# COMP 3721

Tutorial 3

- NFA
- NFA = DFA = Regular Language

# NFA

- In a DFA,
  - *each* symbol read causes a transition to the next state, which is *completely* determined by the current state and current symbol (i.e., there is exactly one next state).
- In an NFA,
  - some state may have more than one outgoing edge labeled with the same symbol
  - some edges may be labeled with $e$, the empty word.

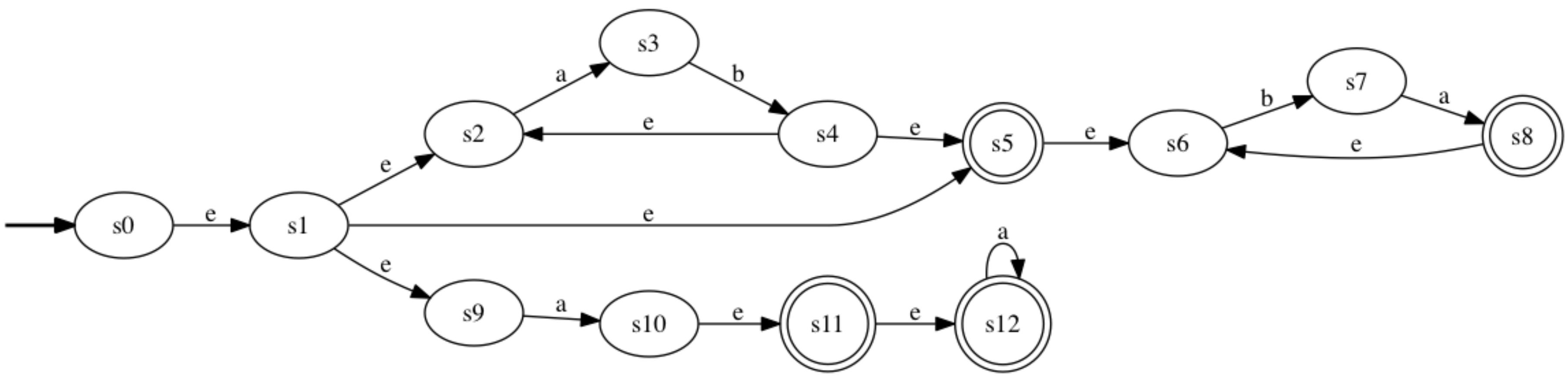
# NFA

1) Draw state diagrams for NFAs that accepts the following languages.
   a) $(ab)^*(ba)^* \cup aa^*$
   b) $((ab \cup aab)^*a^*)^*$

Hints: Use ideas from the proof which shows that every regular language is accepted by some NFA.

# NFA

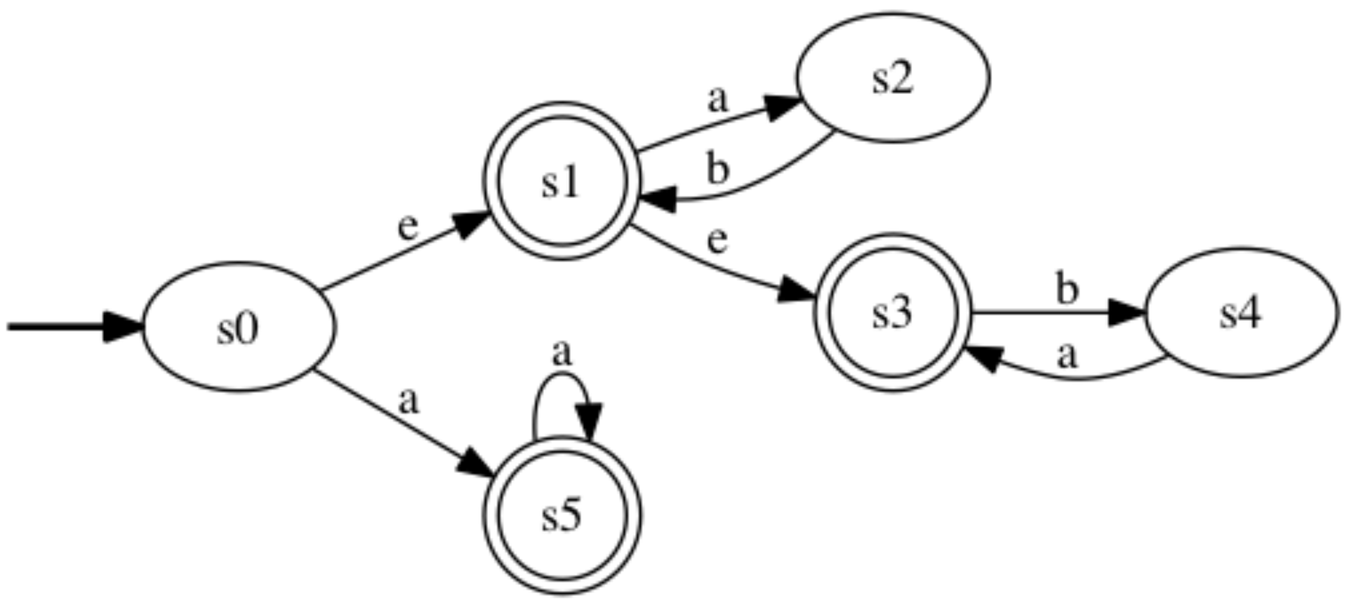
1) Draw state diagrams for NFAs that accepts the following languages.
   a) $(ab)^*(ba)^* \cup aa^*$



# NFA

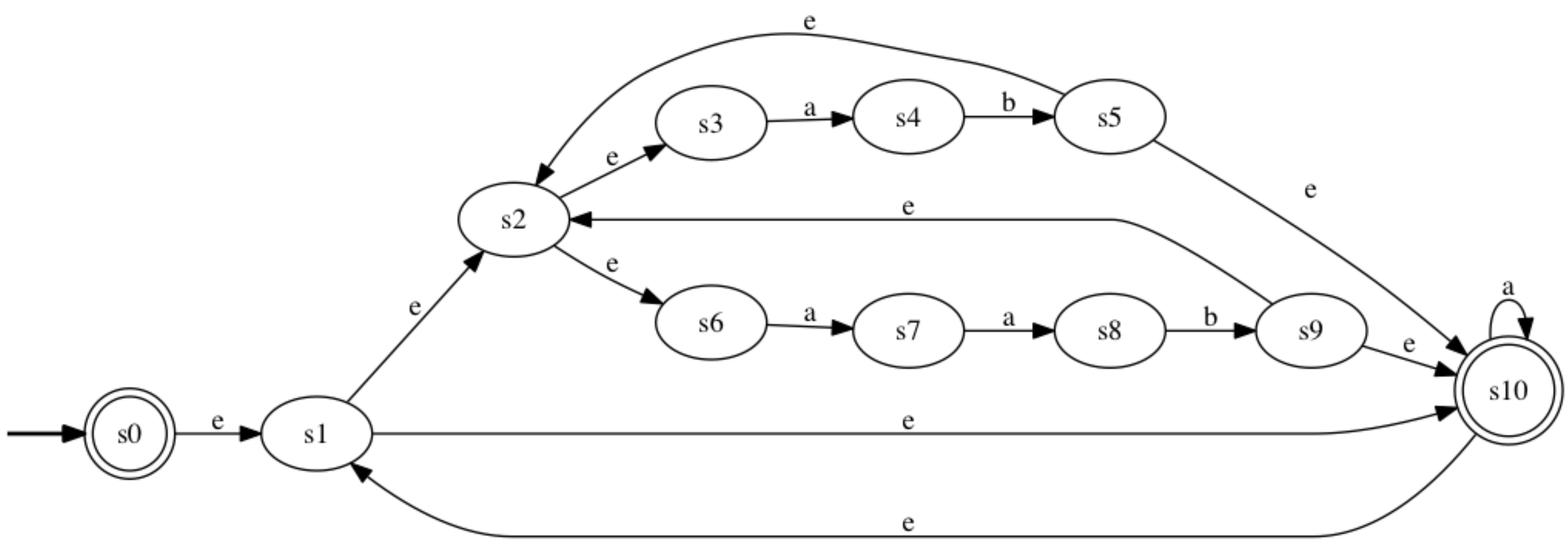
1) Draw state diagrams for NFAs that accepts the following languages.

   a) $(ab)^*(ba)^* \cup aa^*$



# NFA

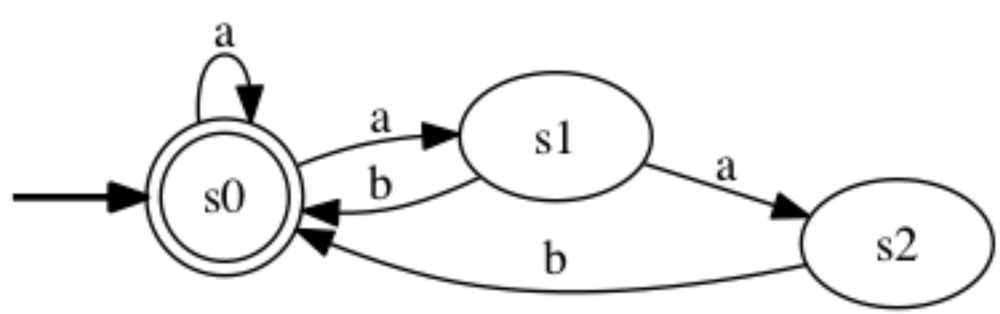
1) Draw state diagrams for NFAs that accepts the following languages.
   b) $((ab \cup aab)^* a^*)^*$



# NFA

1) Draw state diagrams for NFAs that accepts the following languages.

    b) $((ab \cup aab)^* a^*)^*$



# NFA = DFA

Given an NFA $M = \{K, \Sigma, \Delta, s, F\}$.

We want to construct an equivalent DFA that accepts the same language:

$M' = \{K', \Sigma, \delta', s', F'\}$

- $K' = 2^K$
- $s' = E(s)$
- $F' = \{Q \in K' \mid Q \cap F \neq \emptyset\}$
- for all $Q \in K'$, $\sigma \in \Sigma$,
  $\delta'(Q, \sigma) = \cup_{q \in Q}\{E(p) : (q, \sigma, p) \in \Delta\}$.

# NFA = DFA

2)

a) Find a simple NFA accepting $(ab \cup aab \cup aba)^*$ .

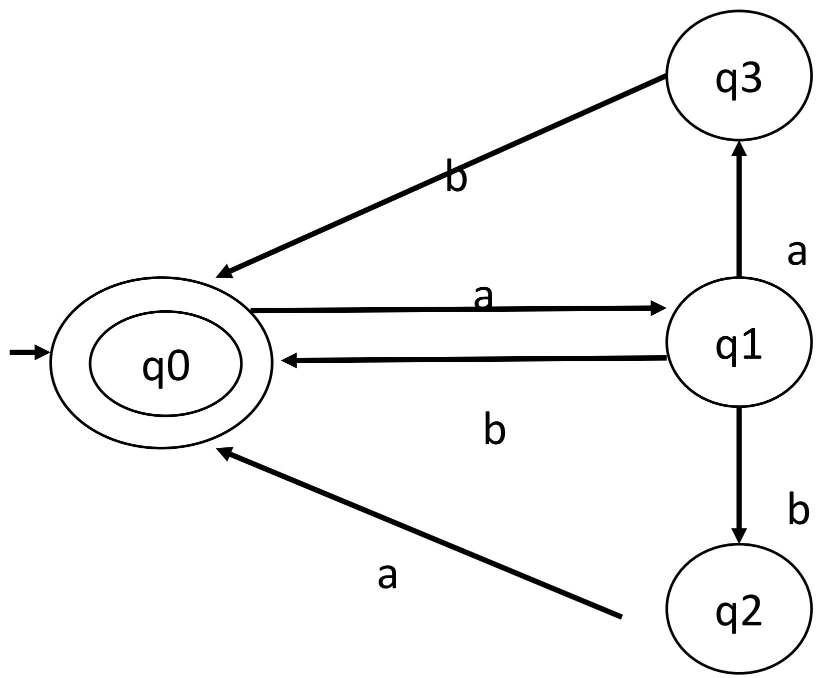
b) Convert the NFA of (a) to DFA.

# NFA = DFA

2)

a) Find a simple NFA accepting $(ab \cup aab \cup aba)^*$ .

b) Convert the NFA of (a) to DFA.



# NFA = DFA

2)

a) Find a simple NFA accepting $(ab \cup aab \cup aba)^*$.

b) Convert the NFA of (a) to DFA.

Then, the states in the corresponding DFA are found as follows:

| **States in the DFA Qi** | $\boldsymbol{\delta'(Qi, a)}$ | $\boldsymbol{\delta'(Qi, b)}$ |
|---|---|---|
| Q0=E(q0)={q0} | {q1} | Ø |
| Q1={q1} | {q3} | {q0,q2} |
| Q2={q3} | Ø | {q0} |
| Q3={q0,q2} | {q0,q1} | Ø |
| Q4={q0,q1} | {q1,q3} | {q0,q2} |
| Q5={q1,q3} | {q3} | {q0,q2} |
| Q6=Ø | Ø | Ø |

# NFA = DFA

2)

a) Find a simple NFA accepting $(ab \cup aab \cup aba)^*$ .
b) Convert the NFA of (a) to DFA.

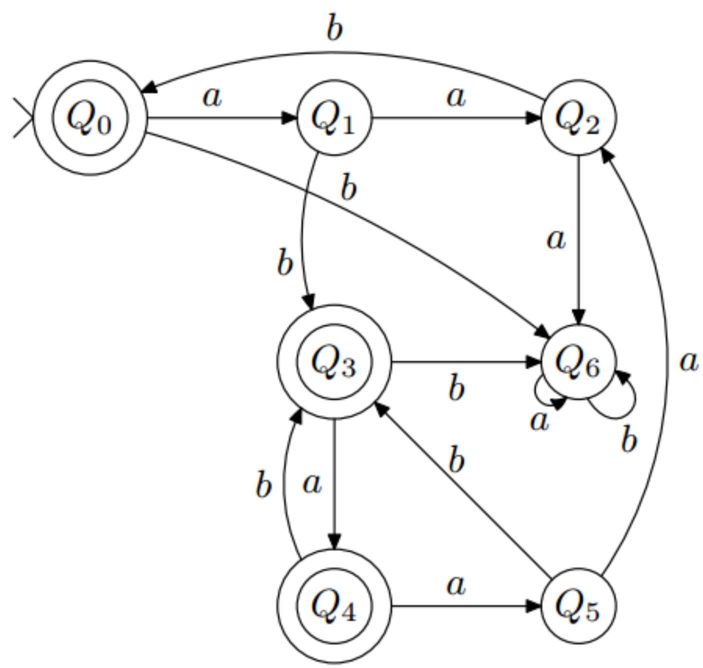
The start state is E(q0) = Q0.
The accepting states are those contains q0. {Q0, Q3, Q4}

# NFA = DFA

2)

a) Find a simple NFA accepting $(ab \cup aab \cup aba)^*$.

b) Convert the NFA of (a) to DFA.



# DFA = NFA = Regular Expression

3) Prove that if *L* is regular, then so is

$$Pref(L) = \{w : wu \in L \text{ for some string } u\}$$

Hints:

Construct NFA/DFA accepting *Pref(L)* from that accepting *L*

# DFA = NFA = Regular Expression

3) Prove that if *L* is regular, then so is

$$Pref(L) = \{w : wu \in L \text{ for some string } u\}$$

Proof: Since L is regular, it is accepted by some NFA.

M = {K, Σ, Δ, s, F}. Next we will construct a NFA M′ that accepts Pref(L). We let M′ = (K, Σ, Δ, s, F′) where

F′ = {q $\in$ K : in M, there is a path from q to f for some f $\in$ F}.

We claim that M′ accepts Pref(L) and hence Pref(L) is regular.

# DFA = NFA = Regular Expression

3) Prove that if *L* is regular, then so is

$$Pref(L) = \{w : wu \in L \text{ for some string } u\}$$

Proof(cont'd):
To see why M' accepts Pref(L) , consider any string w.
w $\in$ Pref(L) iff
wy $\in$ L for some y $\in$ Σ*, iff
wy is accepted by M, iff
for some q $\in$ K and f $\in$ F, $(s, wy) \vdash_M^* (q, y)) \vdash_M^* (f, e)$
iff for some q $\in$ F' $(s', w) \vdash_{M'} (q, e)$
iff w is accepted by M'