Non-deterministic Automata & ϵ-transitions

- Non-deterministic Finite Automata (NFA)
  - Informal notion:
    - With deterministic automata each transition was from a single state to a single state

$$\delta(\sigma, q_i) = q_j \quad \sigma \in \Sigma,\ q_i, q_j \in Q$$

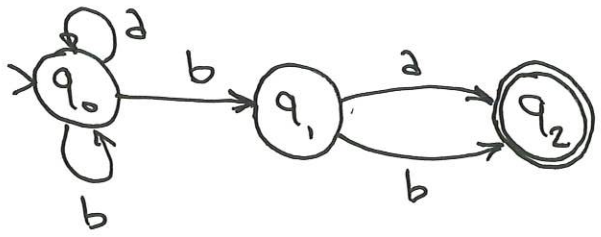
    - With ~~determa~~ non-deterministic automata each transition is from a single state to a set of states.

$$\delta(\sigma, q_i) = \{q_j, q_j', q_j'', \dots\} = Q'$$

$$\sigma \in \Sigma,\ q_i \in Q,\ Q' \subseteq Q$$

  - Example

Let $L = (a+b)^* b (a+b)$, $\Sigma = \{a, b\}$

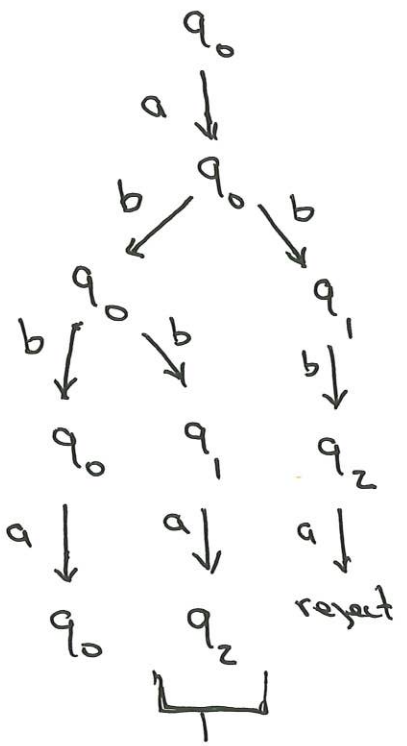


- Note: q_2 has no outgoing transitions. Interpret this as $\delta(\sigma, q_2) = \text{reject}$

NFA and ϵ-transitions

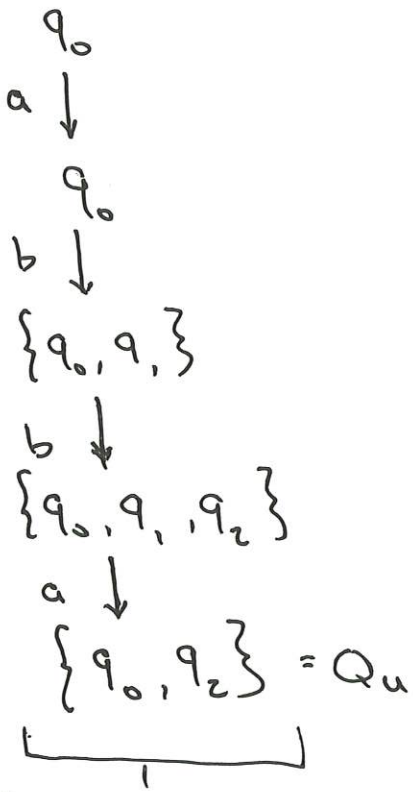
- Two ways to envision processing: Let $u = abba$

1 path at a time
(tree structure)



Since a path exists to q_2 → accept

all paths at once



Since $q_2 \in Q_u$, accept

(Using Q_u to represent the set of states M ends up in after processing string u.)

NFA and ε-transitions

- Acceptance criteria
  - Informally, if any path resulting from processing string u results in a final state, then accept.
  - Let $M = \{Q, \Sigma, q_0, F, \delta\}$
    - Note δ is no longer a transition function, but is a transition relation.
  - Let Q_u be the set of states an NFA ends in by processing string u.
  - Formally, $\delta(u, q_0) = Q_u$
    if Q_u contains a final (accepting) state, then $Q_u \cap F \neq \phi$ (ϕ = empty state)

$$\text{or } |Q_u \cap F| > 0$$

$$\text{or } |\delta(u, q_0) \cap F| > 0 \quad (\text{since } \delta(u, q_0) = Q_u)$$

> Acceptance criteria:
> NFA M accepts string u iff
> $|\delta(u, q_0) \cap F| > 0$

NFA AND ε-transitions

- Every NFA can be converted to a DFA
  - This means NFA's can accept only ~~DFA~~ regular languages, just like DFA's.
    - No computational power gained
    - convenience ~~to~~ may be gained
  - Basic process
    - Given NFA with states $Q = \{q_0, q_1, q_2 \cdots\}$
    - Create powerset Q^2:

$$Q^2 = \underbrace{\{\emptyset, \{q_0\}, \{q_0, q_1\}, \{q_0, q_1, q_2\} \cdots\}}_{\text{every subset of } Q}$$

    - Then create DFA with states Q^2, and δ created by mapping the δ from NFA.

[Examples worked in class.]

Let $L = (a+b)^* b[(a+b) + (a+b)(a+b)]$

ε-transitions

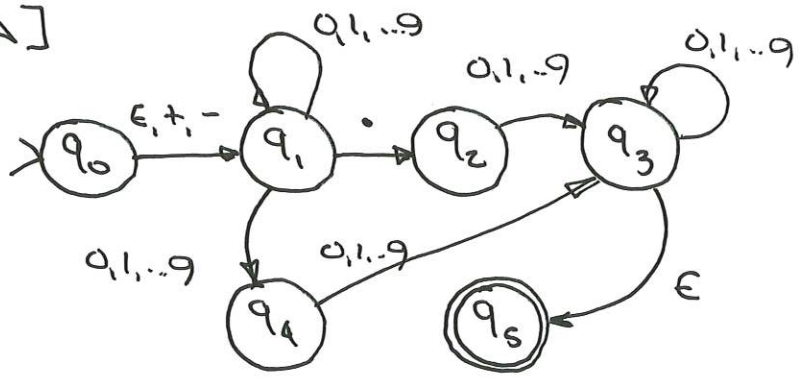
- Informal idea: performing a state transition without processing a symbol.
- Like with NFA, adds convenience, but NFA's with ε-transitions still only accept regular languages
- Goes hand in hand with non-determinism

Example[1]: Let L be the set of decimal numbers with an optional leading + or − sign.

$$\Sigma = \{+, -, \cdot, 0, 1, 2, \ldots, 9\}$$

[make DFA]

NFA M:



(1) Automata Theory, Languages, and Computation
Hopcroft, Motwani, Ullman

NFA and ϵ-transitions

ϵ-transitions and PDA's

- All "accepting empty stack"
  can be converted to "accept by final state"
  using $\delta(\epsilon, q_i, \$) = q_f \mid \$$, where $q_f \in F$

- consider PDA for palindromes

  $L_1 = uc u'$ where u' = reverse of u, $\Sigma = \{a, b, c\}$

  $L_2 = uu'$ where u' = reverse of u.