CONSTRUCTION OF DFA EXTENDED TRANSACTION FUNCTION

Lecture 6

DFA CONSTRUCTION

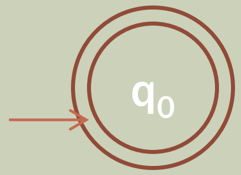
- It categories in 4 types:
  - Language = has string starting with
  - Language = has string end with
  - Language =has string which contains substring
  - Construct DFA with divisibility and conditional problems

  Examples:

  1) L= { w | contains strings which ends with “100”}
  2) L= { w | contains strings which starts with “aba” }
  3) L= {w | contains strings which has substring of “abba”}
  4) L= {Σ=(0,1,2…9), |w contains all strings which are divisible by 3
  5) L = { w |strings over Σ(a,b) , has |w|≥2} infinite
  6) L = { w |strings over Σ*(a,b) , has |w|≤2} finite
  7) L = { w |strings over Σ*(a,b) , has |w|mod2=0} infinite

DFA CONSTRUCTION

- Tips to draw DFA:
  - For ε accept by DFA : draw initial state as a final state



  - Define automata tuples **M={ Q, Σ, q0, F, δ}**
  - Transition Table and diagram

EXERCISE

- 1. Give a DFA for Σ = {0, 1} and strings that have an odd number of 1's and any number of 0's.

- 2. Give a DFA for Σ = {a, b, c} that accepts any string with aab as a substring.

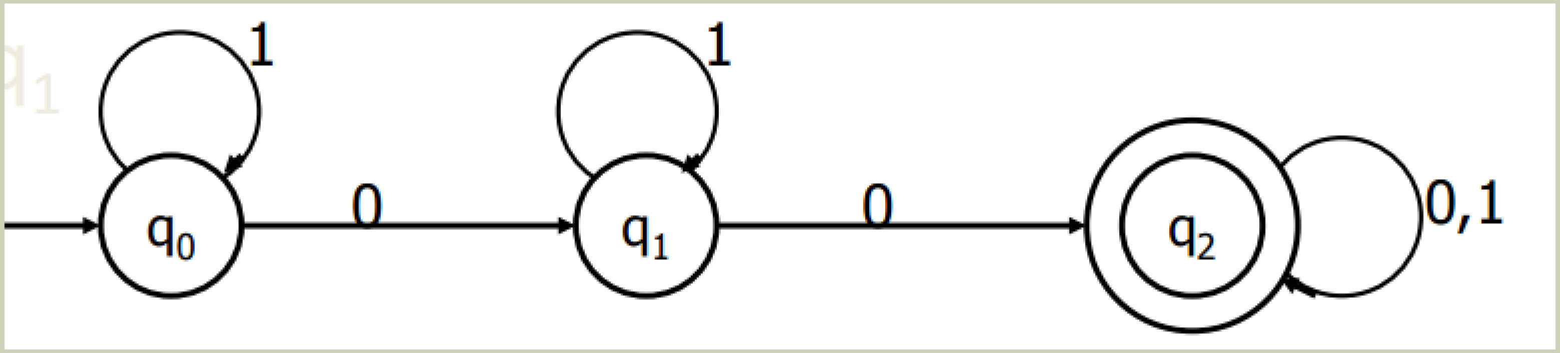
- 3. Give a DFA for Σ = {a, b} that accepts any string with aababb as a substring.

EXTENDING THE TRANSACTION FUNCTION TO STRINGS

- The DFA define a language: the set of all strings that result in a sequence of state transitions from the start state to an accepting state
- Extended transition function
  - Describes what happens when we start in any state and follow any sequence of inputs
  - If δ is our transition function, then the extended transition function is denoted by $\hat{\delta}$
  - The extended transition function is a function that takes a state q and a string w and returns a state p (the state that the automaton reaches when starting in state q and processing the sequence of inputs w)

EXTENDING THE TRANSACTION FUNCTION TO STRINGS

- Formally, the function $\hat{\delta} : Q \times \Sigma^* \to Q$
- • is defined recursively:
  - Note $\hat{\delta}(q,\varepsilon)=q$
  - Example : Trace 1101001 on the diagram below starting,



  - $\hat{\delta}(q0,1101001)= \hat{\delta}(\,\delta(q0,1)\,,101001)$ ….

RECURSIVE DEFINITION OF THE EXTENDED TRANSITION FUNCTION

- Definition by induction on the length of the input string
- **Basis**: $\hat{\delta}(q, \epsilon) = q$
- If we are in a state q and read no inputs, then we are still in state q
- **Induction**: Suppose w is a string of the form xa; that is a is the last symbol of w, and x is the string consisting of all but the last symbol Then:
  - $\hat{\delta}(q, w) = \delta(\hat{\delta}(q, x), a)$

RECURSIVE DEFINITION OF THE EXTENDED TRANSITION FUNCTION

- Flow
  - To compute δˆ(q, w), first compute δˆ(q, x), the state that the automaton is in after processing all but the last symbol of w
  - Suppose this state is p, i.e., δˆ(q, x) = p
  - Then ˆδ(q, w) is what we get by making a transition from state p on input a - the last symbol of w