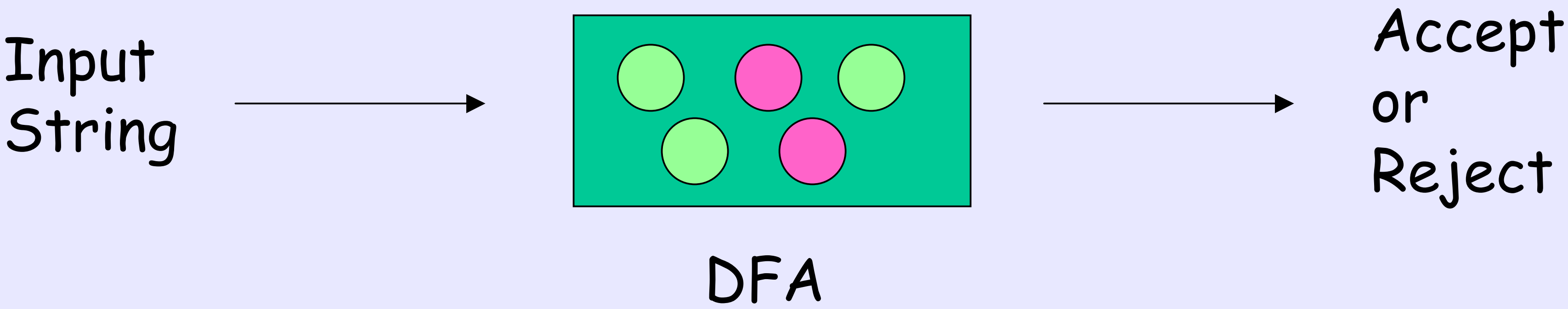
CS5371
Theory of Computation

Lecture 3: Automata Theory I (DFA and NFA)

Objectives

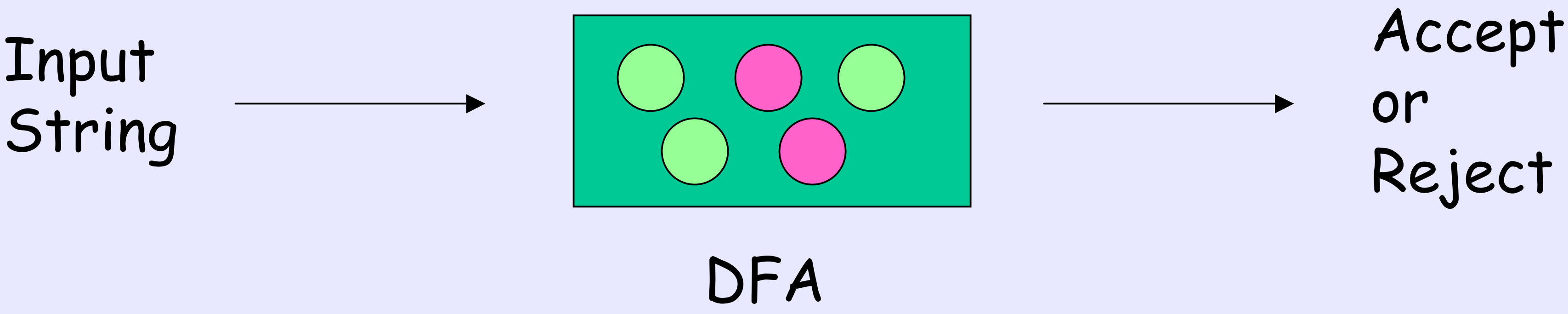
- This time, we will look at how to define a very simple “computer” called deterministic finite automaton (DFA)
- Show that DFA can solve some string decision problem
- Then, we give slightly change the definition of DFA to give another computer called non-deterministic finite automaton (NFA)

DFA



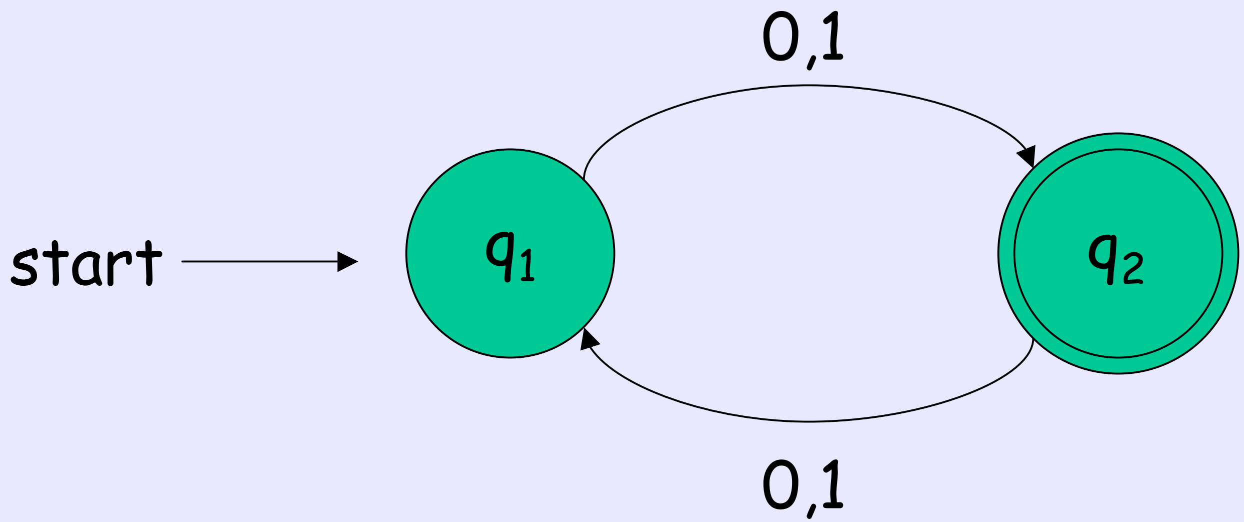
- A machine with finite number of states, some states are accepting states, others are rejecting states
- At any time, it is in one of the states
- It reads an input string, one character at a time

DFA



- After reading each character, it moves to another state depending on what is read and what is the current state
- If reading all characters, the DFA is in an accepting state, the input string is accepted.
- Otherwise, the input string is rejected.

Example of DFA

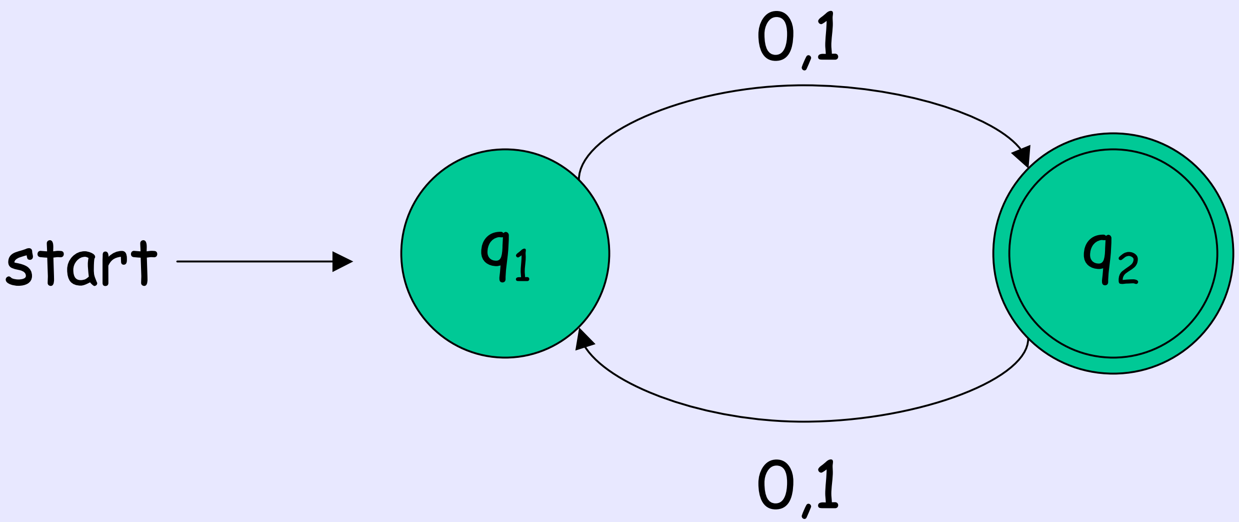


- The circles indicates the states
- If accepting state is marked with double circle
- The arrows pointing from a state q indicates how to move on reading a character when current state is q

Formal Definition of DFA

- A DFA is a 5-tuple $(Q, \Sigma, \delta, q_{start}, F)$, where
  - Q is a set consisting finite number of states
  - Σ is an alphabet consisting finite number of characters
  - $\delta: Q \times \Sigma \rightarrow Q$ is the transition function
  - q_{start} is the start state
  - F is the set of accepting states
- Note: only 1 start state, and can have many accepting states

Formal Definition of DFA



$Q = \{q_1, q_2\}, \Sigma = \{0, 1\}, q_{start} = q_1, F = \{q_2\}$

$\delta(q_1, 0) = q_2, \quad \delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_1, \quad \delta(q_2, 1) = q_1$

Formal Definition of DFA's Computation

- Recall how a DFA performs its computation (to decide whether an input string is accepted or rejected):
  - If after reading the whole string, DFA is in an accepting state, then the input string is accepted; otherwise the input string is rejected
- How to define this formally??

Formal Definition of Computation

- Let $M = (Q, \Sigma, \delta, q_{start}, F)$ be a DFA
- Let $w = w_1 \, w_2 \, \ldots \, w_n$ be a string with each w_i a member of the alphabet Σ
- Then, M accepts w if a sequence of states $r_0, r_1, \ldots, r_n$ in Q exists with the three conditions:
  - $r_0 = q_{start}$
  - $\delta(r_i, w_{i+1}) = r_{i+1}$
  - $r_n \in F$

Some Terminology

Let M be a DFA

- Among all possible strings, M will accept some of them, and M will reject the remaining
- The set of strings which M accepts is called the language **recognized** by M
- That is, M **recognizes** A if

$$A = \{ w \mid M \text{ accepts } w \}$$

Quick Quiz

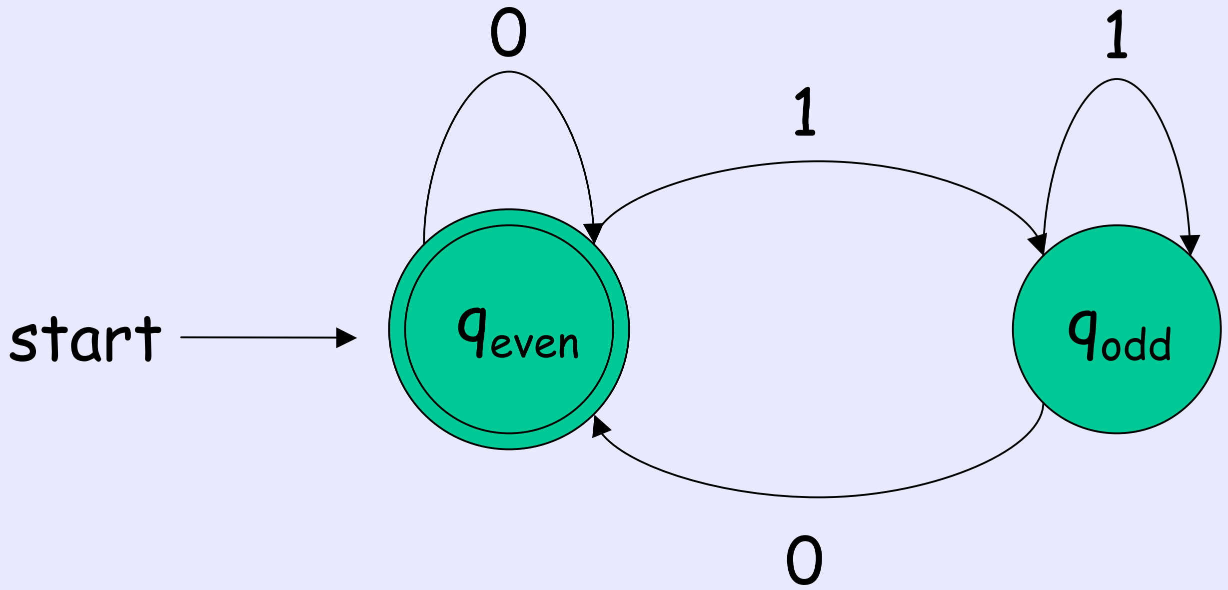
- What is an alphabet?
- What is a language?
- What are the min and max numbers of strings a DFA can accept?
- What are the min and max numbers of languages a DFA can recognize?

Designing a DFA [Example 1]

- How to design a DFA that accepts all binary strings that represent an even number? (e.g., accepts 110, 010, but rejects 111, 010101)
- Let's pretend ourselves as a DFA…
  - After reading a character, we must decide whether the string seen so far is in the language (what is the language we're talking now??) because we don't know if it is the last character…

Designing a DFA

Answer:



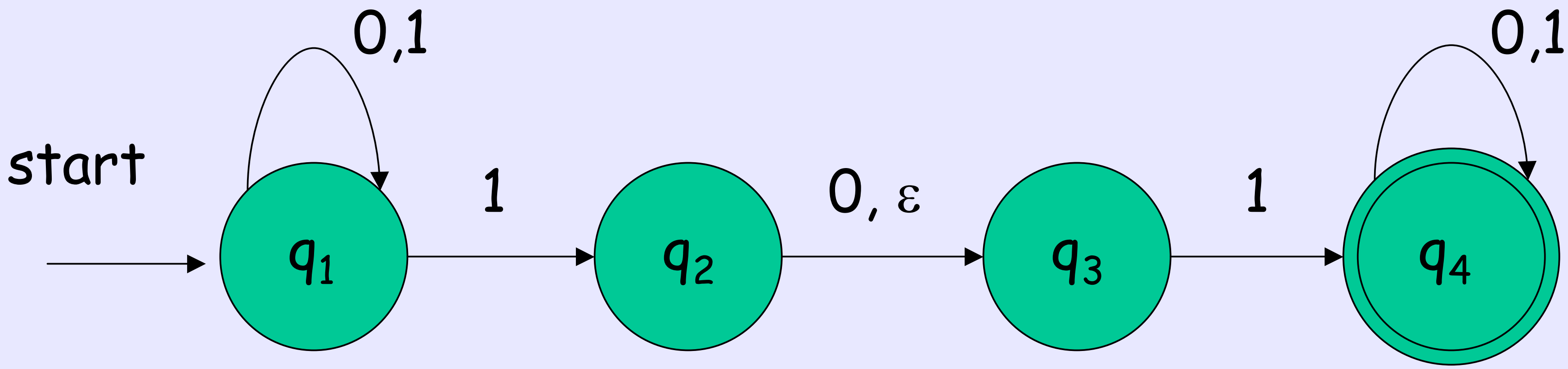
Designing a DFA (Quick Quiz)

- How to design a DFA that accepts all binary strings representing a multiple of 5? (E.g., 101, 1111, 11001, ...)

NFA

- Every step of DFA's computation follows in a unique way from the preceding step
  - When a machine is in a given state, and reads the next input character, we know what the next state is
- In an NFA, several choices may exist for the next state

Example of NFA

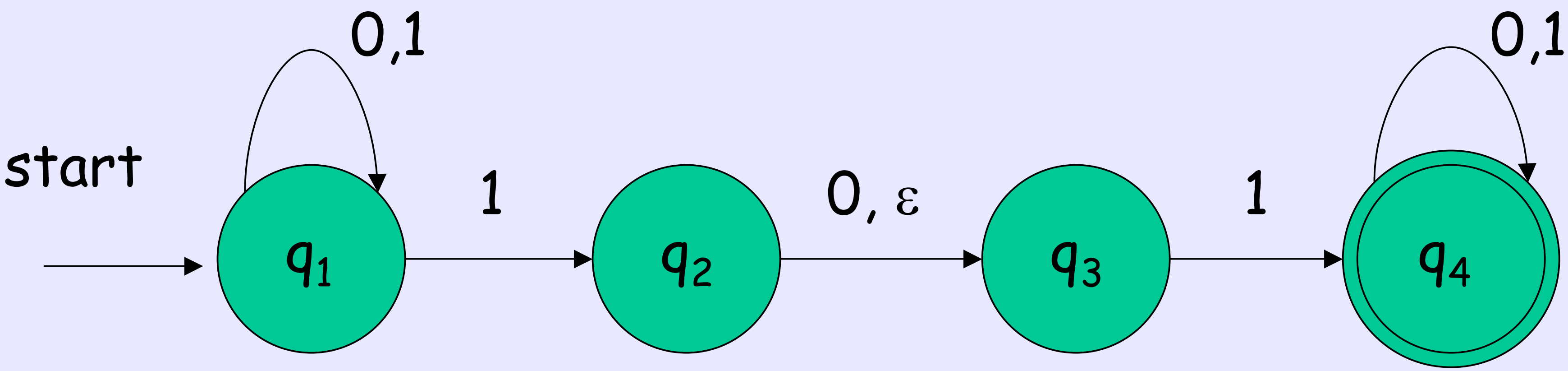


- Difference with DFA
  - Can move to more than 1 states, or nowhere, after reading a character (E.g., at q_1, on reading 1, we can move to q_1 or q_2; at q_2, on reading 1, nowhere to go!)
  - Can move to another state without reading anything (E.g., at q_2, the symbol ε on arrow pointing at q_3 indicates that we can move to q_3 without reading anything)

What is accepted by NFA?

- Let M be an NFA
- (Informally,) M accepts a string w if there is at least one way to move from the start state to a final state according to the transition arrows, such that the concatenation of the true characters (that is, ignoring ε) used by the transition is equal to w

What is accepted by this?



Next time

- Continue the discussion on NFA
- Properties of language recognized by DFA or NFA