CS 6041
Theory of Computation

Nondeterministic finite automata

Kun Suo

Computer Science, Kennesaw State University

https://kevinsuo.github.io/

Design a NFA for a language

- Step 1: list all possible states
- Step 2: draw all the transitions between the states
- Step 3: add start and accept states

Some practices

- L_1 = {Set of all strings that end with 0}, $\Sigma = \{0,1\}$;

Can anyone draw the NFA?

Some practices

- L_1 = {Set of all strings that end with 0}, Σ = {0,1};

Can anyone draw the NFA?

q1: all the strings
q2: last letter is 0

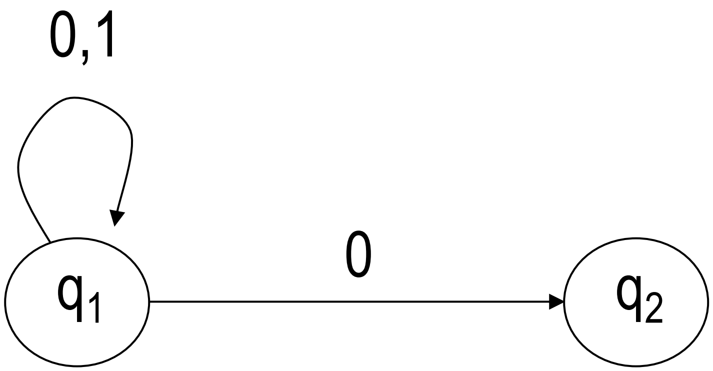
NFA of L_1

Some practices

- L_1 = {Set of all strings that end with 0}, Σ = {0,1};

Can anyone draw the NFA?

q1: all the strings
q2: last letter is 0



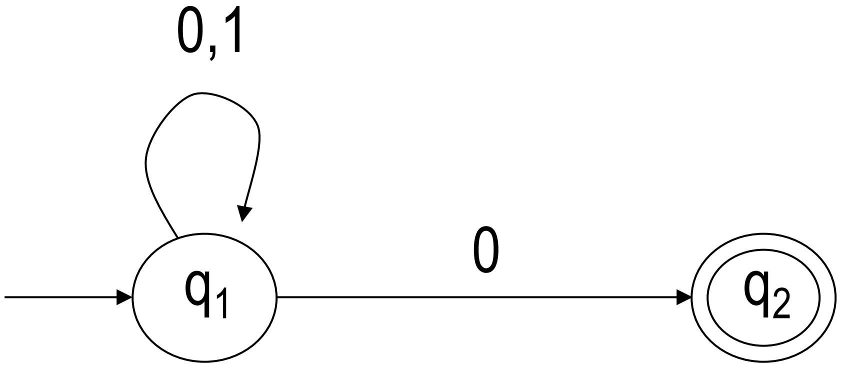
NFA of L_1

Some practices

- L_1 = {Set of all strings that end with 0}, Σ = {0,1};

Can anyone draw the NFA?

q1: all the strings
q2: last letter is 0



NFA of L_1

Some practices

- L_2 = {Set of all strings that start with 0}, Σ = {0,1};

Can anyone draw the NFA?

Some practices

- L_2 = {Set of all strings that start with 0}, Σ = {0,1};

Can anyone draw the NFA?

q1: empty string
q2: first letter is 0

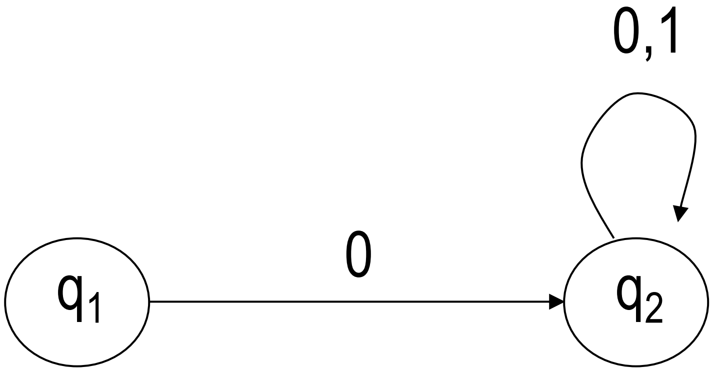
NFA of L_2

Some practices

- L_2 = {Set of all strings that start with 0}, Σ = {0,1};

Can anyone draw the NFA?

q1: empty string
q2: first letter is 0



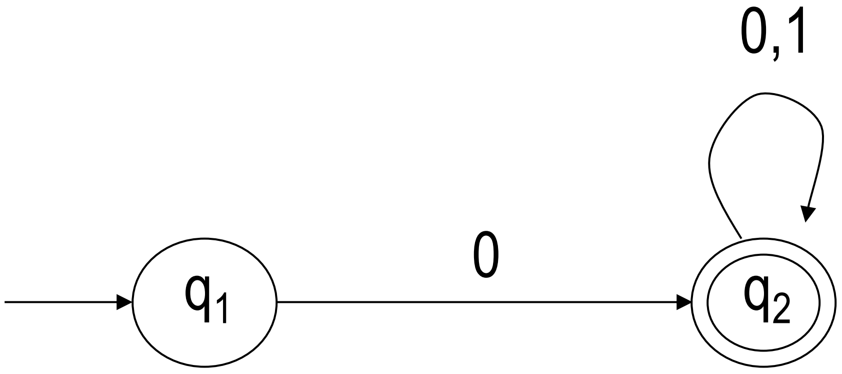
NFA of L_2

Some practices

- L_2 = {Set of all strings that start with 0}, Σ = {0,1};

Can anyone draw the NFA?

q1: empty string
q2: first letter is 0



NFA of L_2

Some practices

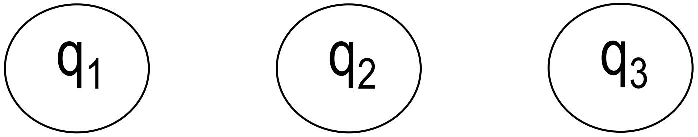
- L_3 = {Set of all strings that length is 2}, Σ = {0,1};

Can anyone draw the NFA?

Some practices

- L_3 = {Set of all strings that length is 2}, Σ = {0,1};

Can anyone draw the NFA?



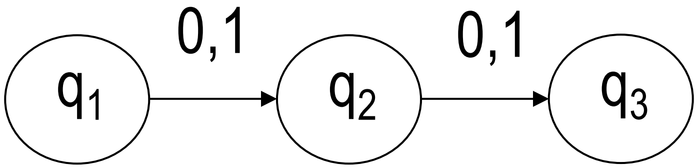
q1: strings that length is 0
q2: strings that length is 1
q3: strings that length is 2

NFA of L_3

Some practices

- L_3 = {Set of all strings that length is 2}, Σ = {0,1};

Can anyone draw the NFA?



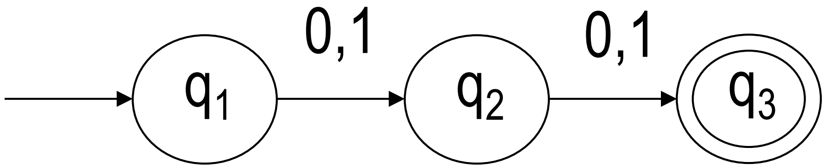
q1: strings that length is 0
q2: strings that length is 1
q3: strings that length is 2

NFA of L_3

Some practices

- L_3 = {Set of all strings that length is 2}, Σ = {0,1};

Can anyone draw the NFA?



q1: strings that length is 0
q2: strings that length is 1
q3: strings that length is 2

NFA of L_3

Some practices

- L_4 = {Set of all strings that contain ‘0’}, Σ = {0,1};

Can anyone draw the NFA?

Some practices

- L_4 = {Set of all strings that contain ‘0’}, $\Sigma = \{0,1\}$;

Can anyone draw the NFA?

q1: all strings
q2: strings that contain 0

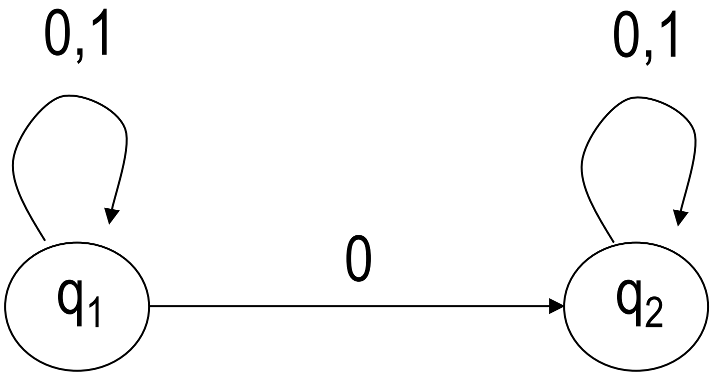
NFA of L_4

Some practices

- L_4 = {Set of all strings that contain ‘0’}, Σ = {0,1};

Can anyone draw the NFA?

q1: all strings
q2: strings that contain 0



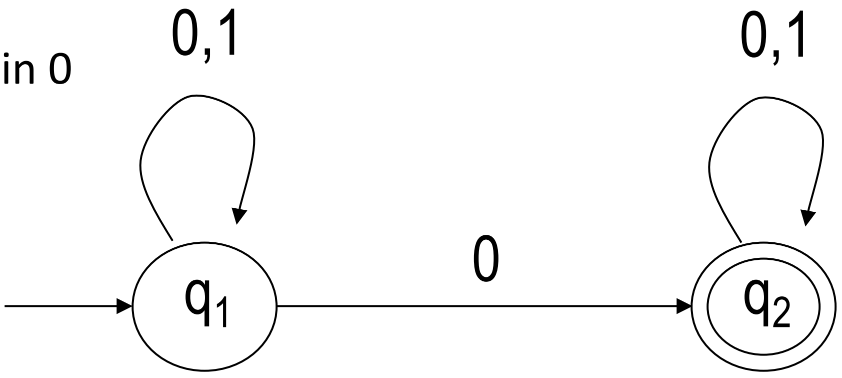
NFA of L_4

Some practices

- L_4 = {Set of all strings that contain ‘0’}, Σ = {0,1};

Can anyone draw the NFA?

q1: all strings
q2: strings that contain 0



NFA of L_4

Some practices

- L_5 = {Set of all strings that starts with ‘10’}, Σ = {0,1};

Can anyone draw the NFA?

NFA of L_5

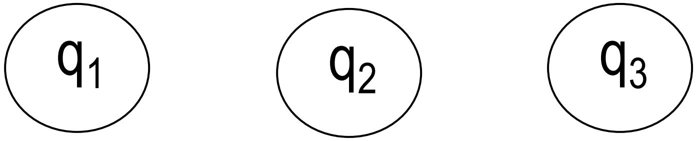
Some practices

- L_5 = {Set of all strings that starts with ‘10’}, Σ = {0,1};

Can anyone draw the NFA?

q1: all strings
q2: strings that start with 1
q3: strings that start with 10

NFA of L_5



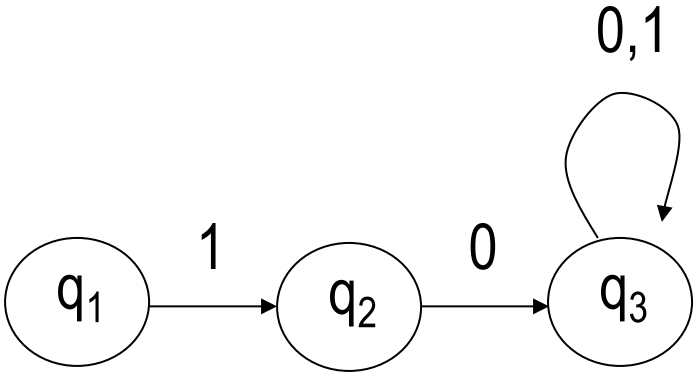
Some practices

- L_5 = {Set of all strings that starts with '10'}, Σ = {0,1};

Can anyone draw the NFA?

q1: all strings
q2: strings that start with 1
q3: strings that start with 10

NFA of L_5



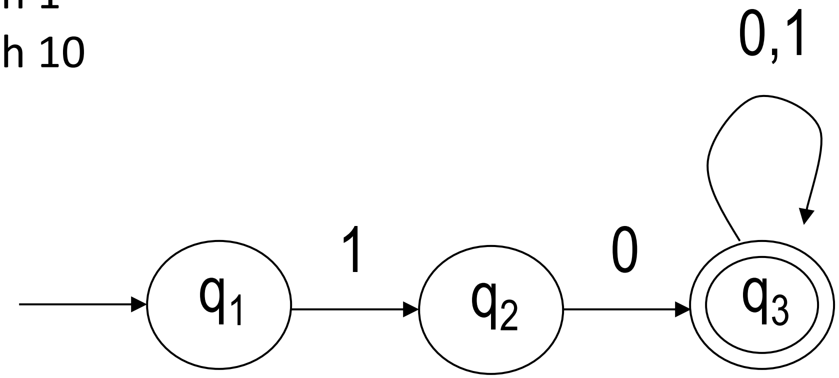
Some practices

- L_5 = {Set of all strings that starts with ‘10’}, Σ = {0,1};

Can anyone draw the NFA?

q1: all strings
q2: strings that start with 1
q3: strings that start with 10

NFA of L_5



Some practices

- L_6 = {Set of all strings that contain ‘01’}, Σ = {0,1};

Can anyone draw the NFA?

NFA of L_6

Some practices

- L_6 = {Set of all strings that contain ‘01’}, Σ = {0,1};

Can anyone draw the NFA?

q1: all strings
q2: strings that contain 0
q3: strings that contain 01

NFA of L_6

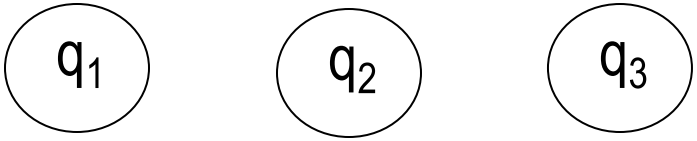
Some practices

- L_6 = {Set of all strings that contain '01'}, Σ = {0,1};

Can anyone draw the NFA?

q1: all strings
q2: strings that contain 0
q3: strings that contain 01

NFA of L_6



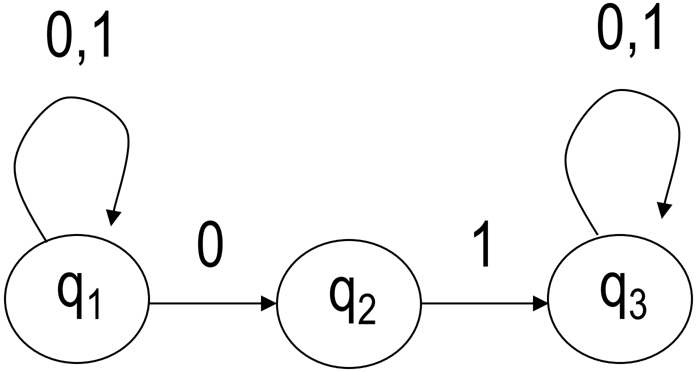
Some practices

- L_6 = {Set of all strings that contain ‘01’}, Σ = {0,1};

Can anyone draw the NFA?

q1: all strings
q2: strings that contain 0
q3: strings that contain 01

NFA of L_6



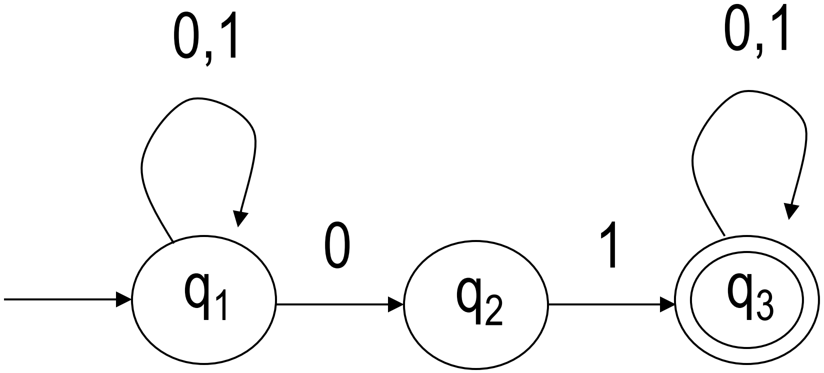
Some practices

- L_6 = {Set of all strings that contain '01'}, Σ = {0,1};

Can anyone draw the NFA?

q1: all strings
q2: strings that contain 0
q3: strings that contain 01

NFA of L_6



Some practices

- L_7 = {Set of all strings that end with '11'}, Σ = {0,1};

Can anyone draw the NFA?

NFA of L_7

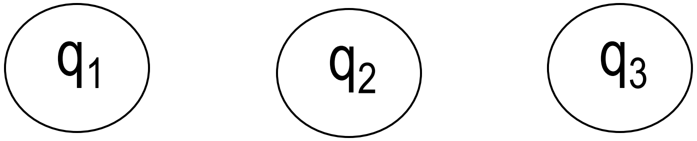
Some practices

- L_7 = {Set of all strings that end with ‘11’}, Σ = {0,1};

Can anyone draw the NFA?

q1: all strings
q2: strings that end with 1
q3: strings that end with 11

NFA of L_7



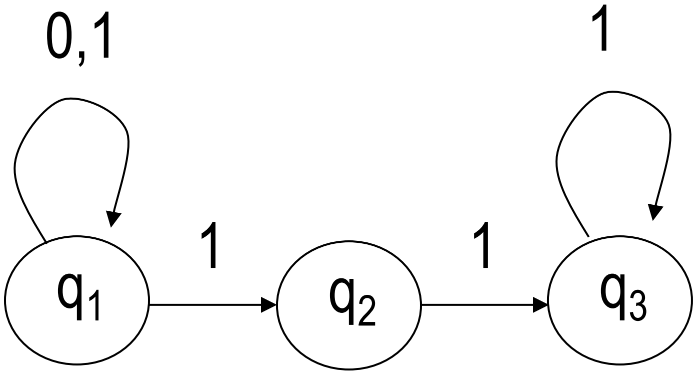
Some practices

- L_7 = {Set of all strings that end with '11'}, Σ = {0,1};

Can anyone draw the NFA?

q1: all strings
q2: strings that end with 1
q3: strings that end with 11

NFA of L_7



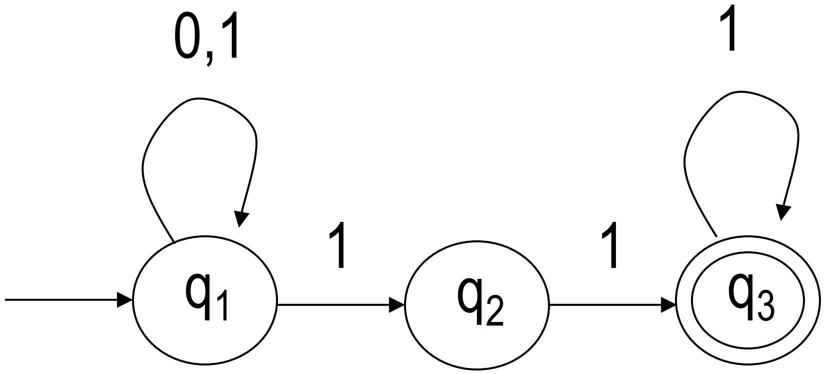
Some practices

- L_7 = {Set of all strings that end with ‘11’}, Σ = {0,1};

Can anyone draw the NFA?

q1: all strings
q2: strings that end with 1
q3: strings that end with 11

NFA of L_7



Equivalence of NFAs and DFAs

- Definition of equivalence
  - two machines are ***equivalent*** if they recognize the same language

Equivalence of NFAs and DFAs

- Theorem: Every nondeterministic finite automaton has an equivalent deterministic finite automaton.

- Proof idea:

  For given NFA, build an equivalent DFA

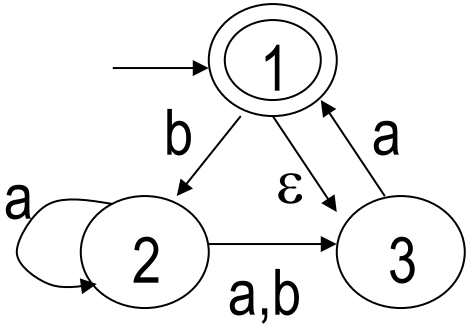
  Use DFA to simulate NFA

  - DFA records all branches of NFA
  - If NFA has k states, then DFA have totally 2^k states (subset of k states in NFA)

Equivalence of NFAs and DFAs

- Closure E(R) on ε

  Collection of states that can be reached from members of R by going only along ε arrows, including the members of R themselves.



E({1}) = {1,3}

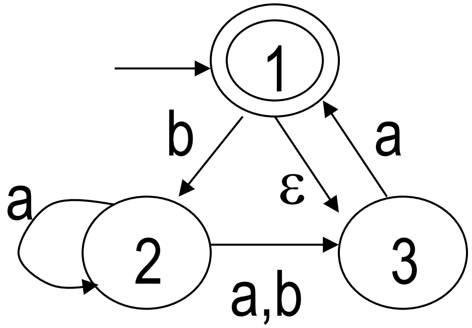
Equivalence of NFAs and DFAs

- Step 1: Draw all the states in DFAs
- Step 2: Define the transitions in DFAs based on the NFAs
- Step 3: Define the start state and accept state in DFAs
- Step 4: Remove all inaccessible states

Example

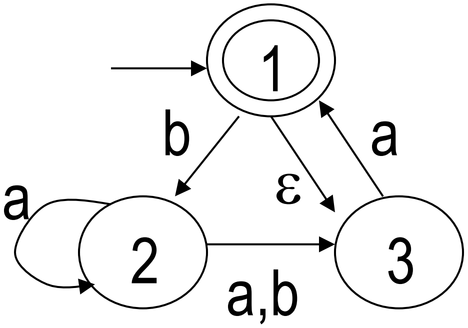
- NFA $N_4=(\{1,2,3\},\{a,b\},\delta,1,\{1\})$

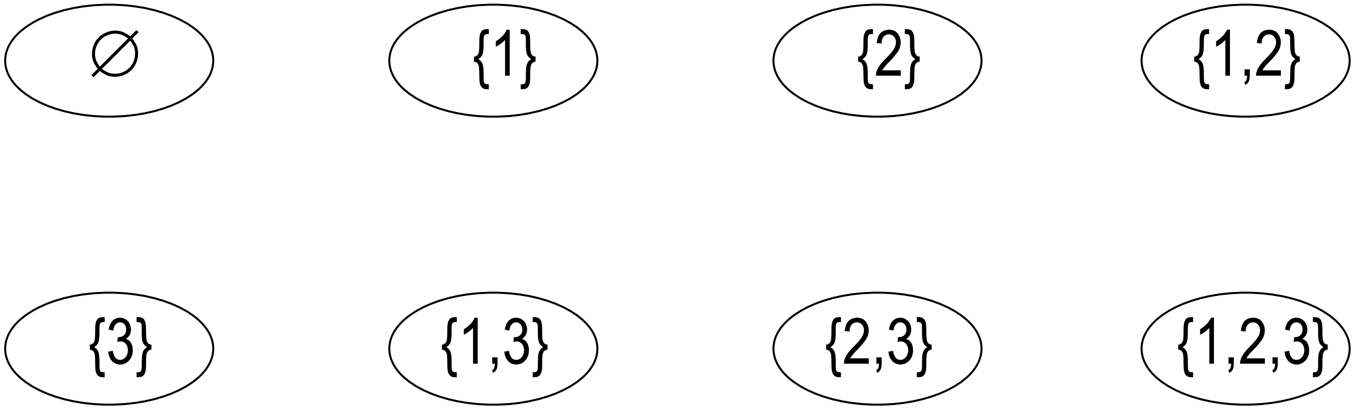
  What is its equivalent DFA?



Example

- List all subset states



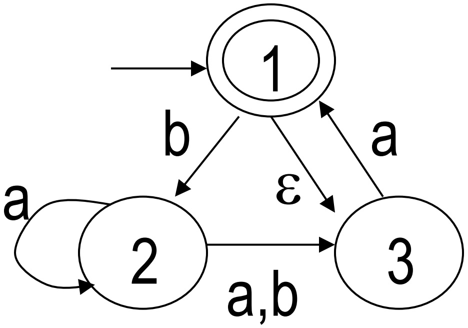


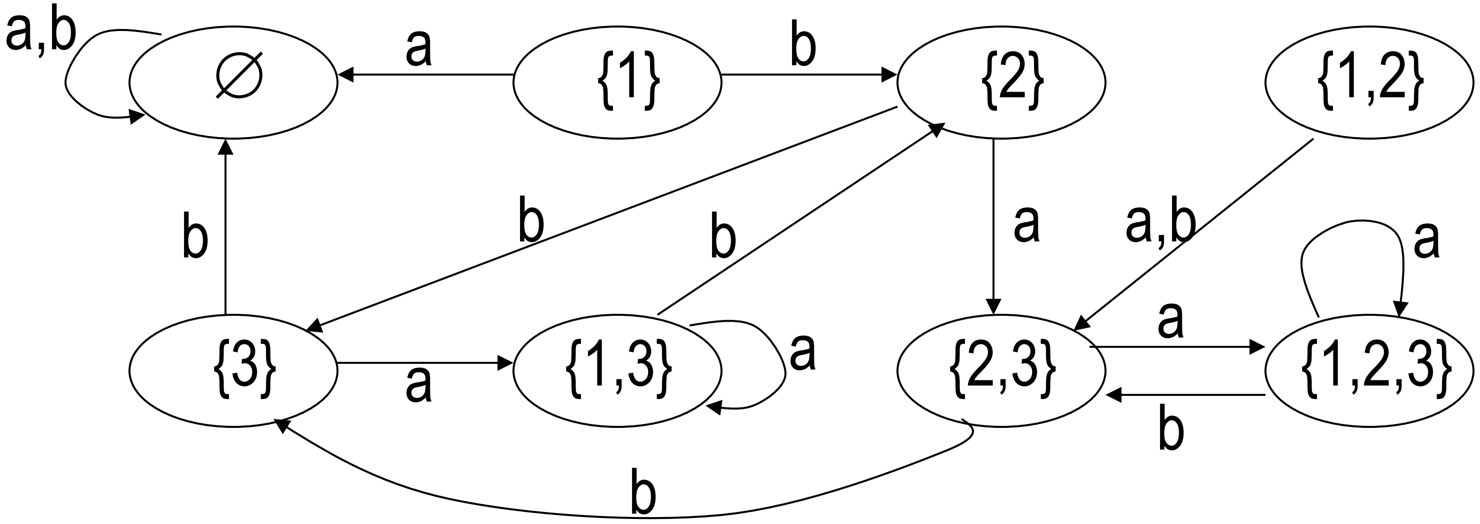
∅ {1} {2} {1,2}

{3} {1,3} {2,3} {1,2,3}

Example

- Add transitions



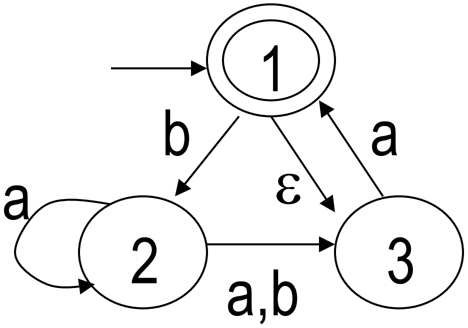


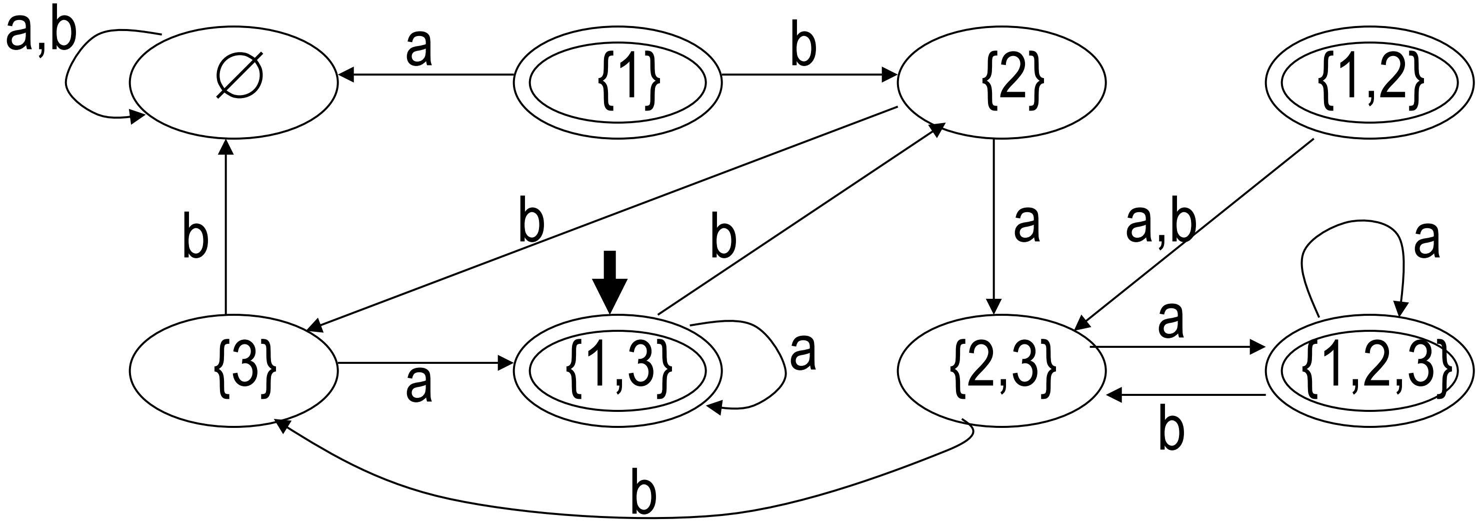
Example

Closure E(R) on ε

Collection of states that can be reached from members of R by going only along ε arrows, including the members of R themselves.

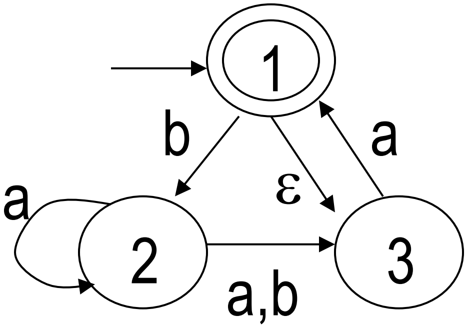
- Start state: E({1}) = {1,3}
- Accept state: all states with 1

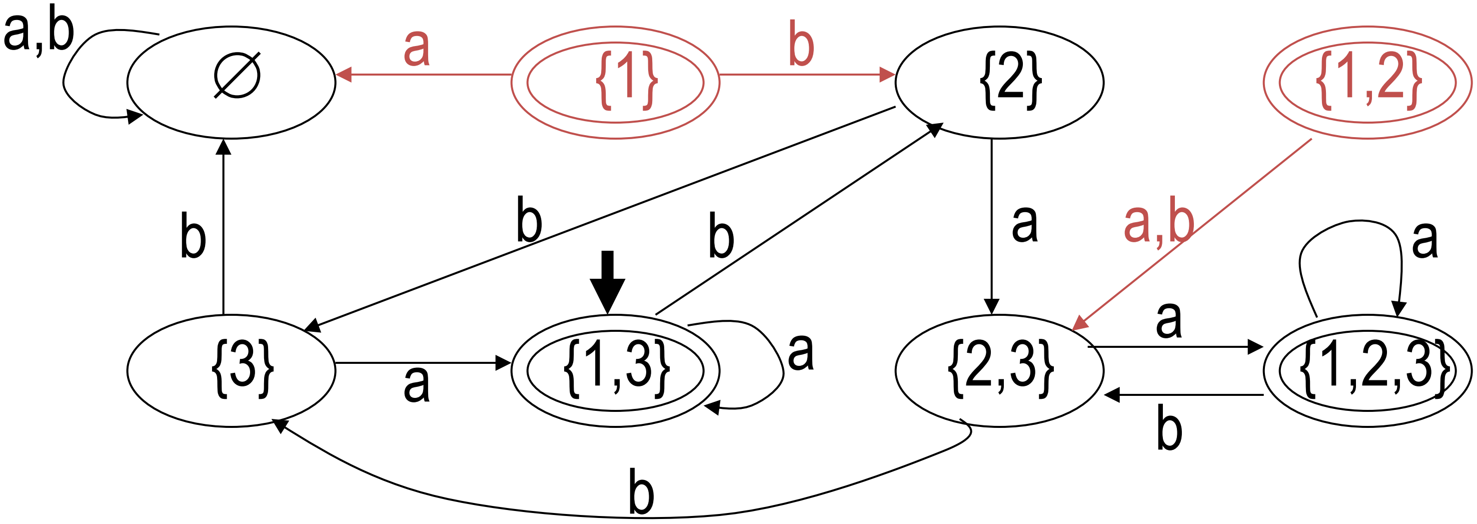




Example

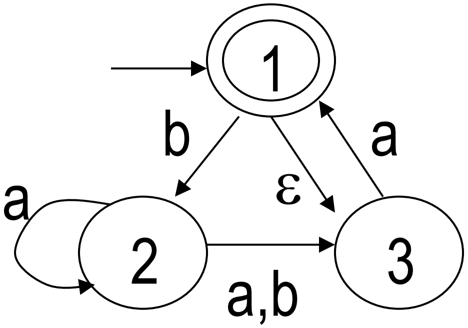
- Remove inaccessible state

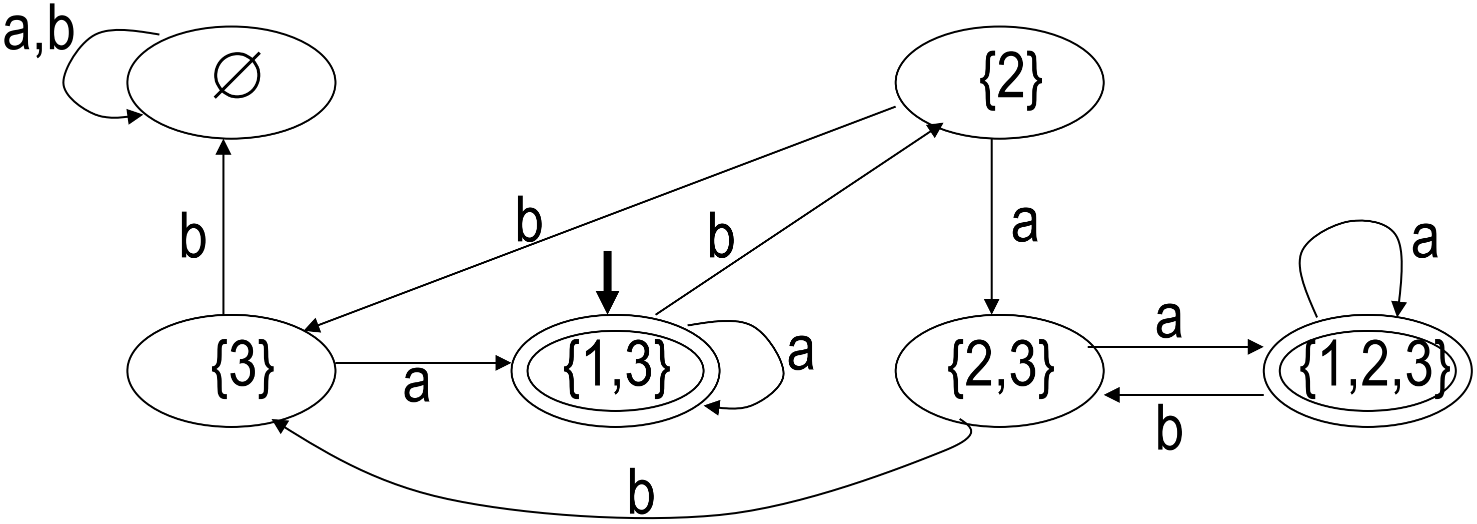




Example

- Remove inaccessible state
  - {1}, {1,2}





Equivalence of NFAs and DFAs

- Theorem: Every NFA has an equivalent DFA.
- Proof:

let NFA N = $(Q,\Sigma,\delta, q_0, F)$,

build DFA $M=(Q',\Sigma,\delta',q_0',F')$, $L(M)=L(N)$.

$E(R)=\{q \mid$ reach q from R by going along $\varepsilon\}$

let $Q'=P(Q)$. // 2^k subsets of Q, k=size of Q.

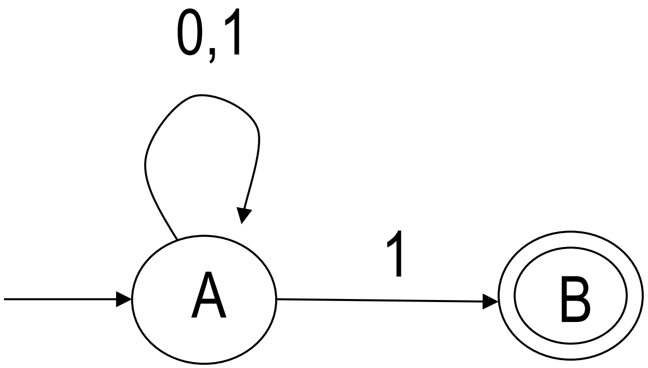
$q_0'=E(\{q_0\})$. // Closure q_0 on transition ε

$F'=\{R\in Q' \mid R\cap F\neq\varnothing\}$. // F' contains accept states in F

$\delta'(R,a)=U_{r\in R}E(\delta(r,a))$, for $R\in Q'$ and $a\in\Sigma$. // δ': all states that can be reached by going along ε arrows after every step

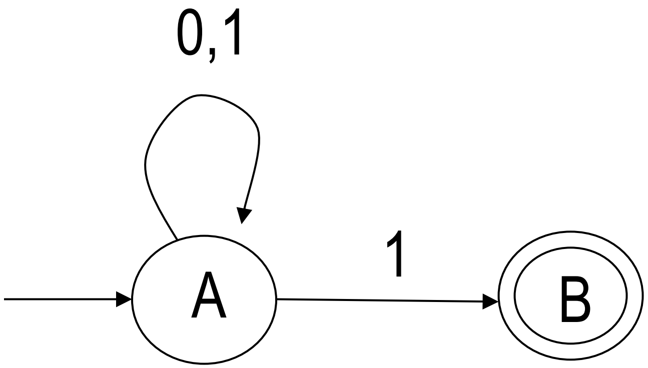
Practices: Conversion of NFA to DFA

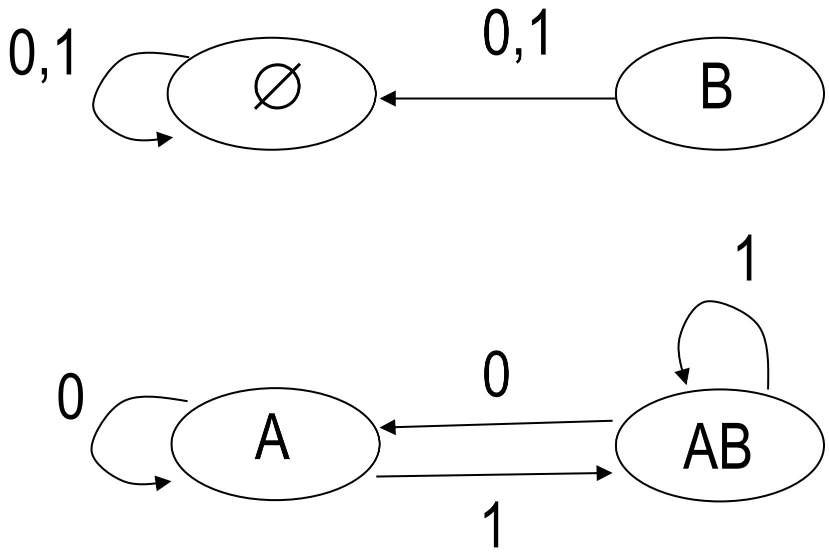
- L = {Set of all string that ends with ‘1’}, Σ = {0,1};



Practices: Conversion of NFA to DFA

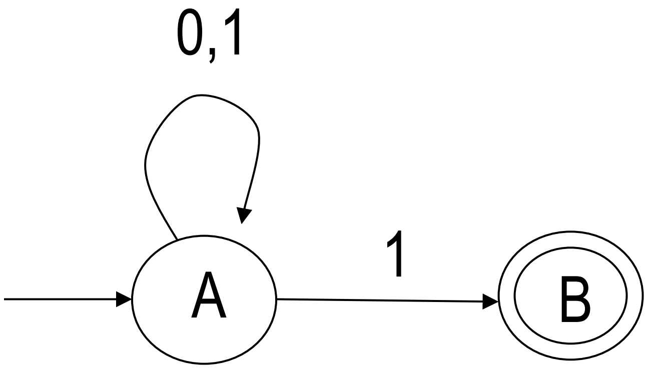
- L = {Set of all string that ends with ‘1’}, Σ = {0,1};

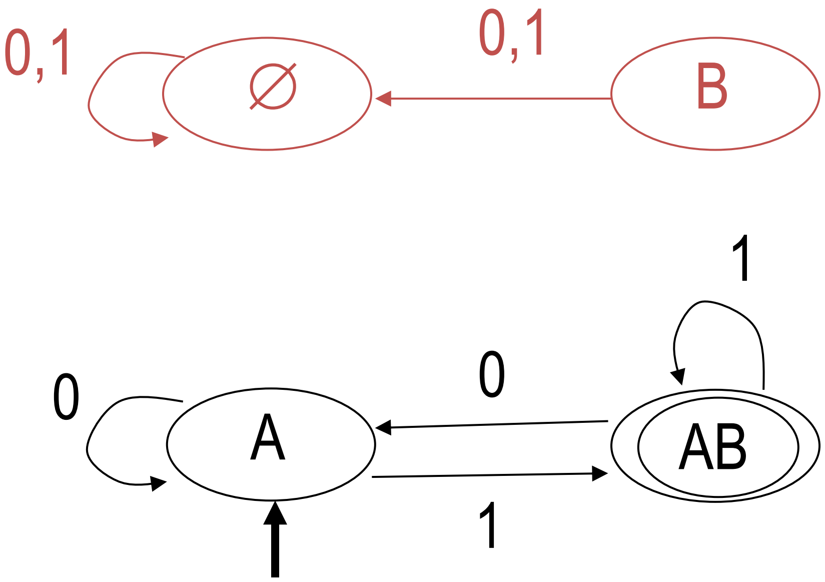




Practices: Conversion of NFA to DFA

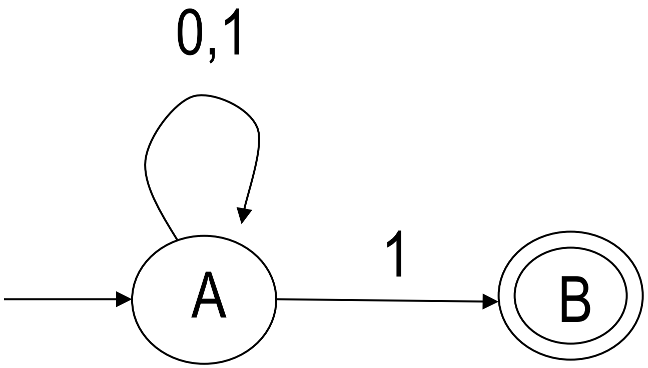
- L = {Set of all string that ends with ‘1’}, Σ = {0,1};

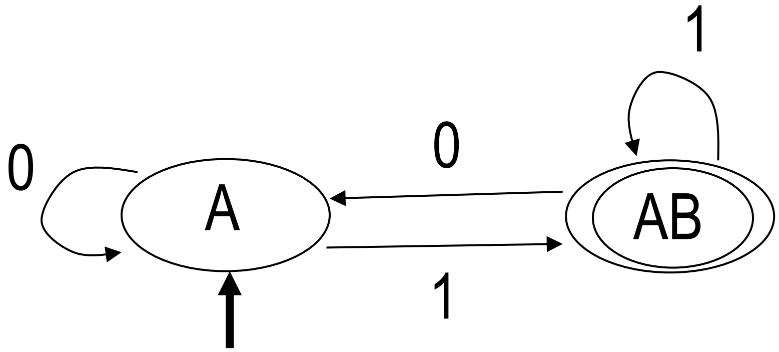




Practices: Conversion of NFA to DFA

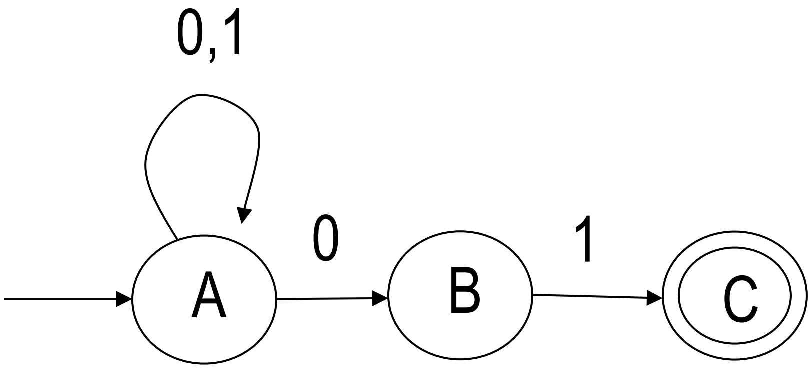
- L = {Set of all string that ends with ‘1’}, Σ = {0,1};



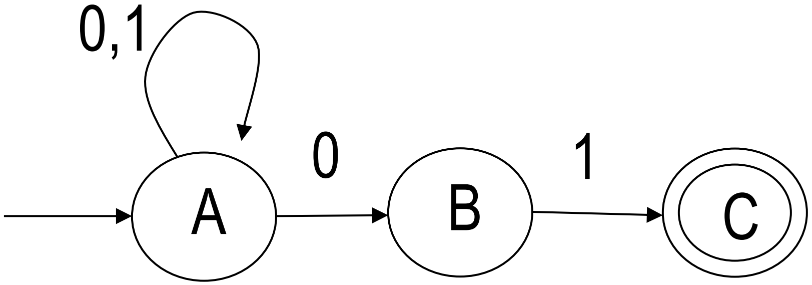


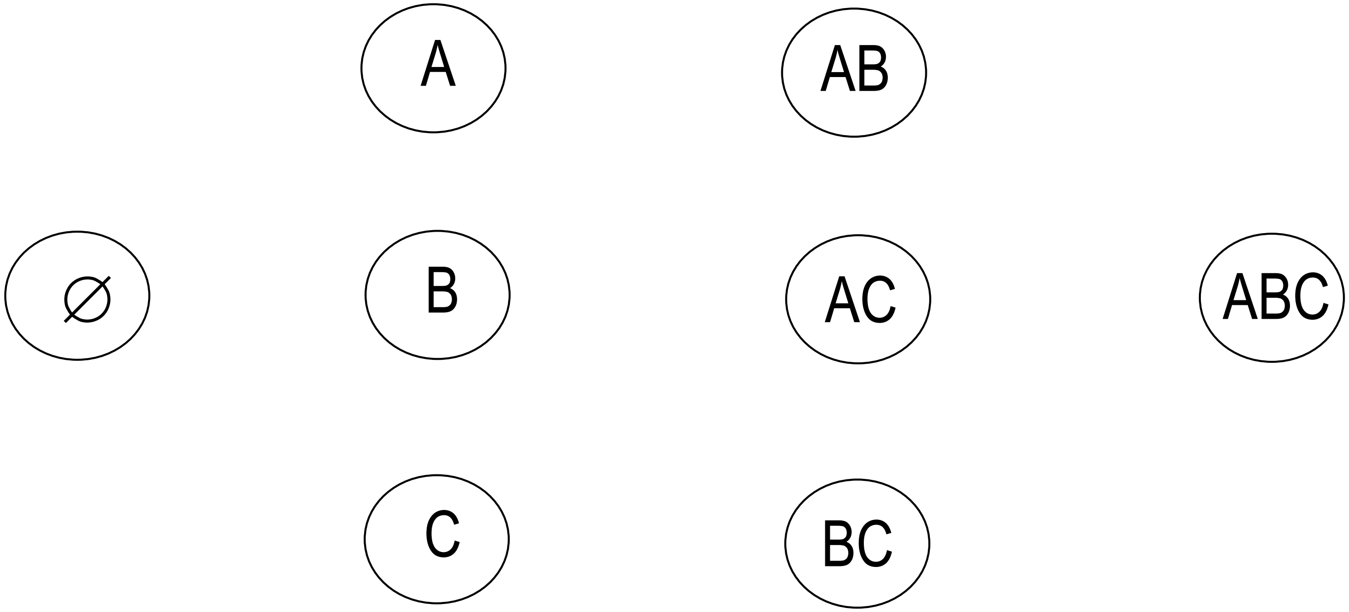
Practices: Conversion of NFA to DFA

- L = {Set of all string that ends with ‘01’}, Σ = {0,1};

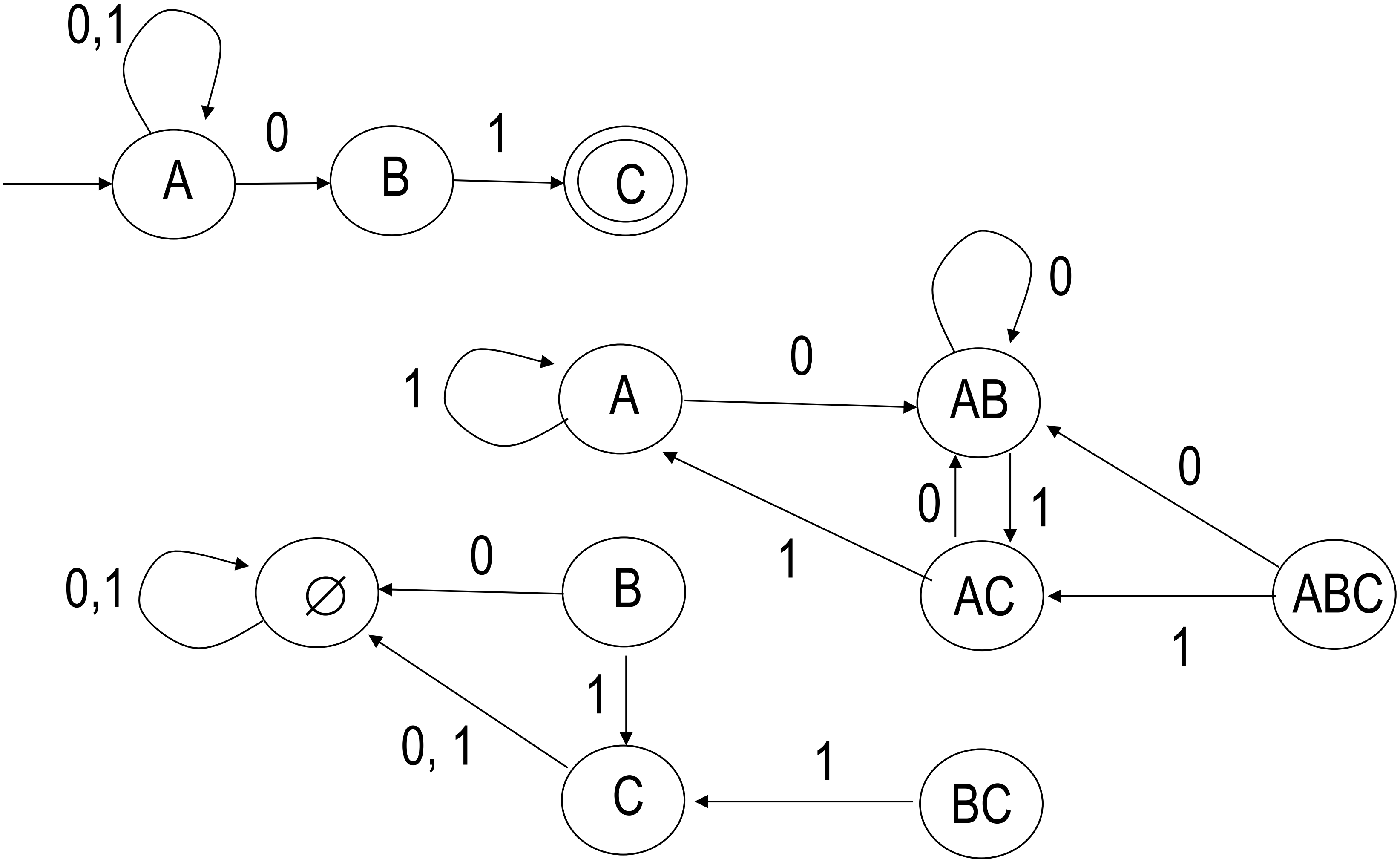


Practices: Conversion of NFA to DFA

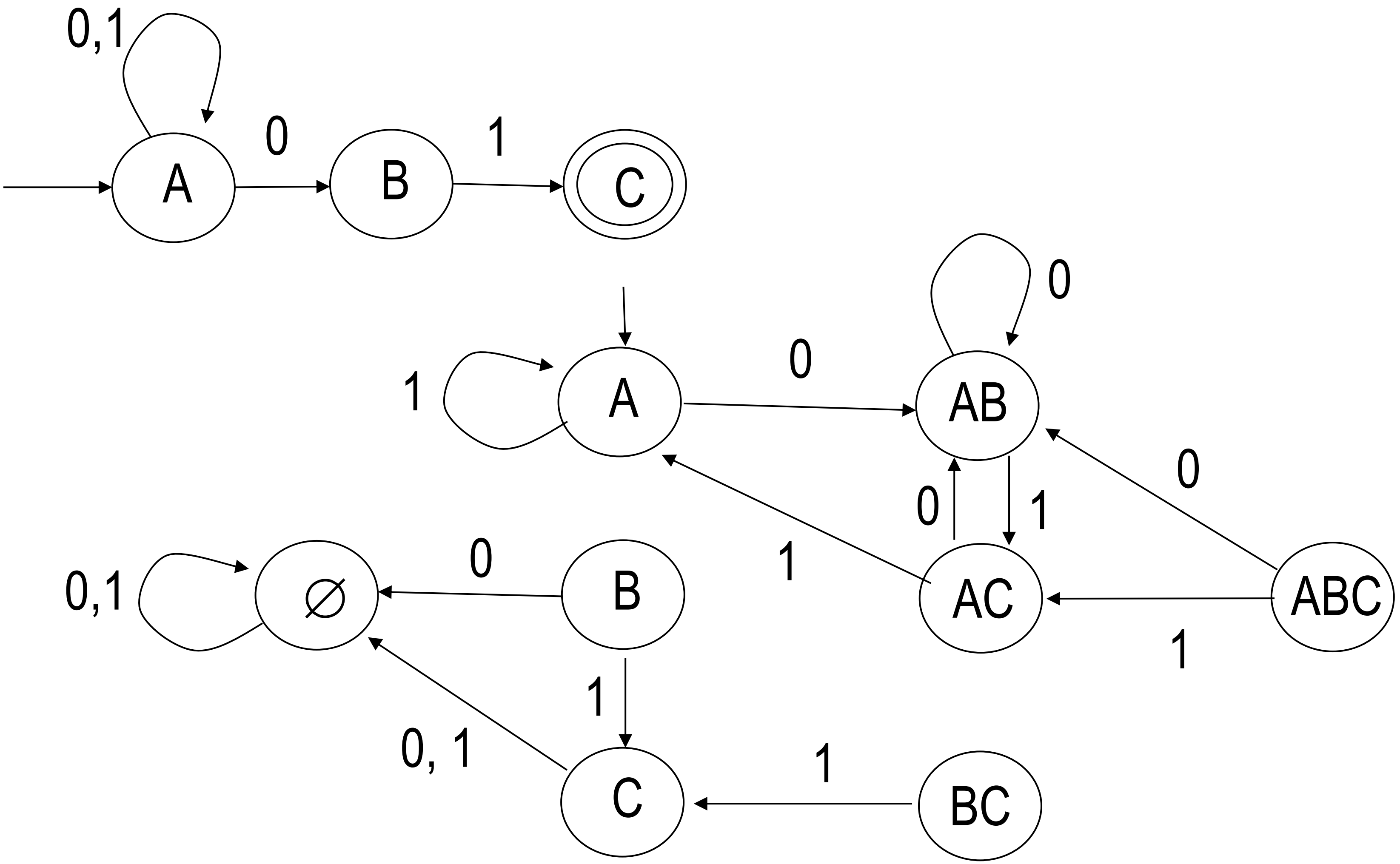




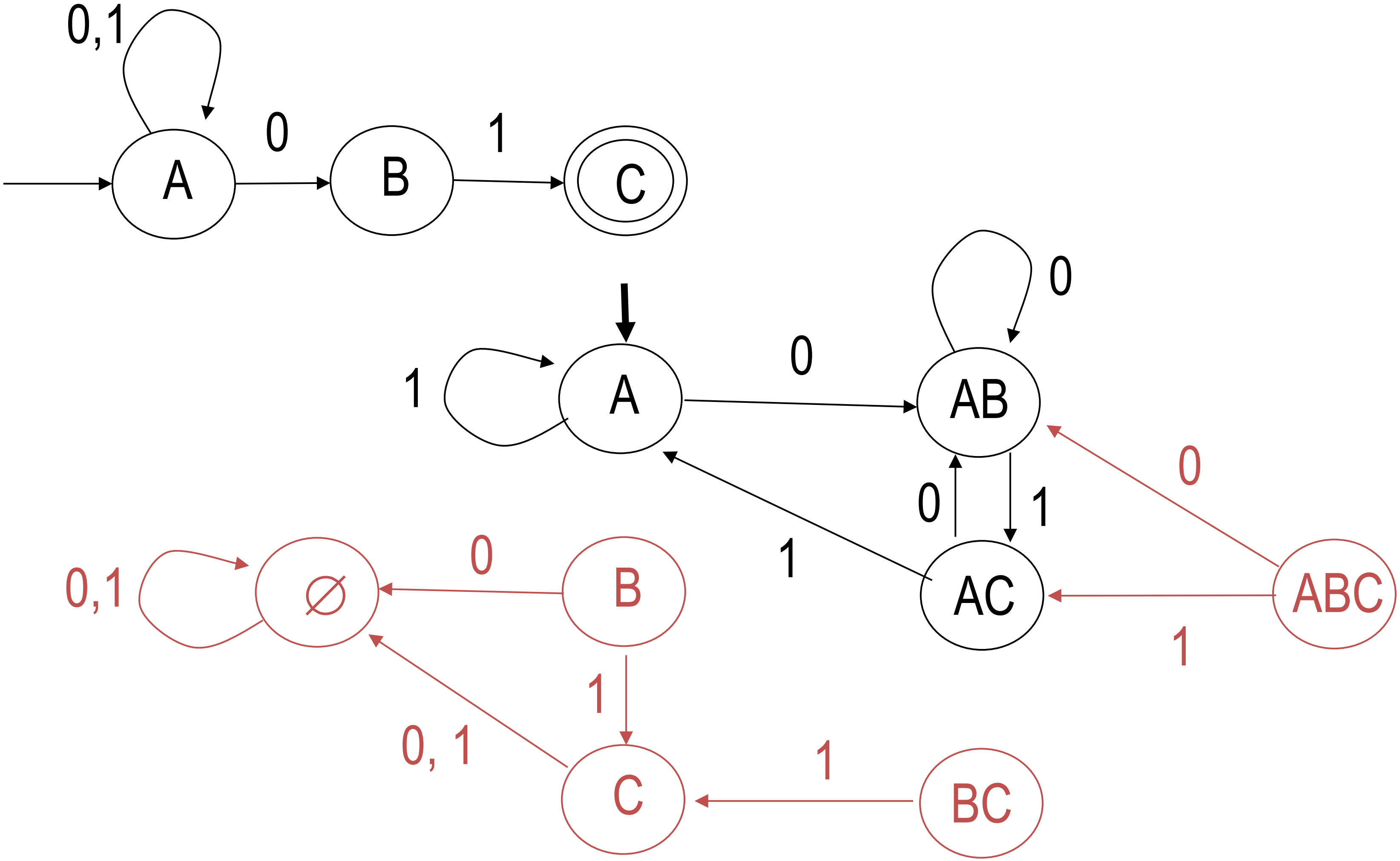
Practices: Conversion of NFA to DFA



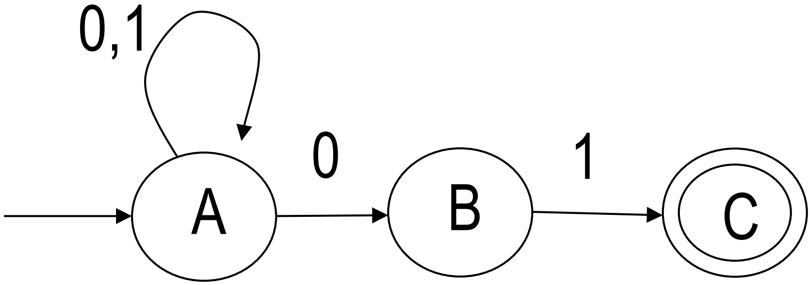
Practices: Conversion of NFA to DFA



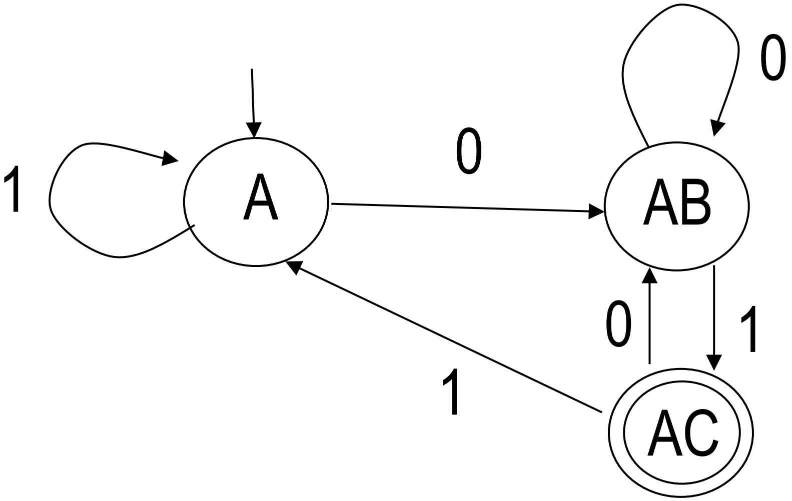
Practices: Conversion of NFA to DFA



Practices: Conversion of NFA to DFA



DFA



Equivalence of NFAs and DFAs

- Step 1: Draw all the states in DFAs
- Step 2: Define the transitions in DFAs based on the NFAs
- Step 3: Define the start state and accept state in DFAs
- Step 4: Remove all inaccessible states

Draw NFA online

- http://madebyevan.com/fsm/

  - Add a state: double-click on the canvas
  - Add an arrow: shift-drag on the canvas
  - Move something: drag it around
  - Delete something: click it and press the delete key (not the backspace key); On Laptop/Macbook, please press “Fn” + “Delete/backspace”.
  - Make accept state: double-click on an existing state
  - Add start state: shift-drag on the canvas to one state
  - Type numeric subscript: put an underscore before the number (like "S_0")
  - Type greek letter: put a backslash before it (like "\beta")