Fun with FRACTRAN

Khushi Kaushik, Tommy Murphy, and David Weed

California State University Fullerton

Turing Machines

Turing Machines model the action of a computer. The main parts of a Turing machine are a way to store data, originally (abstractly) thought of as an infinitely long tape, and a set of rules that allow the conditional change of that data. The starting state of the tape is thought of as the input of the code, and the resulting state is the output. Different programs are then made by changing the rules. Conway's FRACTRAN is a very clever example of a Turing complete Programming language. The significance of this Turing machine is the entirety of the data, input, and output, and the data itself is stored in a finite set of fractions and a single integer.

The key property of numbers FRACTRAN exploits is that prime numbers are the "atoms" of integers. In fact, each integer can be uniquely decomposed in terms of its prime factors.

Theorem: (Euclid, 300BC)

Every $n \in \mathbb{N}$ admits a unique prime decomposition $n = \prod_{i=1}^{k} p_i^{\alpha_i}$.

For instance, $76 = 4.19 = 2^2.19$. Conway's simple idea is to encode a Turing machine using only fractions. Start with N. This admits a prime factor decomposition. Each power of the prime appearing in N tells us the initial state of our system: it tells us what is in each register.

Example 1: Registers

The number $24,500 = 2^2 \cdot 5^3 \cdot 7^2$ encodes three registers 2, 5, and 7, with values 2, 3, and 2, respectively.

Now multiply N by a fraction f_i so that $f_i N$ is also a whole number: if we take the prime factor decomposition of the numerator and denominator of f_i, we have that $f_i N \in \mathbb{N}$ if, and only if, the powers appearing in the prime decomposition of N have been redistributed.

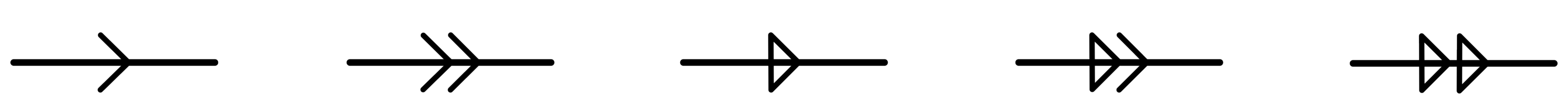
Example 2: Transferring between registers

Take $N = 24 = 2^3 \cdot 3^1 = 2$ and consider $f_1 = \frac{7}{3}$. Then $f \cdot N = 56 = 2^3 7^1$. We have transferred the 1 in the 3-register to the 7-register.

How to play FRACTRAN

To play FRACTRAN, we need an initial state (a stored number) $N \in \mathbb{N}$ which is in our *register* and a fixed list of fractions $\{f_1, f_2, \dots, f_n.\}$. Compute $f_i N$, with $i = 1, 2, \dots, n$, until we reach the first instance where $f_i N \in \mathbb{N}$. Now change the register to $f_i N$ and iterate.

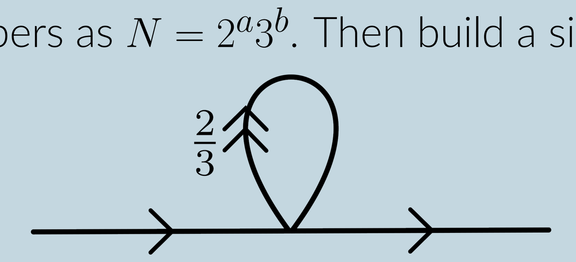
In practice, we think of the game as a flowchart that proceeds from one node (state) to another. To indicate where to go, the nodes are connected by arrows with a well-defined hierarchy. The hierarchy is as follows:



These arrows are then labeled with fractions which tell us how to multiply our registe number. There is a well-understood algorithm to convert this flowchart into a list of fractions.

Example 3: Addition

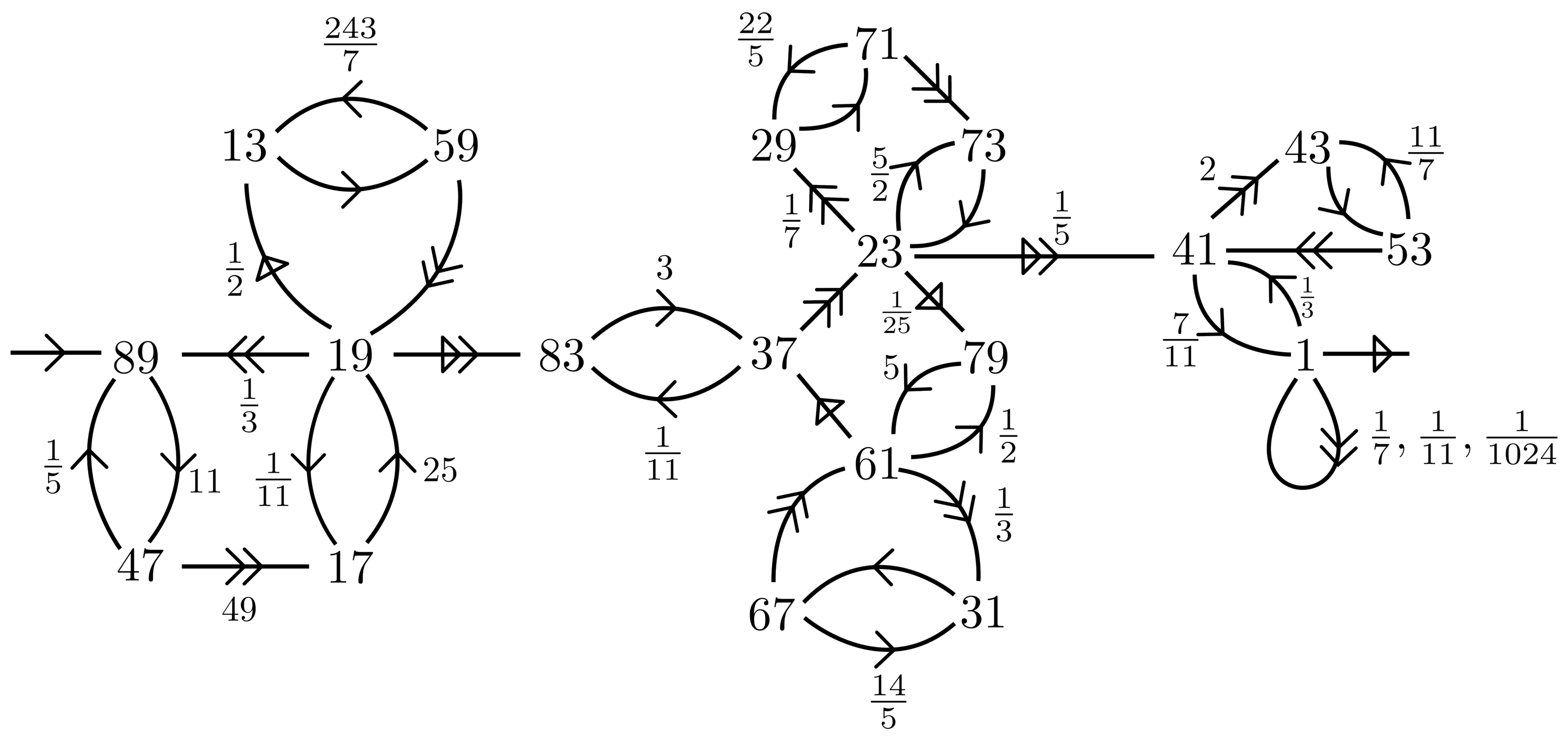
To add a and b, store the numbers as $N = 2^a 3^b$. Then build a single loop labeled with 2/3.



This visually indicated our FRACTRAN game: every time we go around the loop, we multiply N by $\frac{2}{3}$. It is easy to see this game ends with output 2^{a+b}.

Our Fractran code is easy to derive in this example: $N = 2^a 3^b$ is our initial state and $\{\frac{2}{3}\}$ is our list of fractions.

PIGAME



Theorem: J. H Conway [Con]

When started at $2^n \cdot 89$, the FRACTRAN code

$$\{\frac{365}{46}\ \frac{29}{161}\ \frac{79}{575}\ \frac{679}{451}\ \frac{3159}{413}\ \frac{83}{407}\ \frac{473}{371}\ \frac{638}{355}\ \frac{434}{335}\ \frac{89}{235}\ \frac{17}{209}\ \frac{79}{122}\ \frac{31}{183}\frac{41}{115}\ \frac{517}{89}\ \frac{111}{83}\ \frac{305}{79}\ \frac{23}{73}\ \frac{73}{71}$$
$$\frac{61}{67}\ \frac{37}{61}\ \frac{19}{59}\ \frac{89}{57}\ \frac{41}{53}\ \frac{833}{47}\ \frac{53}{43}\ \frac{86}{41}\ \frac{13}{38}\ \frac{23}{37}\ \frac{67}{31}\ \frac{71}{29}\ \frac{83}{19}\ \frac{475}{17}\ \frac{59}{13}\ \frac{41}{291}\ \frac{1}{7}\ \frac{1}{11}\ \frac{1}{1024}\ \frac{1}{97}\}$$

will terminate at $2^{\pi(n)}$, where $\pi(n)$ is the n-th digit in the decimal expansion of π.

Discussion of Conways Proof

The flowchart helps one visualize the three key steps of Conway's construction. For a fixed n, our goal is to find the n-th digit of the decimal expansion of π. From node 89 until node 83 (Phase 1), Conway generates a (very) large positive even number $E \geq 4 \times 2^{10^n}$. In Phase 2 (Node 83 to Node 41), Conway constructs two numbers

$$N_E = 2 * E \cdot (E-2)^2 \cdot \ldots \cdot 2^2 \quad \text{and} \quad D_E = (E-1)^2 \cdot (E-3)^2 \cdot \ldots 3^2 \cdot 1^2$$

. The key idea is to use Wallis' product to approximate π which dates from 1655. Namely

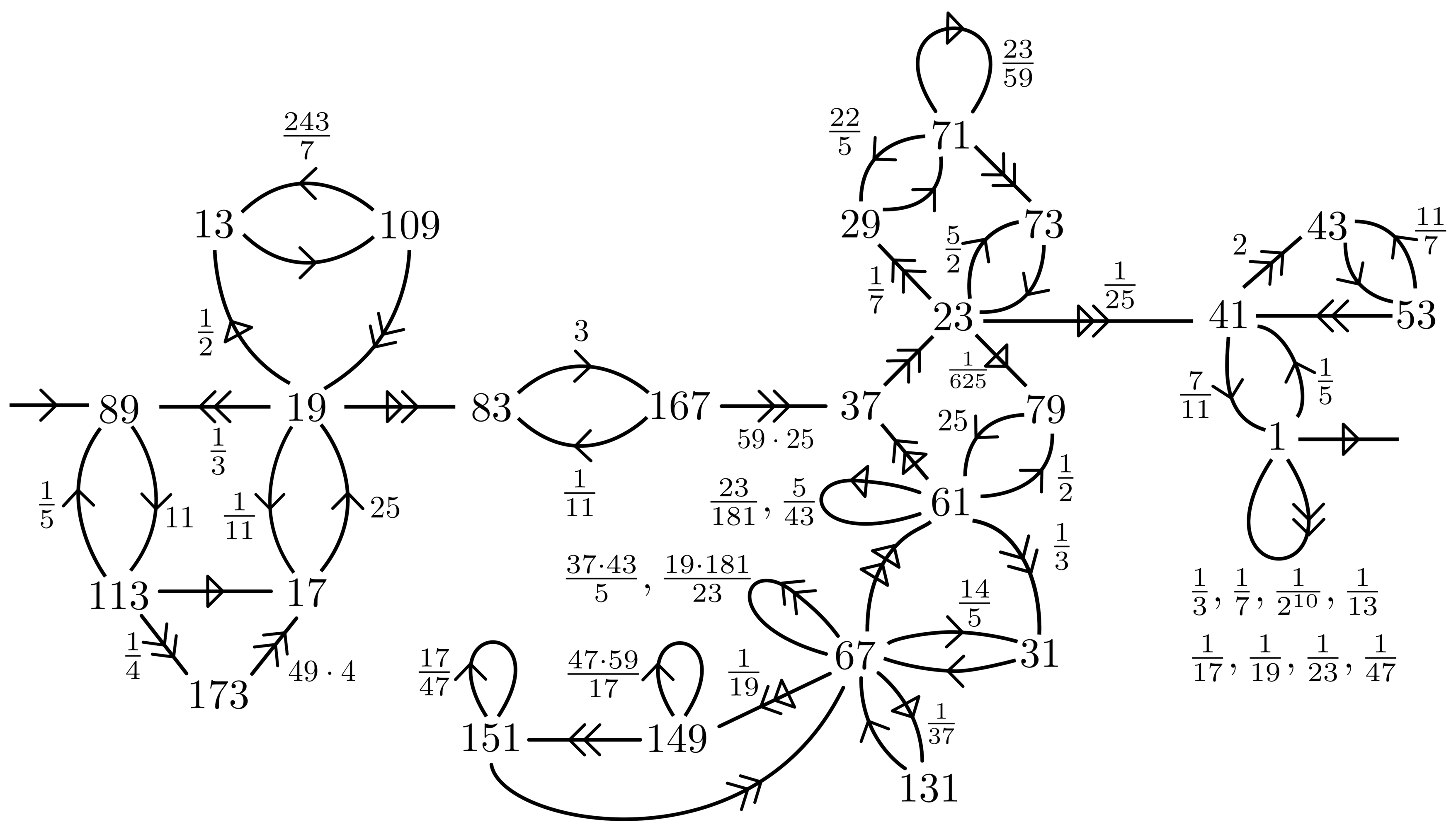
$$\pi = \lim_{E \to \infty} \frac{N_E}{D_E} = 2\left(\frac{2}{1}\right)\left(\frac{2}{3}\right)\left(\frac{4}{3}\right)\left(\frac{4}{5}\right)\left(\frac{6}{5}\right)\ldots$$

In Phase 3 (Node 41 to the end) Conway computes the integer part of

$$\frac{10^n N_E}{D_E}.$$

Multiplying by 10^n shifts the decimal unit of $\frac{N_E}{D_E}$ exactly n places to the right. Taking the floor function turns this into an integer, and reducing mod 10 allows us to find what was the n-th term in the decimal expansion of π. There are several subtle issues here: one has to compute explicitly how close $\pi_E = \frac{N_E}{D_E}$ is to π since it is just an approximation. Another problem comes from the well-known fact that $1 = 0.9999\ldots$. This means two numbers can be very close together but have differing decimal expansions. Conway's argument to show this cannot happen uses Mähler's famous irrationality measure of π, a difficult piece of mathematics. We found an elementary proof which also adapts directly to our main theorem. A notable advantage of the language is that one can compute the Gödel number of any computation explicitly: Conway does this for PIGAME.

SQRT2GAME



Main Theorem

When started at $2^n \cdot 89$, the Fractran code

$$\{\frac{143}{89}, \frac{89}{565}, \frac{833}{113}, \frac{475}{17}, \frac{17}{209}, \frac{89}{57}, \frac{13}{38}, \frac{83}{19}, \frac{109}{13}, \frac{3159}{763}, \frac{19}{109}, \frac{501}{83}, \frac{83}{1837}, \frac{54575}{167}, \frac{23}{37},$$
$$\frac{365}{46}, \frac{29}{161}, \frac{79}{14375}, \frac{41}{575}, \frac{23}{73}, \frac{71}{29}, \frac{2323}{4189}, \frac{638}{355}, \frac{73}{71}, \frac{71}{101}, \frac{1525}{79}, \frac{79}{122},$$
$$\frac{31}{183}, \frac{2921}{11041}, \frac{37}{61}, \frac{67}{31}, \frac{434}{335}, \frac{249787}{335}, \frac{539923}{1541}, \frac{131}{2479}, \frac{149}{1273}, \frac{61}{67}, \frac{61}{131}, \frac{418723}{2533}, \frac{2941}{7097},$$
$$\frac{67}{151}, \frac{151}{173}, \frac{7}{451}, \frac{129}{41}, \frac{97}{3}, \frac{97}{5}, \frac{97}{7}, \frac{97}{20000000000}, \frac{97}{13}, \frac{97}{17}, \frac{97}{19}, \frac{97}{23}, \frac{41}{1}, \frac{1}{97}, \frac{53}{43}, \frac{172}{371}, \frac{41}{265}\}$$

will terminate at $2^{\sqrt{2}(n)}$, where $\sqrt{2}(n)$ is the n-th digit in the decimal expansion of $\sqrt{2}$.

Discussion of Proof

It is obvious from the flowchart that our proof is based on Conway's. The starting point is there is a infinite product formula due to Catalan in 1874 [Cat] for $\sqrt{2}$ which is very similar to Wallis' formula: viz.

$$\sqrt{2} = \left(\frac{2^2}{1.3}\right)\left(\frac{6^2}{5.7}\right)\left(\frac{10^2}{9.11}\right)\ldots$$

It is easy to adjust Conway's proof so that the terms in N_E go down by 4 instead of 2. It is significantly harder to generate the new denominator, as it has a different form. One also has to compute how close the approximation $\sqrt{2}_E$ is to $\sqrt{2}$, and finally worry about this issue of numbers arbitrarily close togehter having very different decimal expansions. We can adapt our new argument from Conway's PIGAME to this setting to conclude our proof.

References

[Con] J.H. Conway, *FRACTRAN: A simple universal programming language for arithmetic*. Open Problems in Communication and Computation. Springer-Verlag New York, Inc. (1987), 4–26.

[Cat] E. Catalan, *Sur la constante d'Euler et la fonction de Binet*, C. R. Acad. Sci. Paris Sér. I Math. 77 (1873) 198–201.