

Shortest Path Using Topo Sort

A directed graph is a DAG iff it has a topological ordering

For $G=(V,E)$, $(u,v) \in E$, $L \equiv$ list of topo sort of V .
Let $d \equiv$ array of length $|V|=m$; $P \equiv$ array of pointers
s.t. $d[1]=0$, $d[u]=\infty$, $P[v]=-\infty$

1. FOR $i=1:n$: $u=L[i]$

2. FOR ALL v s.t. $(u,v) \in E$: $w = \text{weight}(u,v)$

3. IF $d[v] > d[u] + w$: $d[v] = d[u] + w$

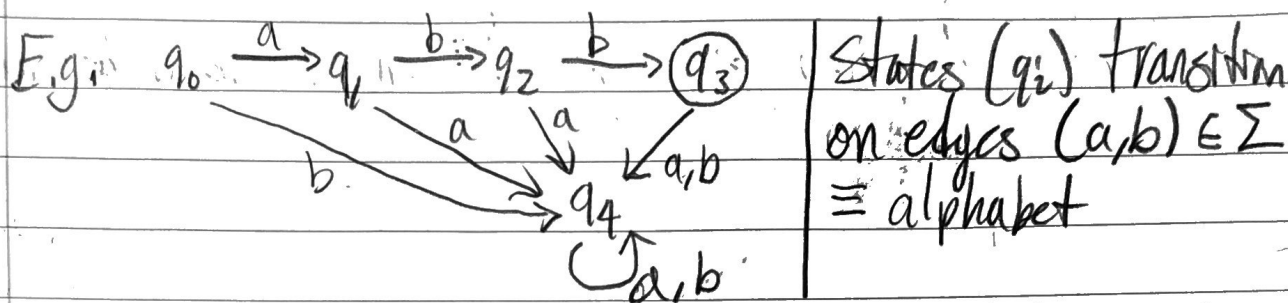
4. $P[v] = u$

This produces shortest path in d , which
we can use for alignment problems.

End Ch.1: Sequence Alignment

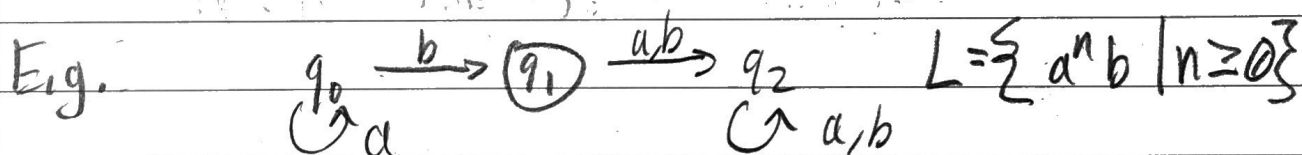
LO7: Finite Automata

The idea w/ DFA is we convert inputs to output.
Model sequences as transition graphs!



In this case, q_3 is our accepting state. Only inputs that reach the accepting state at the end of the input are accepted. Empty input λ also possible.

Let $L \equiv$ alphabet of accepted inputs ($L = \{ 'abb' \}$ here)



Formally, DFA: $M = (Q, \Sigma, \delta, q_0, F)$ where

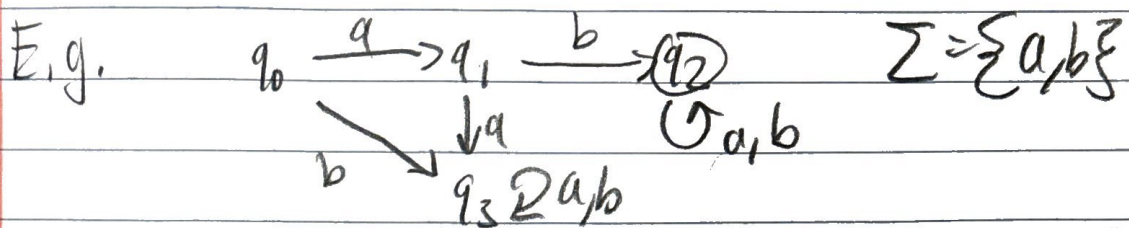
- Q : set of states
- q_0 : initial state
- Σ : alphabet $\lambda \notin \Sigma$
- F : set of accepting states
- δ : transition fcn $[\delta(q, x) \rightarrow q] \quad F \subseteq Q$

In the prior example: $Q = \{ q_0, q_1, q_2 \}$, $\Sigma = \{ a, b \}$, $F = \{ q_1 \}$.

		a	b
δ :	q_0	q_0	q_1
	q_1	q_2	q_2
	q_2	q_2	q_2

Define an extended transition fn δ^* that takes in words w as opposed to just single letters.

$\delta^*(q, w) \rightarrow q' \quad \Sigma^* \equiv$ all possible words including λ

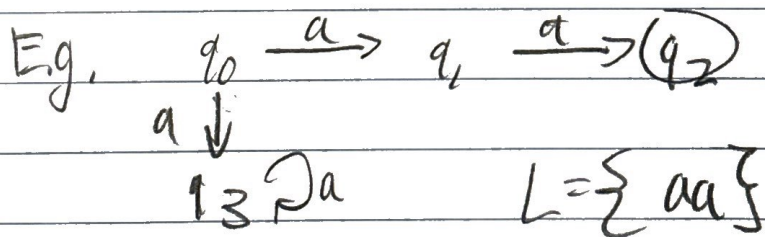


Language accepted $L(M) = \{ \text{any input w/ prefix } ab \}$

A language L is regular if it's accepted by a DFA M .

In deterministic model, each state has 1 edge for each letter of the alphabet.

Non-deterministic models allow multiple edges per letter.



Inputs are accepted if $\exists$ a path to acceptance.

We define NFA: $M = (Q, \Sigma, \delta, q_0, F)$ where only difference from DFA is in δ .