Modeling Computation

Introduction to Formal Languages and Automata

DFAs

What is a “standard computer”?

- It reads input, and stores it in memory.
- It computes functions of the input, using an arithmetic/logic unit, and stores the result in memory.
- It produces output.

What is the “simplest” form of a computer?

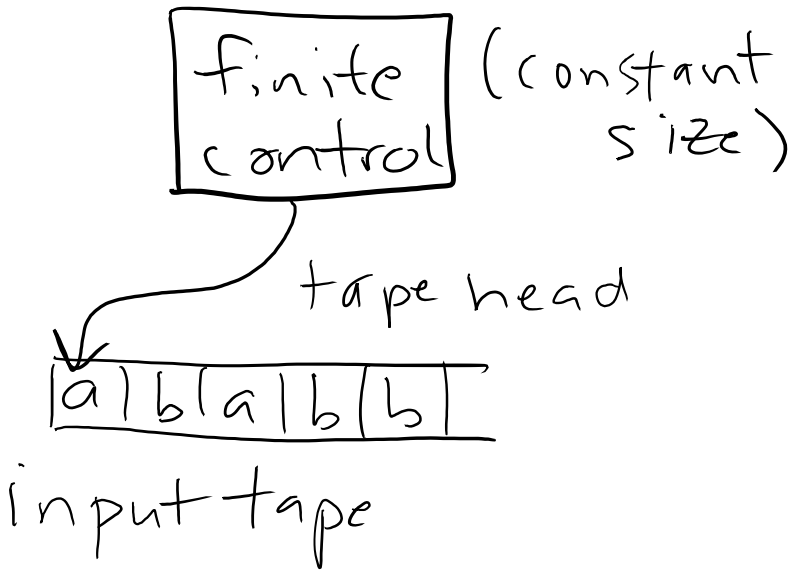
- Have no memory, other than keeping track of which state (from a finite set of states) the computer is in.

- Have output be 0 (reject)
  or 1 (accept)

Computers carry out Computation

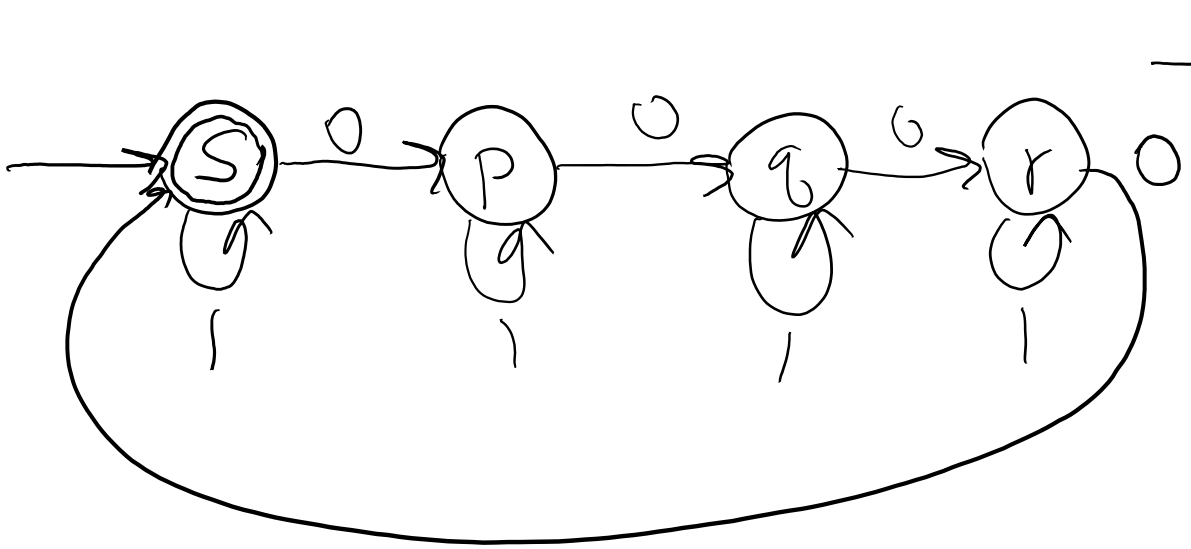
- What is computation?
  - the movement and transformation of data
  - the act or action of computing.
  - the use of a computer
    "computing"
  - an idea in flux.

Finite-State Machine (Automaton)



- Reads input from input tape using a movable tape head.
- At any time, is in one of a finite number of states.
- Reads a symbol from the input tape and changes state based on a specified function.
- Certain states are favorable (accepting)
- The input word is accepted if the machine is in a favorable after the last symbol of the input.

Example: A FSM to determine whether the number of 0s in the input is a multiple of 4.



: start/initial state

: favorable

1 0 0 1 0 1 1 0 s, s, p, q, q, r, r, r, (s) accept

1 0 0 1 0 1 1 0 0 s, s, p, q, q, r, r, r, s, p reject

Deterministic Finite Automata (DFA)

- $M = (Q, \Sigma, \delta, s, F)$
- Q is the finite set of states
- Σ is the finite input alphabet
- δ is the transition function. $\delta : Q \times \Sigma \to Q$
- $s \in Q$ is the initial state
- F $F \subseteq Q$ is the set of favorable states.

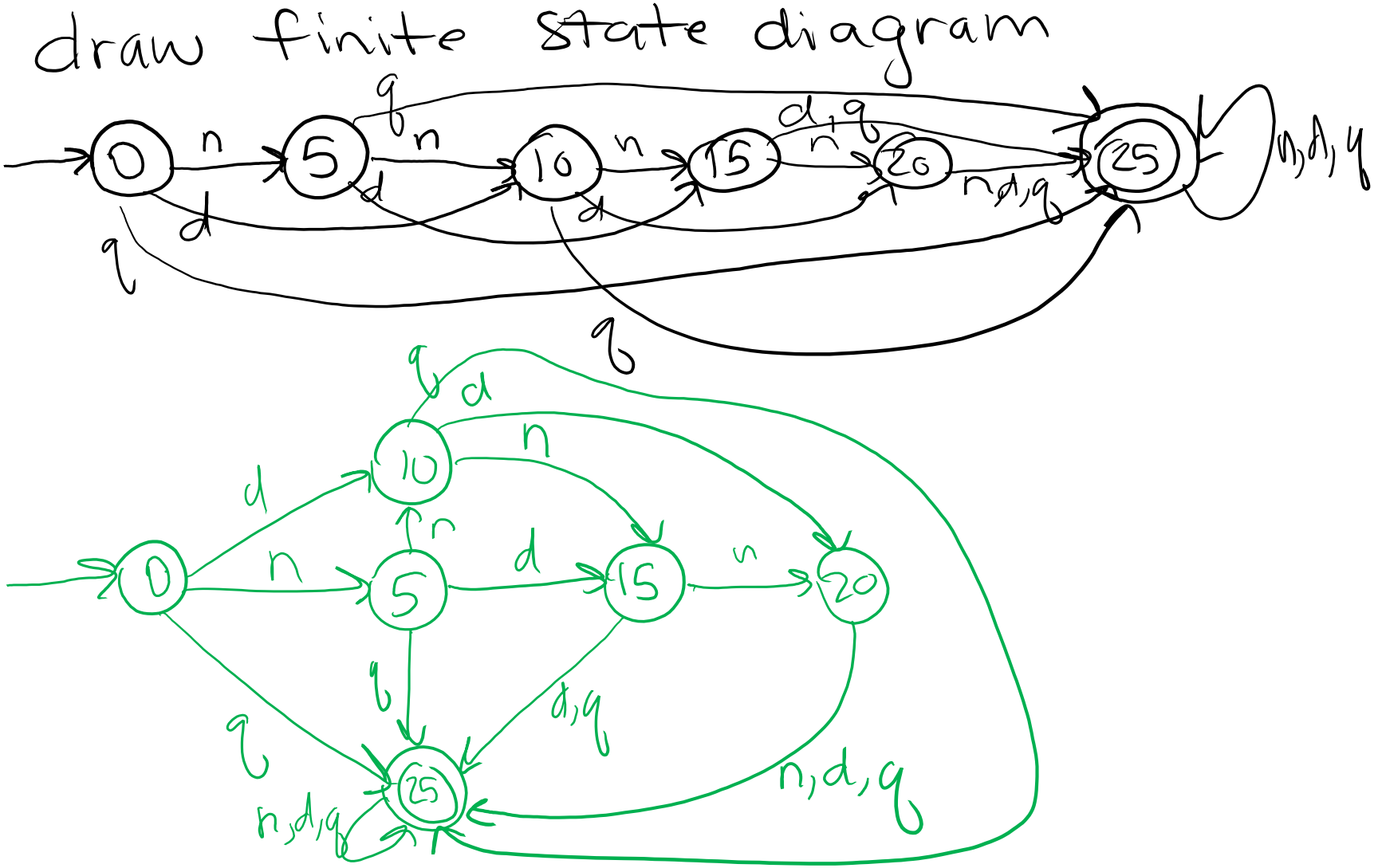
Example: DFA for determining whether the number of 0s in the input is a multiple of 4.

- $Q = \{s, p, q, r\}$
- $\Sigma = \{0, 1\}$
- δ

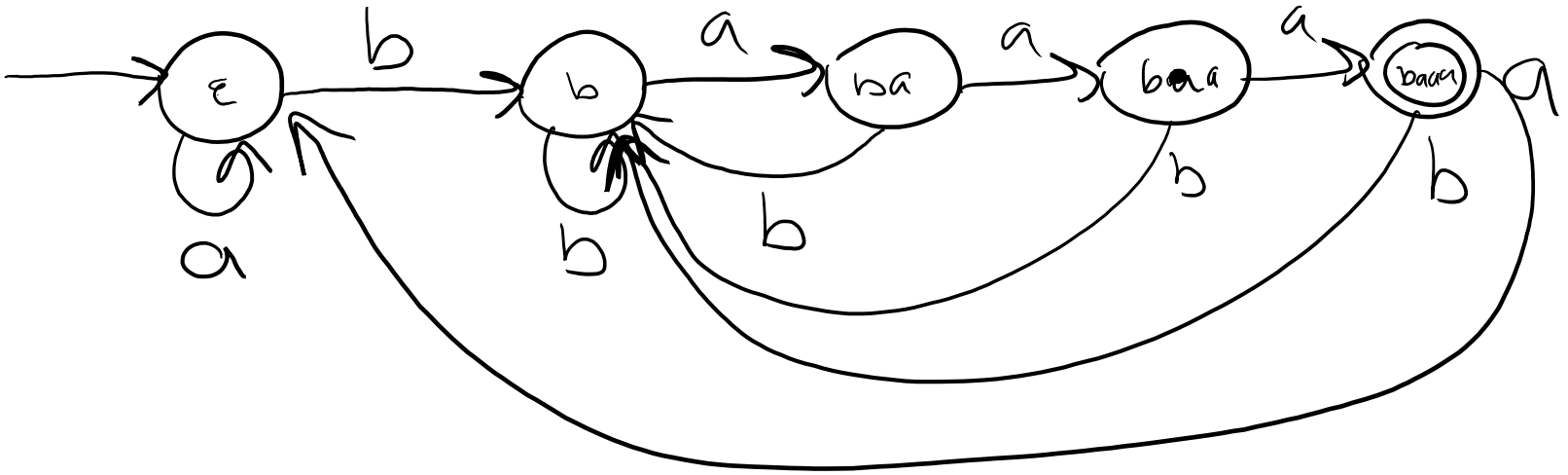
$$\begin{cases} \delta(s,0) = p & \delta(s,1) = s \\ \delta(p,0) = q & \delta(p,1) = p \\ \delta(q,0) = r & \delta(q,1) = q \\ \delta(r,0) = s & \delta(r,1) = r \end{cases}$$

- $s = s$ is the initial state
- $F = \{s\}$

Example: A newspaper vending machine unlocks the door when at least 25¢ in nickels(5¢), dimes(10¢), and quarters(25¢) has been input. No change is given.



Example: DFA that accepts any string in $\{a, b\}^*$ that ends with $baaa$.



Exercise: Design a DFA that accepts any string in $\{0,1\}^*$ that has a number of 1s divisible by 3 and a number of 0s divisible by 2.