Nondeterminstic Finite Automata

(slightly different from book) 8/27/18

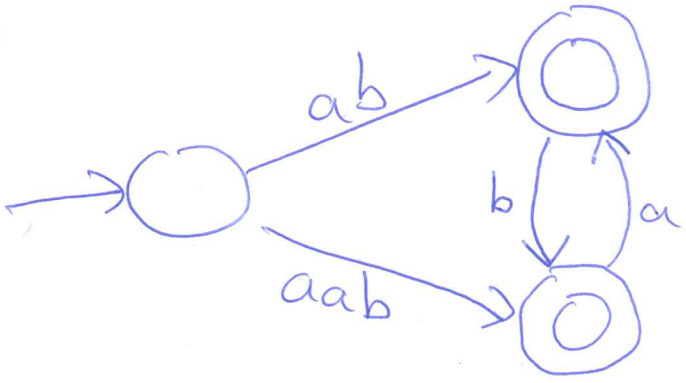
Arrows out of a state are labelled by strings (possibly λ), not single letters.

Nondeterministic: Possibly more than one choice of how to exit a state. (Possibly zero choices).

The machine "guesses" which choice to take if there is one - it will always guess towards accepting the string.

Or: it takes all possible paths - if there is one that leads to accepting, it accepts.

E.g.



abbb

Formal definition:

A NFA is a tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

Q is a finite set of states.

Σ is an alphabet

$\delta \subseteq (Q \times \Sigma^*) \times Q$ is the set of valid transitions

(Σ^*: strings on the alphabet)

$q_0 \in Q$ is the initial state

$F \subseteq Q$ is the set of final states.

This is the part that is different from a DFA.

Def: An NFA M accepts w starting at state q if

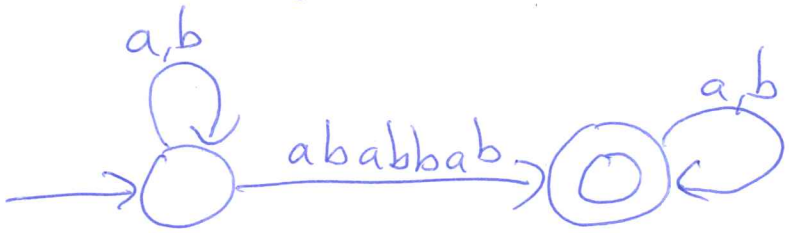
a) $w = \lambda$, and $q \in F$

b) ~~$w = uv$ where $u, v \in \Sigma^*$~~ There exist $u, v \in \Sigma^*$, ~~s.t. $w = uv$, and there~~ $q' \in Q$, s.t. $w = uv$, $(q, u, q') \in \delta$, and M accepts v starting at q'.

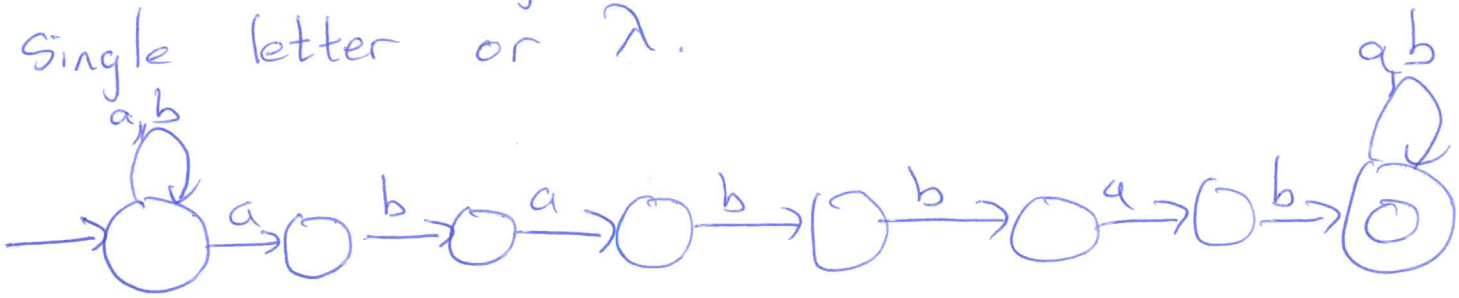
Why NFAs?

- We'll soon see that NFAs are equivallent to DFAs: Given an NFA M, there is a way to construct a DFA M' that accepts and rejects the same strings.
  - It usually will be easier to come up w/ an NFA for a language than a DFA.

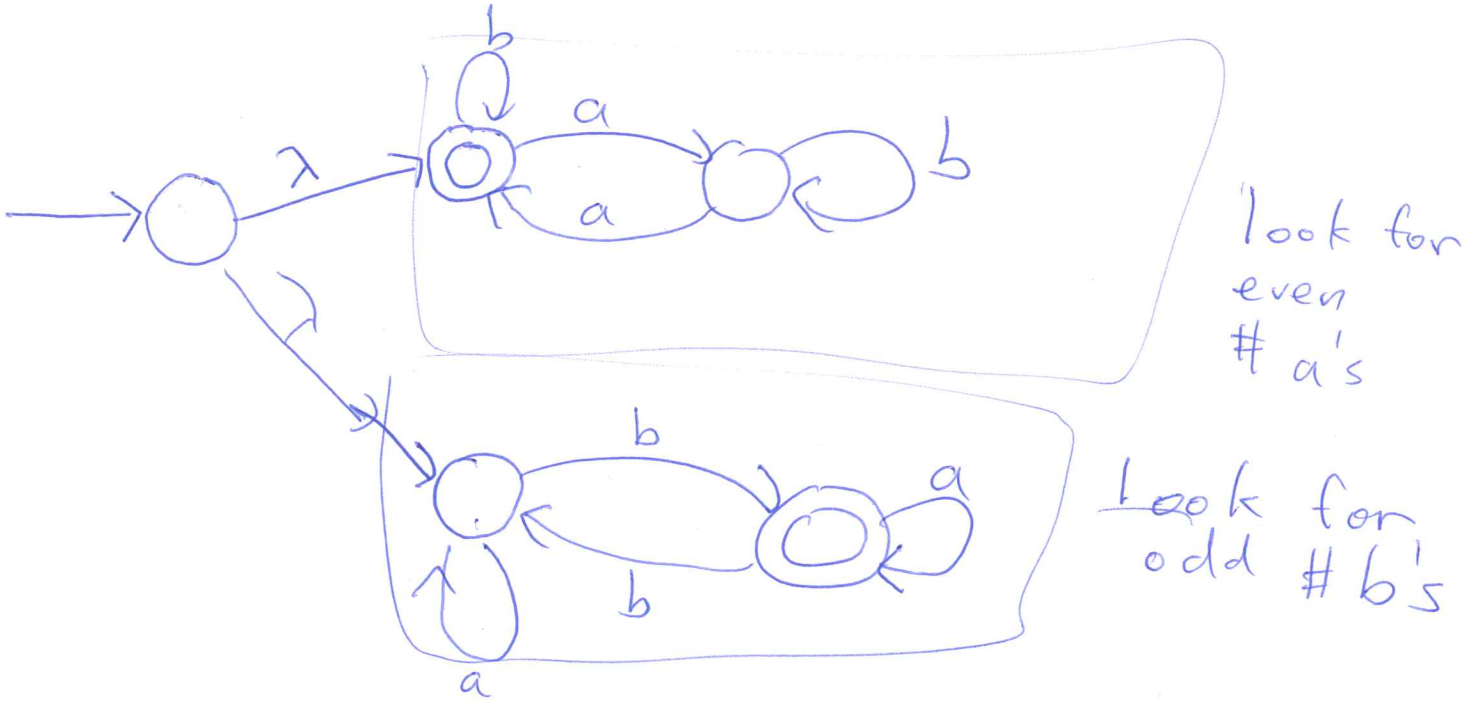
E.g Construct an NFA that accepts strings containing ababbab somewhere (consecutively)

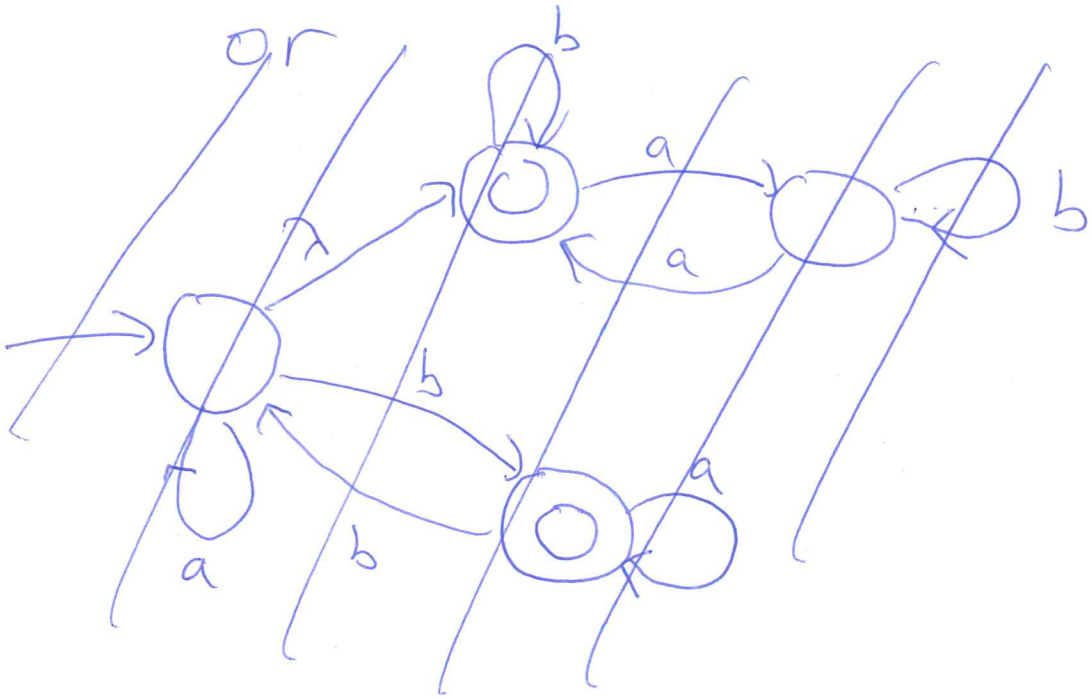


The book only allows transitions w/ a single letter or λ.



Construct an NFA ~~for~~ (on $\{a,b\}$) ~~for~~ accepting all strings w/ an even # of a's or an odd number of b's.





Why do we require finitely many states?

If we allowed infinitely many states, we could simply list all the strings we want to accept and have an arrow from q_0 to ~~each~~ a (final) state for each string.

Then any set of strings could be the set accepted by this infinite automaton.

Def: A language ~~is~~ on an alphabet Σ is some subset of Σ^*. (i.e. a set of strings)

Def: The language accepted by an automaton M (The language of an automaton) is

$$L = \{ w \in \Sigma^* \mid M \text{ accepts } w \}$$