

DAGs: directed acyclic graphs

topological ordering

a graph is a DAG iff it has a topological ordering of its vertices



shortest paths in DAG via topological ordering

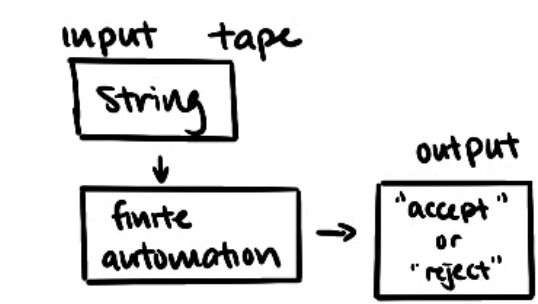
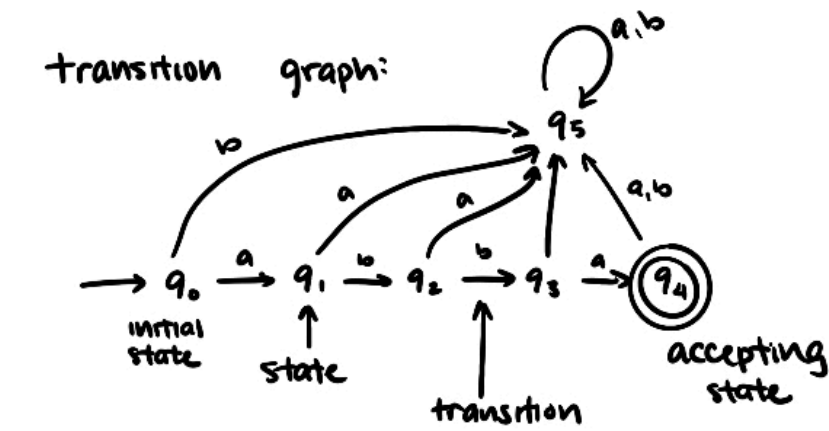
$G = (V, E)$ $|V| = n$ linear time: $O(n)$
 $(u, v) \in E$ weight (u, v)
 L : topological sort of vertices
 $L[1], L[2], \dots, L[n]$

d : an array of length n ; $d[i]$ = shortest path cost from vertex 1 to i
 $d[1] = 0$; $d[u] = \infty, 2 \leq u \leq n$

p : an array of pointers: backtracking pointers.
 $p[v] = -\infty, 1 \leq v \leq n$
 $p[v]$ = the predecessor of vertex v in the shortest path from 1 to v .

for $i = 1$ to n do
 $u = L[i]$
 for every v such that $(u, v) \in E$
 let $w = \text{weight}(u, v)$
 if $d[v] > d[u] + w$
 then
 $d[v] = d[u] + w$
 $p[v] = u$.

deterministic finite automata (DFA)

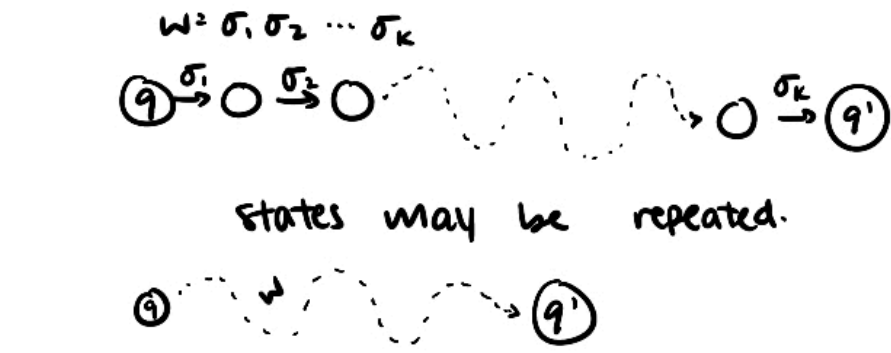


- for every state, there is a transition for every symbol in the alphabet
- ex: $a, b, b, a = \checkmark$, accept
 $a, b, a = \times$, reject
- to accept a string:
 - all the input string is scanned and the last state is accepting
- to reject a string:
 - all the input string is scanned and the last state is non-accepting

formal definition:
 $M = (Q, \Sigma, \delta, q_0, F)$
 Q : set of states
 Σ : input alphabet $\lambda \in \Sigma$
 δ : transition function
 q_0 : initial state
 F : set of accepting states

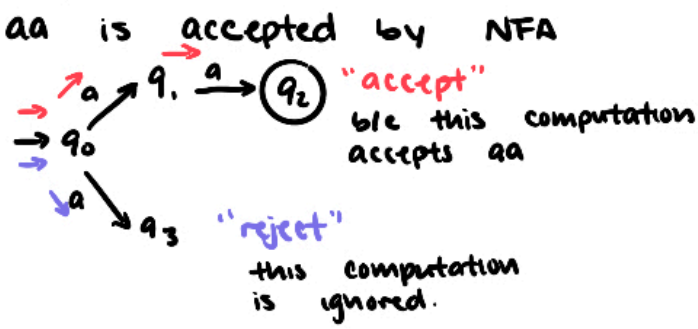
extended transition func:
 $\delta^*: Q \times \Sigma^* \rightarrow Q$
 $\delta^*(q, w) = q'$
describes the resulting state after scanning w from state q .

in general: $\delta^*(q, w) = q'$
implies that there is a walk of transitions



non-deterministic finite automata (NFA)

- an NFA accepts a string:
 - if there is a computation of the NFA that accepts the string
 - i.e. all the input string is processed and the automaton is in an accepting state.
- example:



- an NFA rejects a string:
 - if there is no computation of the NFA that accepts the string
 - for each computation:
 - all the input is consumed and the automaton is in a non-final state
- OR
- the input cannot be consumed