Oberlin College
CSCI 383
Fall 2026
Lecture 23. Universality
Kozen - Section 31

Last week or so (and this week's homework): TMs match our intuition for computation.

Today: One particular capability with implications for the limits of TMs.

A language that is not recursive.

Consequence: Limits on what computer programs can do.

Universal Turing Machines

What does the following Python code do: eval(input("Please don't type Python code: "))

Python code can take Python code as input and run it.

$\Rightarrow$ Python has *universal* programs.

So does any programming language you could care about!

...can Turing machines do this?

i.e. construct a machine M that accepts inputs of the form $N\#x$:

- N description of a machine
- $x \in L(N)$

Simulates N on x.

Encoding Turing Machines

How can we describe any Turing machine using only $\{0,1\}$? (or any fixed alphabet)

A Turing machine is a 9-touple $M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$

Unary encoding: $0^n10^k10^m10^x10^y10^u10^v10^w1\ldots$

- n = number of states
- k = number of input symbols
- $m > k$ = number of tape symbols (first k are input symbols)
- x - label of left endmarker
- y - label of blank tape symbol
- u - label of s
- v - label of t
- w - label of r

Transitions are of form $\delta(p, b) = (q, c, d)$

Encoding: $0^u 1 0^v 1 0^w 1 0^x 1 0^y 1$:

In state u, reading symbol v, transition to state w, write symbol x, and go L if $y = 1$ and R if $y = 2$.

A Universal Machine

Goal: Construct a TM that takes $N\#x$ as input, and simulates N on x.

Equivalent formulation - membership problem. MP $= \{N\#x \mid N$ encodes a TM and $x \in L(N)\}$

Question: Could you write a Python program that simulates N on x?

We can make a Turing machine do the exact same thing.

- Save the description of the Turing machine N in memory.
- Simulate N on x:
  - Store simulated tape/read head.
  - Store simulated state.
  - Transitions: Table lookups.

If simulation accepts/rejects: do the same.

Total?

$\Rightarrow$ MP is RE.

Optional question: Are each of these constructions total?