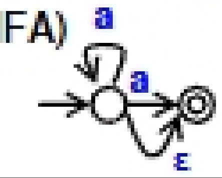
CMSC330

Finite Automata 2

Last Lecture

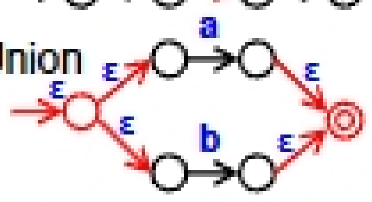
- Finite automata
  - Alphabet, states…
  - (Σ, Q, q_0, F, δ)
- Types
  - Deterministic (DFA)

  - Non-deterministic (NFA)

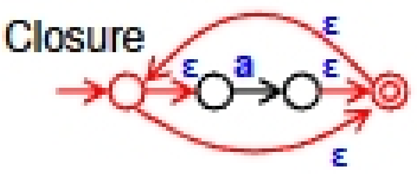


- Reducing RE to NFA
  - Concatenation

  - Union



  - Closure

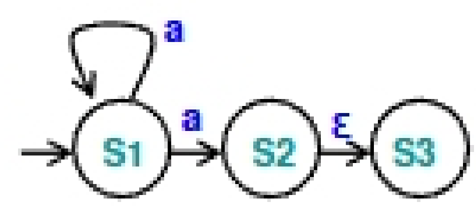


This Lecture

- Reducing NFA to DFA*
  - ε-closure
  - Subset algorithm
- Minimizing DFA*
  - Hopcroft reduction
- Complementing DFA
- Implementing DFA*

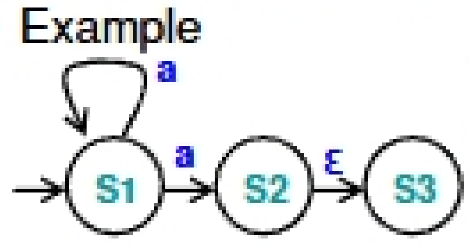
How NFA Works

- When NFA processes a string
  - NFA may be in several possible states
    - Multiple transitions with same label
    - ε-transitions
- Example
  - After processing "a"
    - NFA may be in states
      S1
      S2
      S3

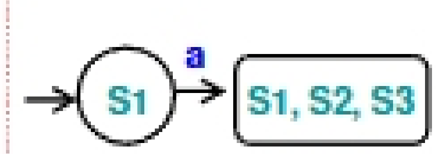


Reducing NFA to DFA

- NFA may be reduced to DFA
  - By explicitly tracking the set of NFA states
- Intuition
  - Build DFA where
    - Each DFA state represents a set of NFA states
- Example



NFA



DFA

Reducing NFA to DFA (cont.)

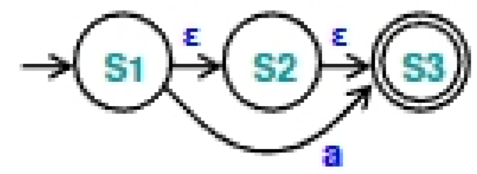
- Reduction applied using the subset algorithm
  - DFA state is a subset of set of all NFA states
- Algorithm
  - Input
    - NFA (Σ, Q, q_0, F_n, δ)
  - Output
    - DFA (Σ, R, r_0, F_d, δ)
  - Using
    - ε-closure(p)
    - move(p, a)

ε-transitions and ε-closure

- We say $p \xrightarrow{\varepsilon} q$
  - If it is possible to go from state p to state q by taking only ε-transitions
  - If $\exists\, p, p_1, p_2, \ldots p_n, q \in Q$ such that
    - $\{p,\varepsilon,p_1\} \in \delta, \{p_1,\varepsilon,p_2\} \in \delta, \ldots, \{p_n,\varepsilon,q\} \in \delta$
- ε-closure(p)
  - Set of states reachable from p using ε-transitions alone
    - Set of states q such that $p \xrightarrow{\varepsilon} q$
    - ε-closure(p) = $\{q \mid p \xrightarrow{\varepsilon} q\}$
  - Note
    - ε-closure(p) always includes p
    - ε-closure( ) may be applied to set of states (take union)

ε-closure: Example 1

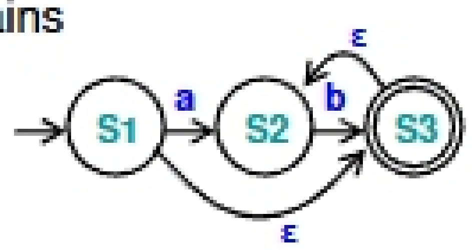
- Following NFA contains
  - $S1 \xrightarrow{\varepsilon} S2$
  - $S2 \xrightarrow{\varepsilon} S3$
  - $S1 \rightarrow S3$



- ε-closures
  - ε-closure(S1) = { S1, S2, S3 }
  - ε-closure(S2) = { S2, S3 }
  - ε-closure(S3) = { S3 }
  - ε-closure({ S1, S2 }) = { S1, S2, S3 } ∪ { S2, S3 }

ε-closure: Example 2

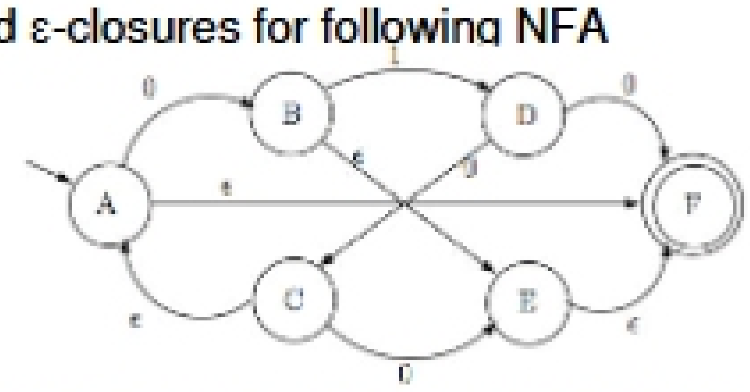
- Following NFA contains
  - $S1 \xrightarrow{\varepsilon} S3$
  - $S3 \xrightarrow{\varepsilon} S2$
  - $S1 \xrightarrow{\varepsilon} S2$



- ε-closures
  - ε-closure(S1) = { S1, S2, S3 }
  - ε-closure(S2) = { S2 }
  - ε-closure(S3) = { S2, S3 }
  - ε-closure({ S2,S3 }) = { S2 } ∪ { S2, S3 }

ε-closure: Practice

- Find ε-closures for following NFA



- Find ε-closures for the NFA you construct for
  - The regular expression (0|1*)111(0*|1)

Calculating move(p,a)

- move(p,a)
  - Set of states reachable from p using exactly one transition on a
    - Set of states q such that $\{p, a, q\} \in \delta$
    - move(p,a) = $\{q \mid \{p, a, q\} \in \delta\}$
  - Note move(p,a) may be empty Ø
    - If no transition from p with label a

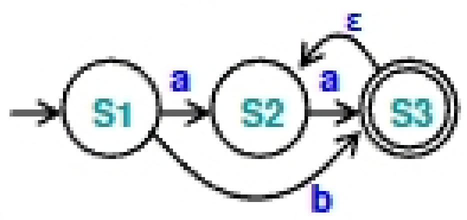
move(p,a) : Example 1

- Following NFA
  - Σ = { a, b }

- Move
  - move(S1, a) = { S2, S3 }
  - move(S1, b) = Ø
  - move(S2, a) = Ø
  - move(S2, b) = { S3 }
  - move(S3, a) = Ø
  - move(S3, b) = Ø

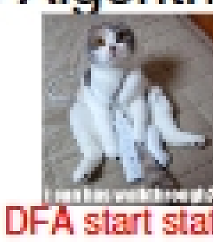
move(p,a) : Example 2

- Following NFA
  - Σ = { a, b }



- Move
  - move(S1, a) = { S2 }
  - move(S1, b) = { S3 }
  - move(S2, a) = { S3 }
  - move(S2, b) = Ø
  - move(S3, a) = Ø
  - move(S3, b) = Ø

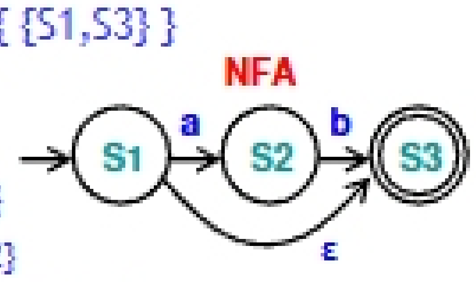
NFA → DFA Reduction Algorithm



- Input NFA (Σ, Q, q_0, F_n, δ),
  Output DFA (Σ, R, r_0, F_d, δ)
- Algorithm
  - Let r_0 = ε-closure(q_0), add it to R // DFA start state
  - While ∃ an unmarked state r ∈ R // process DFA state r
    - Mark r // each state visited once
    - For each a ∈ Σ // for each letter a
      - Let S = {s | q ∈ r & move(q,a) = s} // states reached via a
      - Let e = ε-closure(S) // states reached via ε
      - If e ∉ R // if state e is new
        Let R = e ∪ R // add e to R (unmarked)
      - Let δ = δ ∪ {r, a, e} // add transition r→e
  - Let F_d = {r | ∃ s ∈ r with s ∈ F_n} // final if include state in F_n

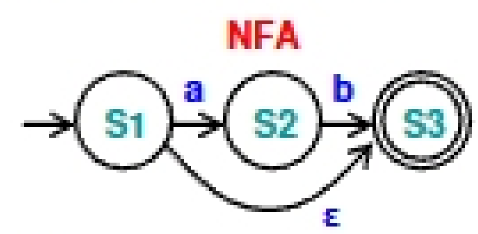
NFA → DFA Example 1

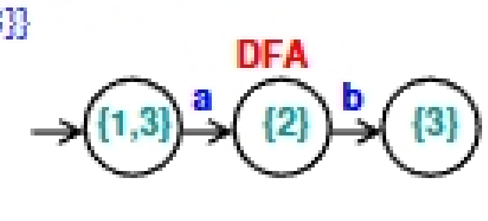
- Start = ε-closure(S1) = { {S1,S3} }
- R = { {S1,S3} }
- r ∈ R = {S1,S3}
- Move({S1,S3},a}) = {S2}
  - e = ε-closure({S2}) = {S2}
  - R = R ∪ {S2} = { {S1,S3}, {S2} }
  - δ = δ ∪ {{S1,S3}, a, {S2}}
- Move({S1,S3},b}) = Ø



NFA → DFA Example 1 (cont.)

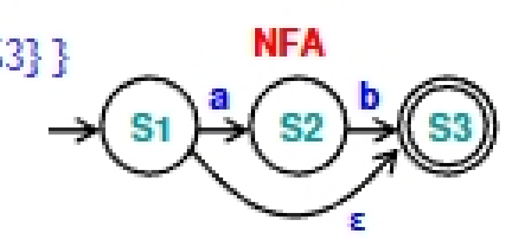
- R = { {S1,S3}, {S2} }
- r ∈ R = {S2}
- Move({S2},a} = Ø
- Move({S2},b} = {S3}
  - e = ε-closure({S3}) = {S3}
  - R = R ∪ {S3} = { {S1,S3}, {S2}, {S3} }
  - δ = δ ∪ {{S2}, b, {S3}}

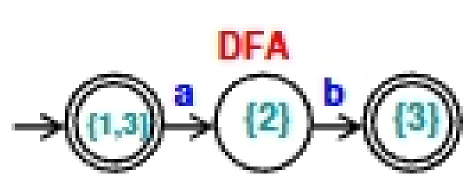




NFA → DFA Example 1 (cont.)

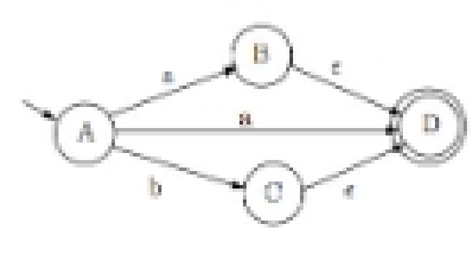
- R = { {S1,S3}, {S2}, {S3} }
- r ∈ R = {S3}
- Move({S3},a} = Ø
- Move({S3},b} = Ø
- F_d = {{S1,S3}, {S3}}
  - Since S3 ∈ F_n
- Done!



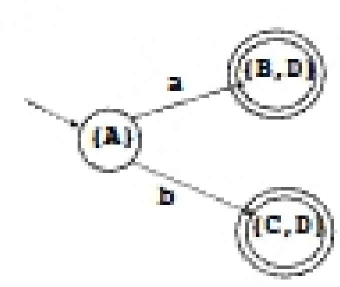


NFA → DFA Example 2

- NFA



- DFA



R = { {A}, {B,D}, {C,D} }