Oberlin College
CSCI 383
Fall 2026
Lecture 20. Turing Machines
Kozen - Section 28

First 1/3 of course: DFAs - finite memory.

Second 2/3 of course: NPDAs - finite memory + stack (infinite memory, limited access)

Today: Infinite memory, easy access.

Turing Machines, Informally

Setup for DFA:

- Input was written on tape.
- Read left to right, transitioning through states. (Finite control.)

Setup for NPDAs:

- Left to right, finite control was same as DFA
- Added stack (and nondeterminism)

Turing machines (deterministic):

- Input written on tape. (as before)
- Read head attached to finite control. (as before)
- Read head can move left or right.

  Need to add a left endmarker $\vdash$
- Can write as we go.
  We have infinite space to write: tape is $x\sqcup^\omega$
  That is, every step, the machine:
  1. Reads the current character.
  2. Writes a new character. (can be in tape alphabet, or blank, or some other special character)
  3. Moves left or right.
  4. Changes state.
- Acceptance: Accept state

Example: $a^n b^n c^n$. Two phases.

- Phase 1: Test $a^*b^*c^*$. Not of the right form? Reject. Write $\dashv$
- Phase 2: Count a's, b's, and c's.
  - Sweeping right to far left: cross out one a, one b, and one c.

    Can't find one of those letters? Reject.
  - Sweeping left to right: same thing.
- Final pass: If entire tape is blank, accept.

Turing Machines, Formally

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

- Q - states
- Σ - input alphabet (stuff that starts on tape)
- Γ - tape alphabet (stuff you're allowed to write)
- δ - transitions.

Special stuff.

- $\vdash \in \Gamma \setminus \Sigma$ - left endmarker
- $\sqcup \in \Gamma \setminus \Sigma$ - blank tape symbol
  Initial tape: $\vdash x \sqcup^{\omega}$
- $s \in Q$ - start state
- $t \in Q$ - accept state (sink)
- $r \in Q$ - reject state (sink)

Transitions: $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$.

Interpretation of $\delta(p, a) = (q, b, d)$:

- I'm in state $q \in Q$, I'm reading $a \in \Gamma$
- Write b on tape.
- Go direction d
- Transition to state p

Special cases:

- For all p, $\delta(p, \vdash) = (q, \vdash, R)$ for some q.
- For all b, $\delta(t, b) = (t, c, d)$ for some c, d.
- For all b, $\delta(r, b) = (r, c, d)$ for some c, d.

Languages of Turing machines: Can define configurations and transition relation. (See book. Don't need them yet.)

Three things a TM can do:

1. Run until it enters an accept state.
2. Run until it enters a reject state.
3. Runs forever.

Def: A Turing machine *halts* if item 1. or item 2. happens.

$L(M)$ is the set of all strings where the TM halts and accepts.

Def. A Turing machine is *total* if it halts on every input.

A language A is:

- *recursively enumerable* if there exists a TM M such that $L(M) = A$
- *recursive* if there exists a *total* TM M such that $L(M) = A$.