Pecan: An Automated Theorem Prover

Authors: Zhengyao Lin, Dun Ma, Reed Oei, Yikai Teng, Pavle Vuksanovic Advisors: Christian Schulz, Mary-Angelica Tursi, Prof. Philipp Hieronymi

Introduction

Pecan is an automated theorem prover for reasoning about *automatic sequences*, which are sequences that can be recognized by some (typically finite) automaton. Pecan is capable of proving any statement expressed in terms of Büchi automata and first-order logic connectives.

Pecan programs are made up of *predicates* and *directives*:

- predicates: a linear temporal logic formula, a first order logic formula with equality, or loading an existing automaton.

```
y is successor_of(x) := x < y ∧ ∀z. z <= x ∨ y <= z
```

- directives: commands to the Pecan interpreter, such as: `Theorem`, which asks Pecan to prove a theorem, or `save_aut`, which asks Pecan to build the automaton corresponding to some predicate and save it to a file.

```
Theorem ("Addition is commutative.", { ∀x,y. x + y = y + x }).
```

We have used Pecan to prove many theorems about *Sturmian words*, and we are currently exploring extensions including deciding sentences involving linear inequalities with integer and quadratic irrational coefficients, and visualization of fractals.

You can try out Pecan online at `http://reedoei.com/pecan`.

Characteristic Sturmian Words

A *cutting sequence* for a curve is a sequence of 0's and 1's, corresponding to when the line crosses vertical and horizontal grid lines, respectively. The *characteristic Sturmian word with slope* α is an infinite binary sequence defined by the cutting sequence of $y = \alpha x$ for some irrational $\alpha \in (0, 1)$ in the Cartesian plane. Figure 2 shows the beginning of the characteristic Sturmian word with slope $\frac{1}{\phi}$, which begins $0100101001\ldots$.

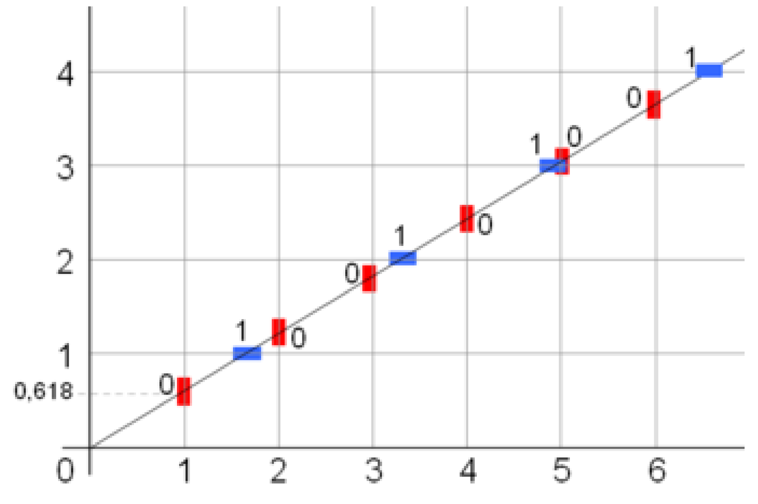


Figure 2: Characteristic Sturmian word with slope $\frac{1}{\phi}$

Sturmian words are automatic sequences: there are automata which calculate their n-th digit given the representation of n in the appropriate numeration system. For Sturmian words, we use a family of numeration systems called *Ostrowski numeration systems*. With the addition automaton for general Ostrowski numeration systems, we are able to use Pecan to *automatically prove* properties about Sturmian words.

Theorems about Sturmian Words

We can use Pecan to prove many interesting properties of Sturmian words: one fundamental result is that Sturmian words are not *eventually periodic*.

Definition. *A word is* eventually periodic *if it is of the form* $abbbbb\ldots$ *for some subwords* a *and* b *(e.g.,* $0.10245454545\ldots$ *where the repeating part is* 45*).*

Theorem. *Sturmian words are not eventually periodic.*

Proof. In Pecan, prove the statement by writing the definition of "eventually periodic" and stating the theorem. Running the Pecan program below proves the theorem.

```
eventually_periodic(a, p) :=
    p > 0 ∧ ∃n. ∀i. if i > n then C[i] = C[i+p]

Theorem ("Sturmian words are not eventually periodic", {
    ∀a,p. if p > 0 then ¬eventually_periodic(a,p)
}).
```

We omit the pictures of the intermediate automata, as they have hundreds (or even thousands) of states, and so it is nearly impossible to understand them by looking at pictures of them. □

In this example, we state and prove a theorem about **all** Sturmian words.

- Previous theorem provers (e.g., Walnut [2]) in the same area could only prove theorems about a single Sturmian word, or small subsets of Sturmian words.

Using Pecan, we proved many other theorems about Sturmian words, including many classical results, some recent results, and notably, some **new** results.

Multiplying Ostrowski Representations

In the Ostrowski numeration systems mentioned previously, every natural number has a unique α*-Ostrowski representation*.

Specifically, for any irrational α with an infinite continued fraction expansion, there is a unique way to represent any natural number as a sum of products of the denominators of the consecutive continued fraction approximations of α.

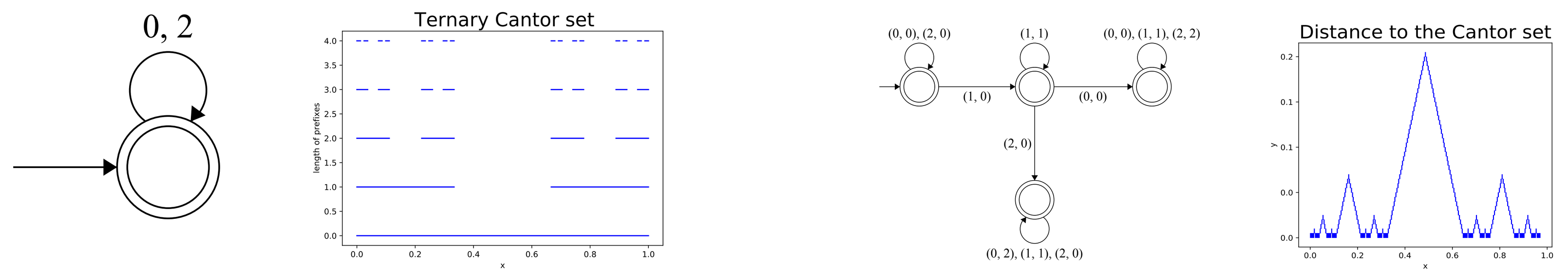
We implemented an extension to Pecan to construct an automaton recognizing order in the ring $\mathbb{Z}[\alpha]$ using α-Ostrowski representations, where α is a quadratic irrational using a method developed by Hieronymi et al. [4]. Unfortunately, the size of this automaton scales quickly with the complexity of the continued fraction expansion of α.

- For $\alpha = 1/\phi = (\sqrt{5} - 1)/2$, with continued fraction expansion $[0; 1, 1, \ldots]$, the automaton for recognition of order in $\mathbb{Z}[\alpha]$ had 2,134 states.
- For $\alpha = \sqrt{2} - 1$, with continued fraction expansion $[0; 2, 2, \ldots]$, the automaton for recognition of order in $\mathbb{Z}[\alpha]$ had 218,072 states.
- For more complex α, we are unable to compute the automata due to their large size.

Fractals

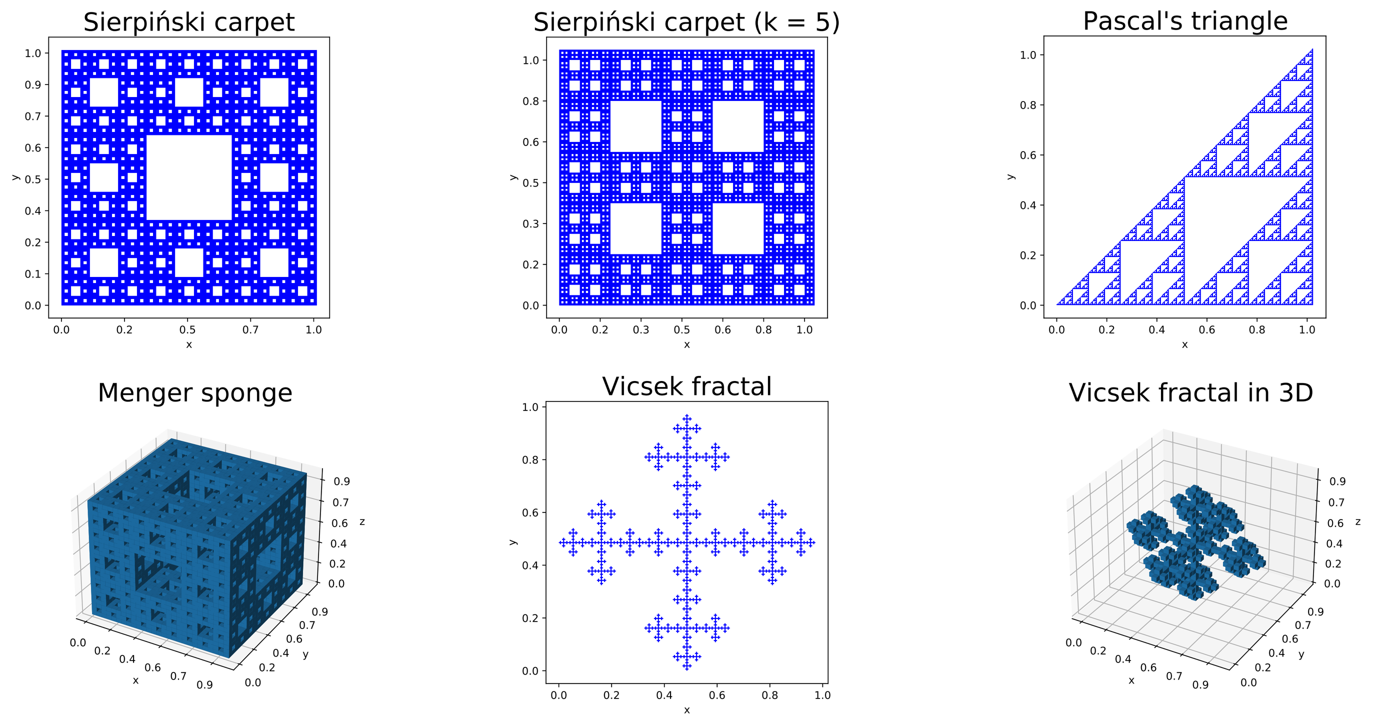
Automatic fractals are fractals recognizable by an automaton, via suitable mappings between words and reals. We developed an extension to Pecan that maps the set of words accepted by an automaton to a subset of $[0, 1]^n$ and used it to plot some fractals.

- Automata and plot of the Cantor set and Cantor distance function:

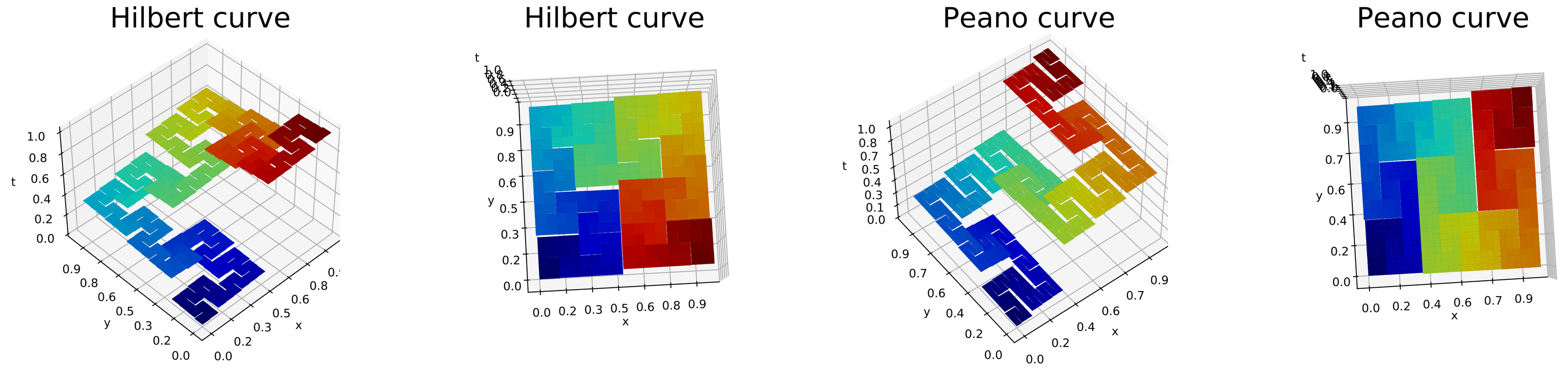


The automaton for the distance function to the Cantor set comes from [1].

- Using a similar automata to the Cantor set, we have the following:



- By adding a time variable to the automata, we can draw space-filling curves like the Hilbert curve and Peano curves:



References

[1] Alexi Block Gorman, Philipp Hieronymi, Elliot Kaplan, Ruoyu Meng, Erik Walsberg, Zihe Wang, Ziqin Xiong, and Hongru Yang. Continuous regular functions. 2019.

[2] Khoussainov, Bakhadyr and Nerode, Anil. (2001) *Automata Theory and its Applications*, MA : Birkhäuser Boston

[3] H. Mousavi. *Automatic Theorem Proving in Walnut.* In: CoRR abs/1603.06017 (2016).

[4] P. Hieronymi, D. Nguyen, I. Pak. (2019) *Presburger Arithmetic with algebraic scalar mutliplications.* arXiv:1805.03624.

Support for this project was provided by the Illinois Geometry Lab and the Department of Mathematics at the University of Illinois at Urbana-Champaign. This project was partially supported by NSF grant DMS-1654725. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.