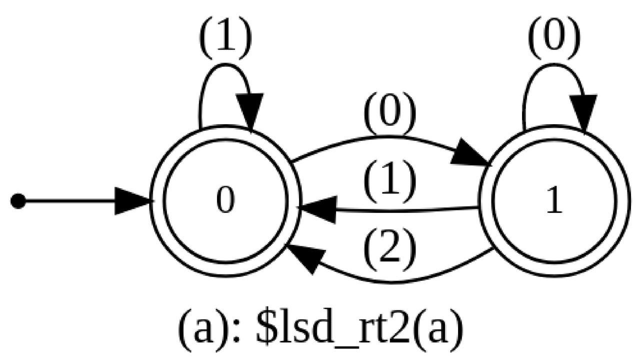

Figure 1: *Ostrowski-*$\sqrt{2}$ *Recognition*

The Addition Algorithm and Corresponding Automaton

An algorithm for addition in Ostrowski numeration systems contains four sub-algorithms. We translated each sub-algorithm into a corresponding automaton which exactly accepts the pairs (x, y) where the algorithm takes in x and gives result y.

- Algorithm 0 simply adds the operands.
- Algorithm 1 checks constraint 1.
- Algorithms 2 and 3 check constraint 2.

By running the following command in Walnut, we combine our automata for these sub-algorithms, which is shown in *Figure 3*:

$$\exists y, x, w \ \text{alg}_0(a, b, w) \ \& \ \text{alg}_1(w, x) \ \& \ \text{alg}_2(x, y) \ \& \ \text{alg}_3(y, c)$$

Automaton for Algorithm 1

Definition. *alg*$_1(\alpha = [a_0; a_1, a_2, \dots])$ *is a nondeterministic finite automaton* $\{S, I, T, F\}$ *over* Σ *where:*

- ***alphabet*** $\Sigma = \{(s, t) : 0 \leq s \leq 2\,max(a_i), 0 \leq t \leq max(a_i)\}$.
- *set of states* $S = \left\{ \left(\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}, g, i \right) \right\}$*. The matrix represents the input,* g *is a carry number that is 0 or 1, and* i *represents the position of* c *and* f *in the continued fraction.*
- *set of initial states I =* $\left\{ \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, 0, i \right) \right\}$ *for all integers* $i > 0$*.*
- *set of final states F =* $\left\{ \left(\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}, 0, 0 \right) : B(a, b, c) = (d, e, f) \right\}$ *where* B *does not change the represented values between* (a, b, c) *and* (d, e, f) *while the latter satisfies constraint 1.*
- ***transition table*** *:* $\left(\left(\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}, g, i \right), \begin{pmatrix} s \\ t \end{pmatrix}, \left(\begin{bmatrix} b' & c' & s' \\ e & f & t \end{bmatrix}, g', i - 1 \right) \right) \in T$ *if* $(a + g, b, c, s, g)$ *and* (d, b', c', s') *represent the same value, and the latter satisfies constraint 1.*

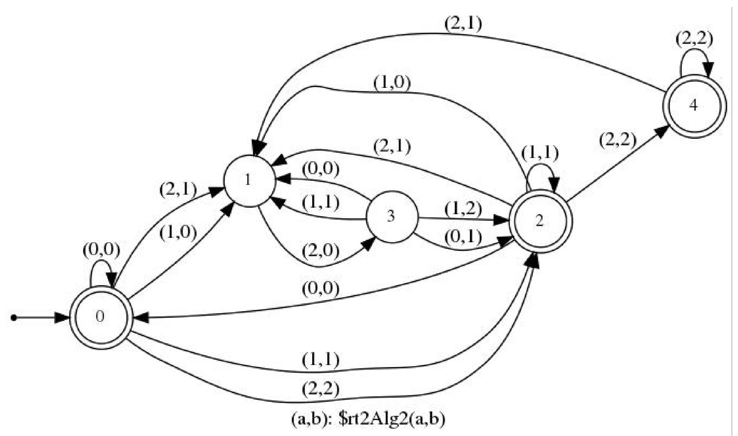

Figure 2: *Algorithm 2 of Ostrowski-*$\sqrt{2}$ *(least significant digit first)*

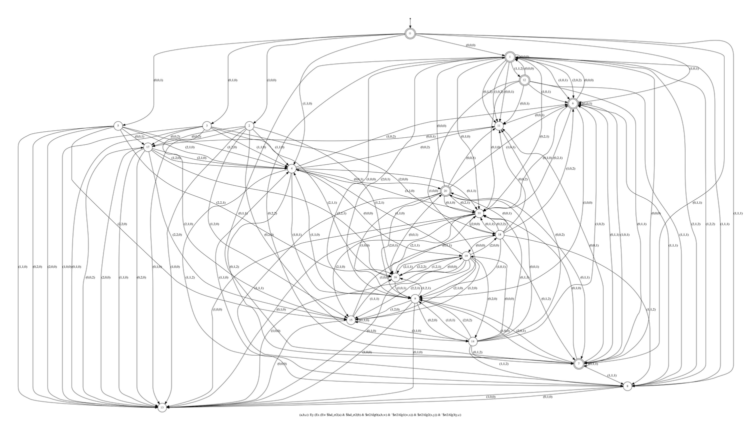

Figure 3: *Ostrowski-*$\sqrt{2}$ *Addition*

Example of addition algorithm in Ostrowski-$\sqrt{2}$

We want to add 103_{10} and 132_{10} in Ostrowski numeration with $\alpha = \sqrt{2} = [1; 2, 2, 2, \dots]$.

1. Convert into Ostrowski representation:
   $109_{10} = 0 \times 1_{10} + 0 \times 2_{10} + 2 \times 5_{10} + 0 \times 12_{10} + 1 \times 29_{10} + 1 \times 70_{10} = 0110200_{\sqrt{2}}$
   $138_{10} = 0 \times 1_{10} + 0 \times 2_{10} + 2 \times 5_{10} + 0 \times 12_{10} + 2 \times 29_{10} + 1 \times 70_{10} = 0120200_{\sqrt{2}}$
2. Run Algorithm 0 (add operands): $0110200 + 0120200 \Rightarrow 0230400$.
3. Run Algorithm 1 (check constraint 1): $0230400 \Rightarrow 1020400 \Rightarrow 1021111$.
4. Run Algorithms 2 and 3 (check constraint 2): $1021111 \Rightarrow 1100111$. (Try this on *Figure 2*.)

Therefore, $0110200_{\sqrt{2}} + 0120200_{\sqrt{2}} = 1100111_{\sqrt{2}}$.
Verify: $1100111_{\sqrt{2}} = 1 \times 1_{10} + 1 \times 2_{10} + 1 \times 5_{10} + 1 \times 70_{10} + 1 \times 169_{10}$ = $247_{10} = 109_{10} + 138_{10}$.

Propositions Evaluated in Walnut

With the definition of the automaton for addition in Ostrowski numeration systems, we were able to construct similar proofs as those in Du, Mousavi, Schaeffer, and Shallit's paper "Decision Algorithms for Fibonacci-Automatic Words, with Applications to Pattern Avoidance" for the **characteristic Sturmian word with slope** $\sqrt{2}$ instead of for the original Fibonacci word.

Definition. *The* ***characteristic Sturmian word with slope*** $\sqrt{2}$*, which we denote as* C_2*, is the infinite word obtained as the limit of the sequence of* ***standard words*** s_n *defined by*

$$s_n = s_{n-1}^{d_n} s_{n-2} \text{ when } n \geq 2, \text{ where } s_1 = 0^{d_1 - 1}1 \text{ and } s_0 = 0.$$

By having built the Ostrowski-$\sqrt{2}$ numeration system and the automatic word C_2 in Walnut, we may use the command $C2[i]$ to return the i^{th} digit of C_2.

Example theorem

Theorem: C_2 *contains fourth powers.*

Proof: We create the following predicate that accepts the length n of fourth powers in C_2

$$(n > 0) \wedge \exists i \ \forall t < 3n \ C_2[i + t] = C_2[i + n + t]$$

We then translate the predicate into one that Walnut recognizes:

$$?\text{msd\_rt2}((n > 0) \ \& \ (Ei \ At(t < 3 * n => C2[i + t] = C2[i + n + t])))$$

For this predicate, Walnut generates the following automaton:

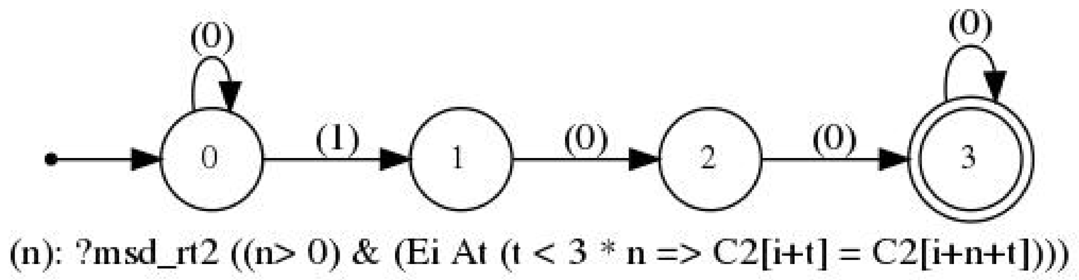

Figure 4: *Automaton for the Theorem*

which means that there are fourth powers of length $100_{\sqrt{2}}, 1000_{\sqrt{2}}, 10000_{\sqrt{2}}, \dots$.

Future Work

- Use the automata for Ostrowski numeration systems to prove more theorems regarding characteristic Sturmian words.
- Investigate the critical exponent of infinite balanced words.
- Find a faster algorithm to generate automata for Ostrowski addition.
- Provide a general automaton for Ostrowski numeration, which would include α as input.

References

[1] Khoussainov, Bakhadyr.Nerode, Anil. (2001) *Automata Theory and its Applications*, MA : Birkhäuser Boston [2] Rampersad, Narad. Shallit, Jeffrey and Vandomme, Elise. (2018) *Critical exponents of infinite balanced words*, Canada
[3] Du, Chen Fei. Mousavi, Hammoon. Schaeffer, Luke and Shallit, Jeffrey. (2014) *Decision Algorithms for Fibonacci-Automatic Words, with Applications to Pattern Avoidance*
[4] Hieronymi, P., & Terry Jr, A. (2018). *Ostrowski Numeration Systems, Addition, and Finite Automata.* Notre Dame Journal of Formal Logic, 59(2), 215-232.