CS121 Lecture 7

September 24

Coding Assignment: released

Uniform Computation (turing machines):

Functions on arbitrary length inputs

Algorithm:

- Finite recipe to compute on potentially unbounded inputs
- Computation is done w/ a sequence of basic operations
  - each operation deals w/ constant amount of information
  - # of times we repeat operations can be unbounded

- unbounded memory / arrays
- finite state / local variables
- Addressing mechanism / indexing
- Finite logic
- Looping / halting

components of programming languages

Turing Machine:

- Finite state: # in $\{0, 1, 2, \dots, k-1\}$
- Memory: tape containing symbols in $\Gamma = \{\triangleright, 0, 1, \dots\}$
- Head position: $i \in \{0, 1, 2, 3, \dots\}$
- Transition function: $\delta_M : [k] \times \Gamma \to [k] \times \Gamma \times \{L, R, S, H\}$

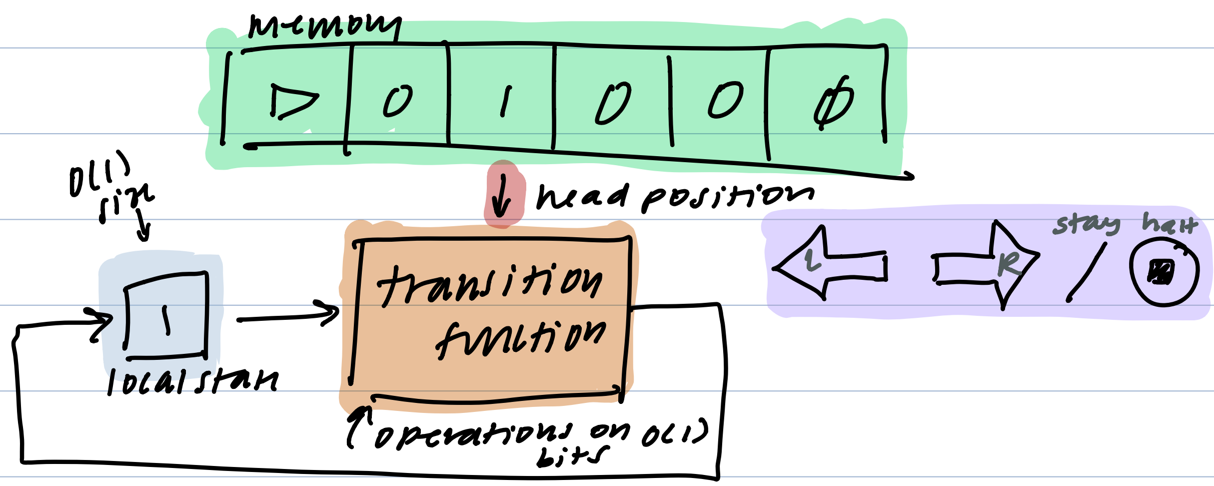
Initially:

- state $= 0$, head position $= 0$, tape $= (\triangleright, x_0, x_1, x_2 \dots x_{n-1}, \emptyset, \emptyset, \dots)$

- Each step based on current state and symbol read decide new state, symbol, and if to move Left, Right, Stay, or Halt.

Finite constants: k, $|\Gamma|$, $|\delta_M|$

Unbounded: tape contents, head position, # of steps



Q: design turing machine M s.t. for every $x \in \{0,1\}^n$, when M halts the tape is $\triangleright\, \mathrm{XOR}(x)\, \emptyset \emptyset \emptyset$

$\equiv$ computes XOR : $\{0,1\}^n \to \{0,1\}$

Boaz Turing Definitions

Def: If M is a Turing Machine and $x \in \{0,1\}^*$ then $M(x)$ denotes the contents of tape of M from the second position till the first empty spot.

If $M(X)$ doesn't halt we denote $M(X) = \perp$.

***
Important Def: A Turing Machine M computes $F: \{0,1\}^* \to \{0,1\}^*$ if for every $x \in \{0,1\}^*$, $M(x) = F(x)$

F is computable if there is a TM that computes it

Def: M represented by δ

Sipser Section 3.1: (only output 1 bit)

- States arbitrary set Q. Input alphabet Σ
- Output one bit obtained by going to accepting or rejecting state
- Task is to compute $F: \Sigma^* \to \{0,1\}$, aka decide $x \in L$ for $L \subseteq \{0,1\}^*$

$\delta: Q \times$

Def: M is a 7-tuple $(Q, \Sigma, \Gamma, q_0, q_{accept}, q_{reject})$

Central Definition

- TM M computes $F:\{0,1\}^* \to \{0,1\}^*$ if for every $x \in \{0,1\}^*$ on input x the TM halts and outputs F(x)

Central Observation / Thm

F computable by a TM iff F computable by

turing complete/ equivalent: Python / C / Java / OCaml, NAND-TM program, Cellular Automation, λ calculus, RAM machines, ...

Church-Turing Thesis

F computable by a TM iff F computable by any physical means

TM Program

```
while (true):
    sym = Tape[i]
    state, sym, move = transition(state, sym)
    Tape[i] = sym
```

NAND-TM : programming language variant of TMs

- NAND-TM = NAND-CIRC + loops + arrays
- every line is similar except:
  - index i initialized to 0
  - Foo[i] = i^{th} element of Foo
  - X, Y (and X_nonblank, Y_nonblank) for in/out

- last line is special MODANDJUMP(a,b)

a	b	Do
0	b	halt
0	1	i--, go to start
1	0	go to start
1	1	i++, go to start