Department of Mathematics and Statistics

Gandhar Joshi, 2nd year PhD student
PI: Z574709X Email: gandhar.joshi@open.ac.uk
Supervisors: Dr. Dan Rust, Dr. Reem Yassawi, Dr. Ian Short
EPSRC grant number: EP/V007459/1; OU funded

How'd you like your ~~Eggs~~ Numbers, sunshine?

(Representations of Integers)

Introduction, Motivation and Objective

We usually think of numbers as finite sequences of symbols 0 to 9, i.e. decimals. Computers interpret them using only 0's and 1's, i.e. binary. Civilisations of the past have been known to have utilised different integer bases to simplify their particular problems. Ever wondered why, in the famous 'Hitchhiker's guide' by Douglas Adams, $6 \times 9 = 42$ makes sense? (Hint: Maybe the numbers are not written in decimal!?...)

A correct choice of **representation** of numbers often simplifies the analysis of a particular problem. Therefore, our objective is to determine a **custom numeration system** for such problems.

1 Recipe of a numeration system

To represent any integer $N \geq 0$, we follow these steps:

1. Take $U := (u_n)_{n\geq 0}$, a strictly-increasing integer sequence with $u_0 = 1$.
2. If the ratio $r = \lim_{n\to\infty} u(n+1)/u(n)$ is bounded by a constant, then there is a canonical representation of N called the **greedy U-representation** of N associated with r, such that

$$N := \sum_{i\geq 0} a_i u_i \tag{1}$$

3. It is denoted as $\langle N \rangle_U = a_i \dots a_0$; where a_i takes values from a **canonical alphabet** A_U defined as:

$$A_U := \{0, 1, \dots, \lceil r - 1 \rceil\}.$$

2 Examples

1. Take U to be a geometric sequence with $r \in \mathbb{N}$. This gives us the usual base-r representation. $r = 10 \implies$ decimal; and $r = 2 \implies$ binary. Then 24 can be represented in decimal using alphabets $A_D = \{0, \dots, 9\}$ since $24 = 2 \times 10 + 4 \times 1$. Can you figure out the representation of 24 in binary using $A_B = \{0, 1\}$?

...	100	10	1
		2	4

...	32	16	8	4	2	1
		1	?	?	?	?

2. We even have $r \notin \mathbb{N}$. For example, take U to be the sequence of **Fibonacci numbers** $(F_n)_{n\geq 2}$ where $u_0 = F_2 = 1$, $F_3 = 2$ and $F_{n+1} = F_n + F_{n-1}$, so r approaches the **golden ratio** $\phi \approx 1.618\dots$. This is called the **Zeckendorf representation**, a.k.a. the **Fibonacci base**. So using $A_F = \{0, \dots, \lceil \phi - 1 \rceil\} = \{0, 1\}$, the Zeckendorf representation of 24 is $\langle 24 \rangle_F = 1000100$.

...	21	13	8	5	3	2	1
	1	0	0	0	1	0	0

3 Addition Rules

For adding numbers in integer bases, we have an elementary rule of addition with carry that we are taught in primary school.

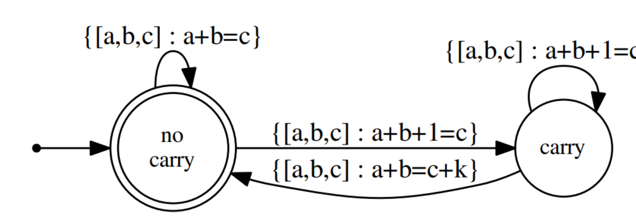
For example, adding 24 and 37 in decimal:

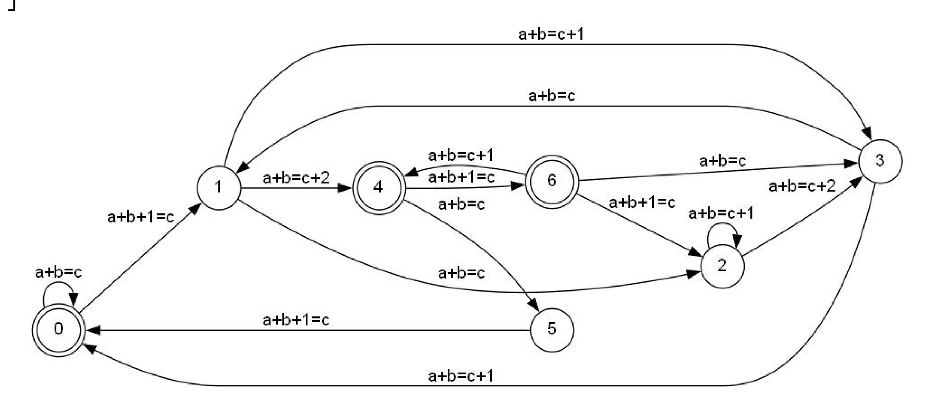
	2	4
+	3	7
carry	1	0
	6	1

But in bases like Fibonacci, the same addition rules do not apply anymore. Check that $2 + 3 = 5$ becomes $10 + 100 = 1000$ in the Fibonacci base. So we have to define custom rules to be able to perform the addition in bases like Fibonacci directly.

4 Automata Theory

A **Deterministic Finite Automaton** (DFA) is a simple model of computation that takes input and either accepts or rejects the input, described as a directed graph. [2]





It consists of **states** $[S]$ or nodes (circles) including the initial state (where the automaton starts reading the input; **edges** $[E]$ (arrows) labelled with input(s) from a finite alphabet; the **transition function** given by $t : S \times E \to S$ that determines which edge to follow from one state to another; and **accepted states** given by double concentric circles, reaching which, means the input is accepted and **rejected states**, reaching which, means the input is rejected.

The highlight of this method is that the states are finite but your input can be an arbitrarily long string and it will determine nevertheless! For addition in base k: $\mathbf{a} + \mathbf{b} = \mathbf{c}$ where $\mathbf{a} = a_0 \dots a_n$, $\mathbf{b} = b_0 \dots b_n$ and $\mathbf{c} = c_0 \dots c_n$, such that $a_i, b_i, c_i \in A_k$ (for k=10, recall $A_D = \{0, \dots, 9\}$), the DFA (Figure on the left) reads the addition in triplets $[a_0, b_0, c_0], \dots, [a_n, b_n, c_n]$ from the most significant digit. It should reach an accepting state if the addition is correct and reach a rejecting state otherwise.
For addition in Fibonacci base, the DFA (Figure on the right) has 7 states and there are other numeration systems with even more states! The following theorem makes sure that the number of states are finite, which is a crucial condition.

Theorem 1 (Frougney '91)**.** *If the ratio r is equal to a **Pisot number**, then there exists a finite adder for the numeration system U associated with r.*

5 Possible Applications and Future tasks

- A potential application would be using different number systems in **security** keys, encoding in **cryptography**, but otherwise, pure mathematics is famous for generating difficult questions that could be solved by computers and in result, improves the **computational abilities** of computers.
- Pisot substitutions are important objects in the study of dynamical systems and mathematical quasicrystals. Our goal is to generate finite adders based on them to prove statements about Pisot substitutions using the **Walnut** [1] software. The statements that can be converted to first order logic when fed to the software result in a quick True/False answer or a set of true values. This saves a lot time and space compared to the proofs done by analysis.
- The challenge here is to determine which open problems can be encoded as first order logical statements. Thankfully there are lots of open problems in the literature!

References

[1] Hamoon Mousavi. Automatic theorem proving in Walnut, 2021.

[2] Jeffrey Shallit. *The Logical Approach to Automatic Sequences: Exploring Combinatorics on Words with Walnut.* London Mathematical Society Lecture Note Series. Cambridge University Press, 2022.