Lexical Analysis (Scanning)

Lecture 3

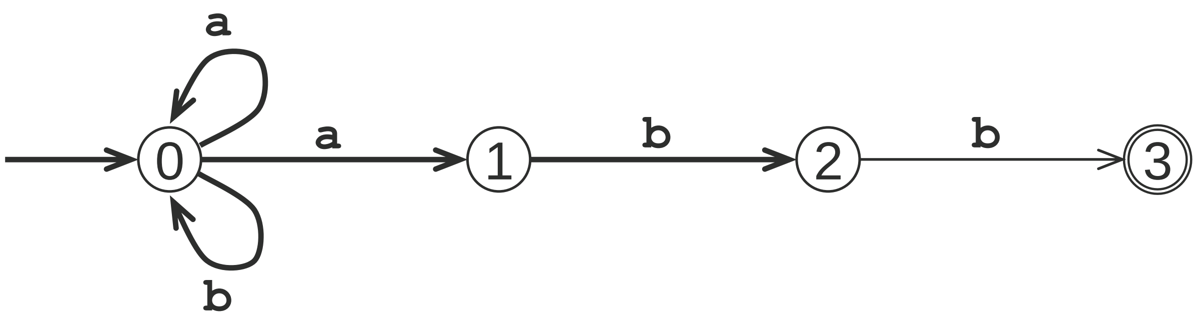
Finite Automata (FA)

DFA is a 5-tuple $(S, \Sigma, \delta, s_0, F)$ where:

- S is a finite set of *states.*
- Σ a finite set of symbols, the input *alphabet.*
- δ transition function is a *mapping* from $S \times \Sigma \rightarrow$ Set of states
- $s_0 \in S$ is the *start state*
- $F \subseteq S$ is the set of *accepting* (or *final*) *states*

Finite Automata

- States are nodes, transitions are directed labeled edges, some states are marked as *final*, one state is marked as *starting*.
- The mapping δ of an FA can be represented in a *transition table*



$S = \{0,1,2,3\}$

$\Sigma = \{a,b\}$

$s_0 = 0$

$F = \{3\}$

$\delta(0,\mathbf{a}) = \{0,1\}$

$\delta(0,\mathbf{b}) = \{0\}$

$\delta(1,\mathbf{b}) = \{2\}$

$\delta(2,\mathbf{b}) = \{3\}$

→

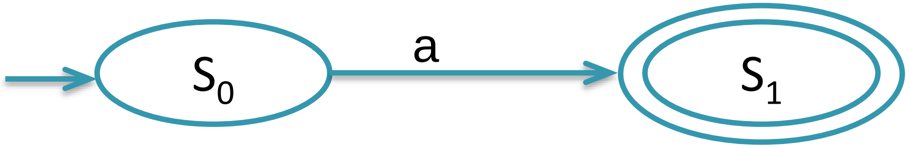
State	*Input* **a**	*Input* **b**
0	{0, 1}	{0}
1		{2}
2		{3}
*3		

NFA vs DFA

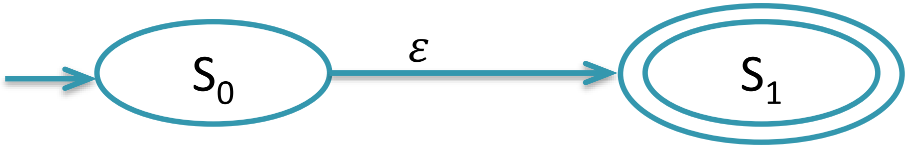
NFA has either:

- ➢Multiple transitions from one state on the same input
- ➢ε-move

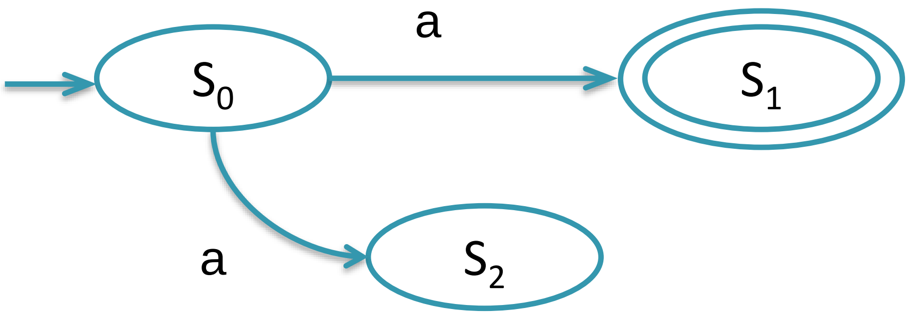
- A string is accepted by an NFA if there exists a sequence of transitions leads from the start state to some final state.



DFA



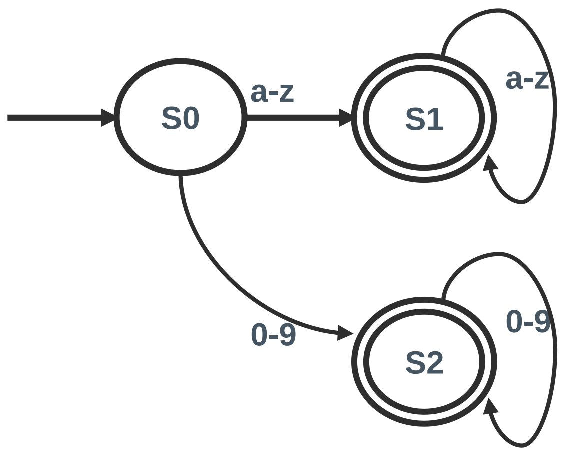
NFA



NFA

Finite Automata(Example)

1) Design FSM that accept RE=$[a\text{-}z]^+|[0\text{-}9]^+$
2) Match RE=$[a\text{-}z]^+|[0\text{-}9]^+$ with "Abcd 2004"
3) Which of the following Strings are accepted by the FSM?
   a. a
   b. Abcd 2004
   c. Abcd   2004
   d. 20000007
   e. abcd2004



2) {bcd,2004}
3) a,d

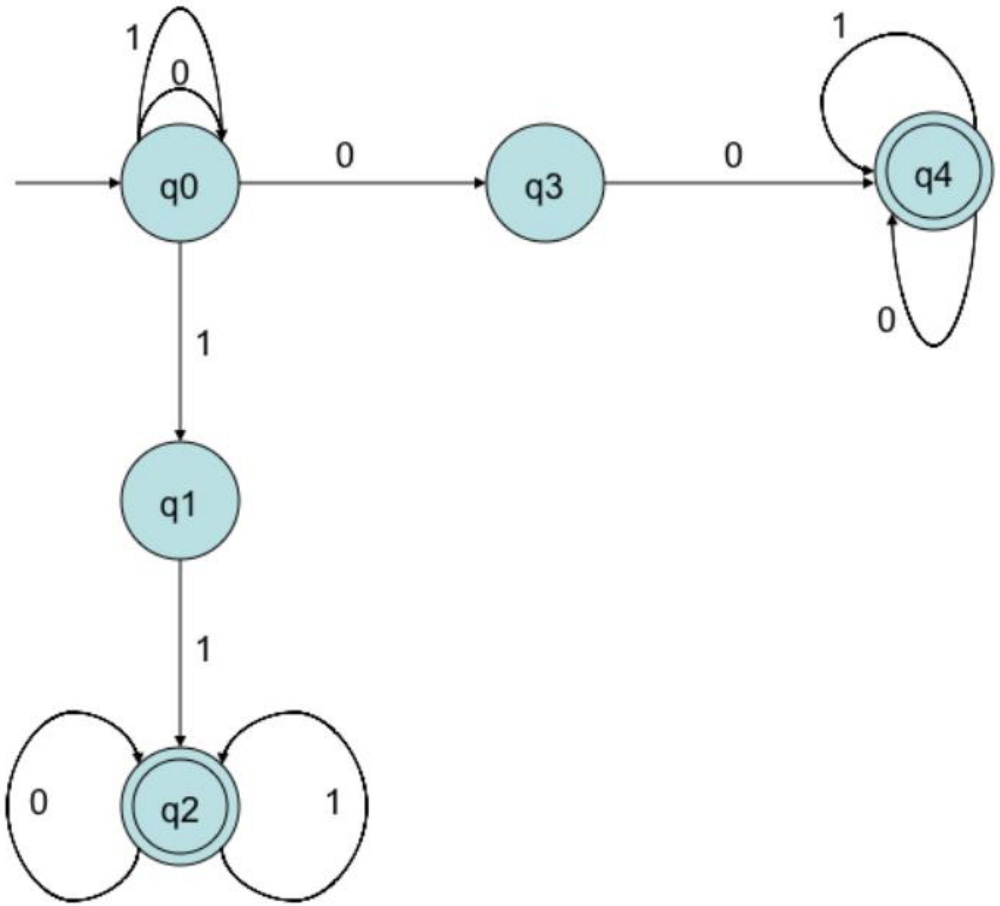
Finite Automata(Example)

1) Define the following FSM
2) Define language accepted by this machine
3) Is Machine DFA or NFA?

1) FA is a 5-tuple (S, Σ, δ, s_0, F) where:

1. S={q0,q1,q2,q3,q4)
2. Σ={0,1}
3. s0=q0
4. F={q2,q4}
5. δ=

state	symbol	
	0	1
q_0	$\{q_0, q_3\}$	$\{q_0, q_1\}$
q_1	ϕ	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_2\}$
q_3	$\{q_4\}$	ϕ
q_4	$\{q_4\}$	$\{q_4\}$

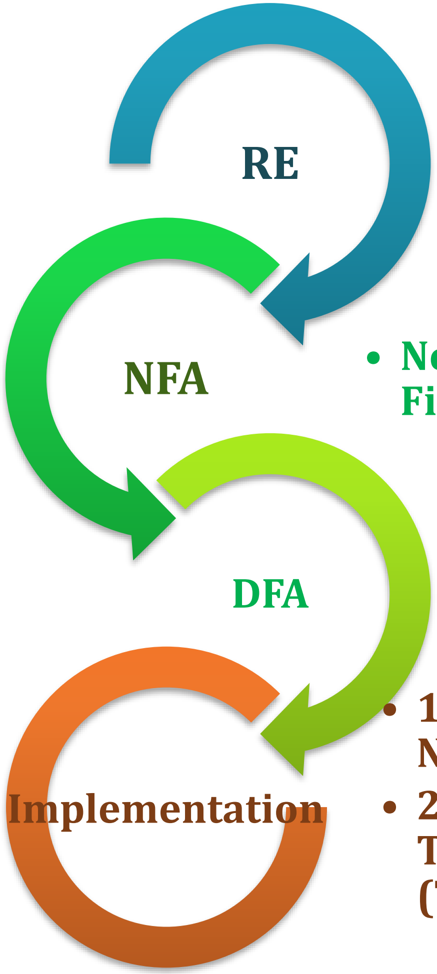


2) Language contains two consecutive 0's or two consecutive 1's

3) NFA because q0 has more than one transition on the same input.

Scanner Phases

- The scanning process is a pattern matching process and it can be divided in two main phases:
  - Pattern specification using regular expressions
  - Pattern recognition using finite automata for recognizing patterns



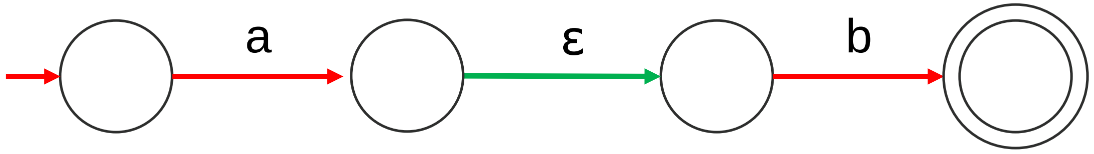
- Regular Expression
- Non Deterministic Finite Automata
- Deterministic Finite Automata
- 1- Using Doubly Nested Case Analysis
- 2- Using the Transition Table (Table Driven)

RE to NFA
(Thompson's construction)

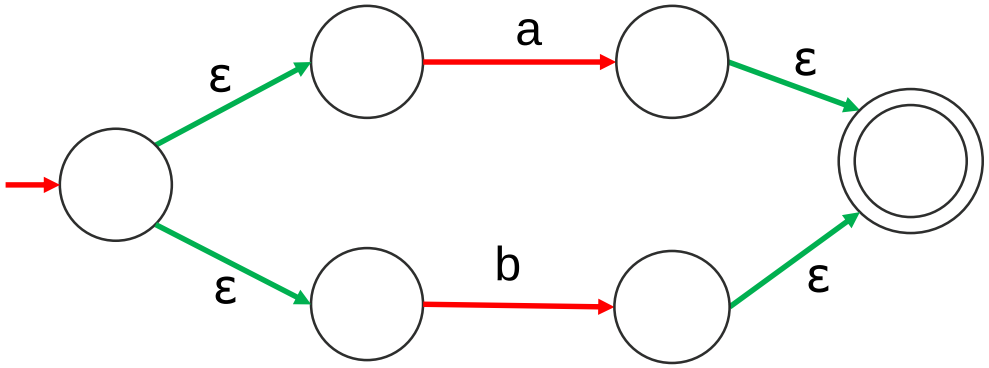
RE to NFA

- The construction of an NFA using a regular expression. The ε-transitions are used to “glue together” the machines of each piece of a regular expression to form a machine that corresponds to the whole expression

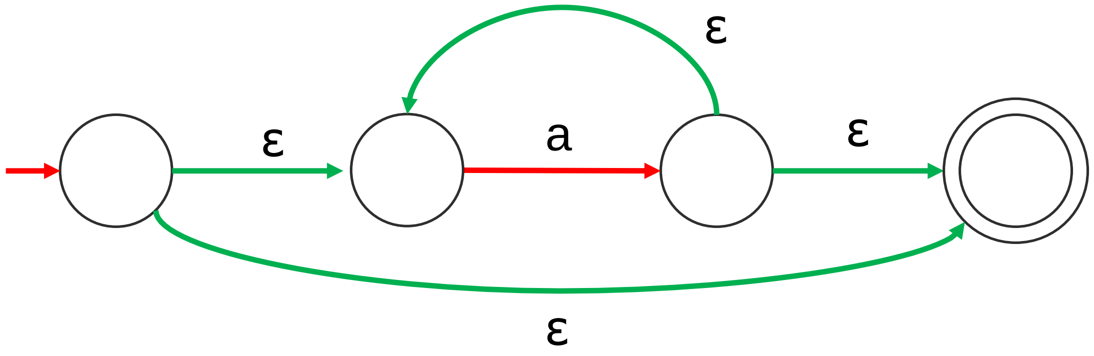
RE to NFA(Continued)



NFA for: ab



NFA for a|b

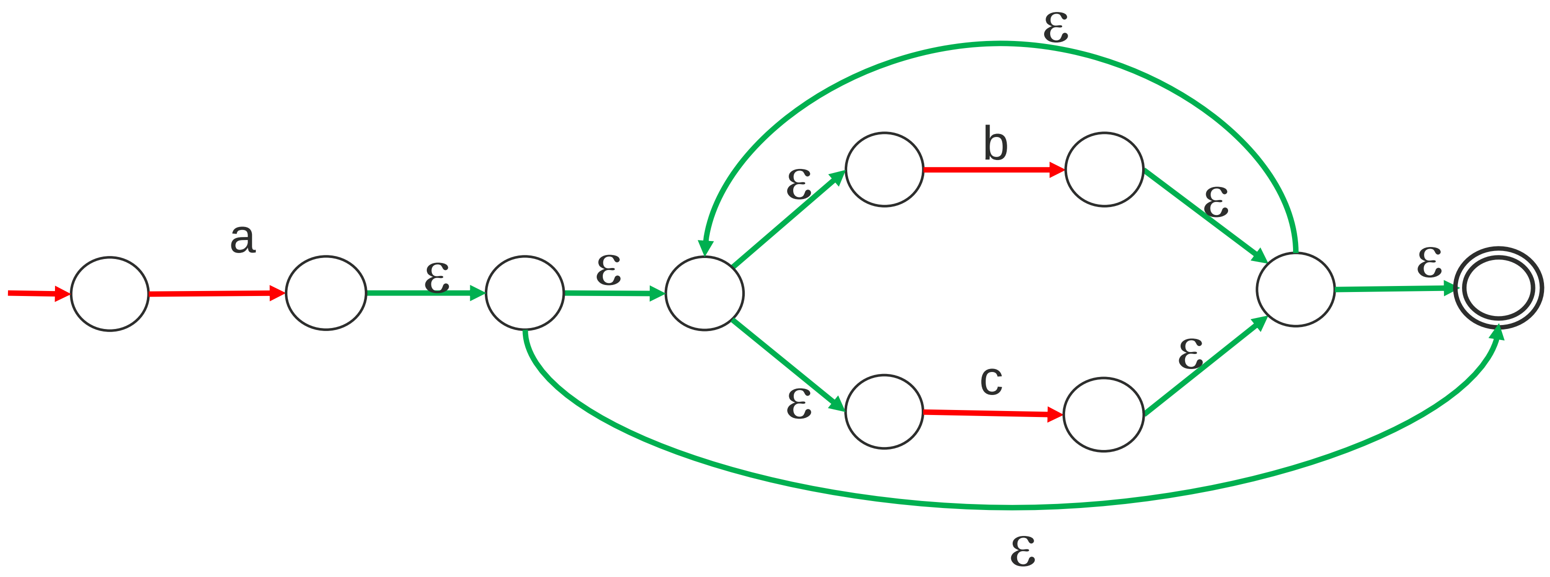


NFA for: a*

* Jim Hogg :2014 Washington university

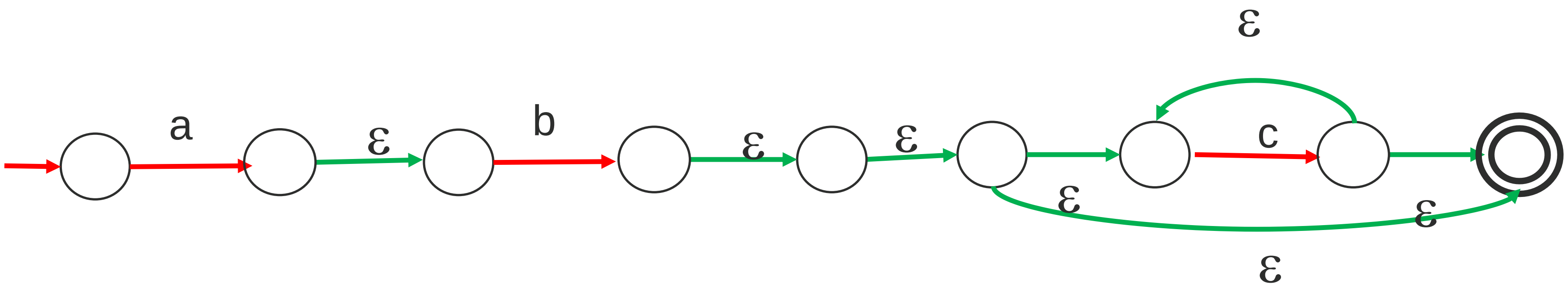
RE to NFA(Example-1)

- NFA for a(b|c)*



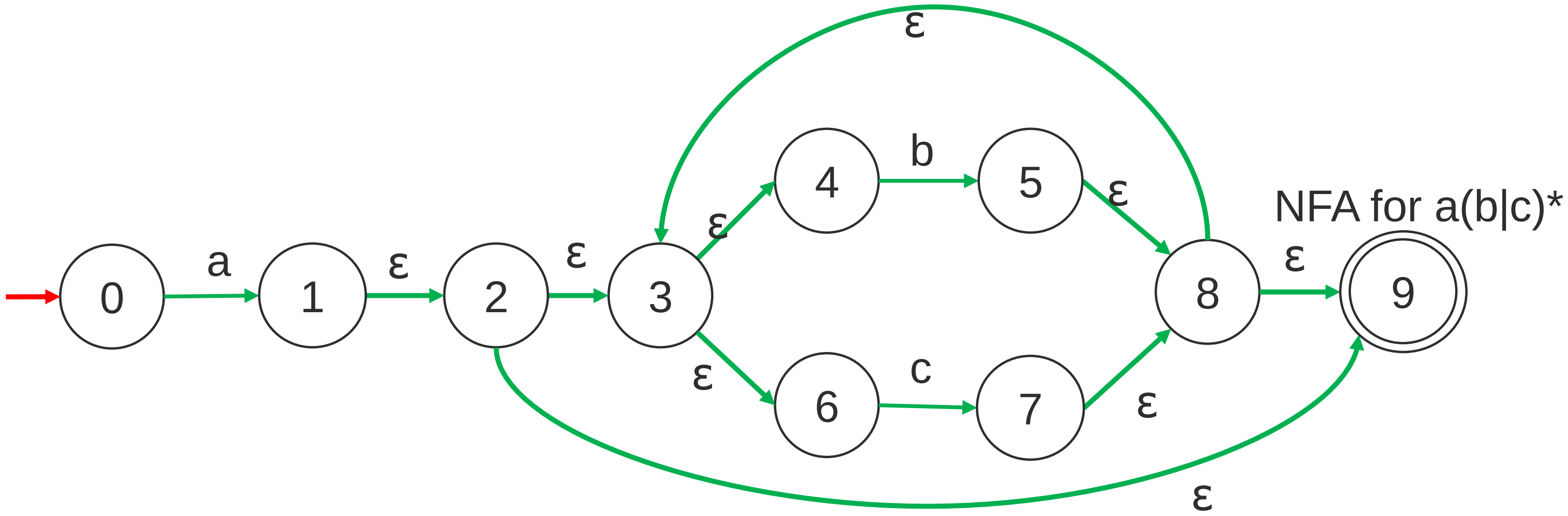
RE to NFA(Example-2)

- NFA for abc*



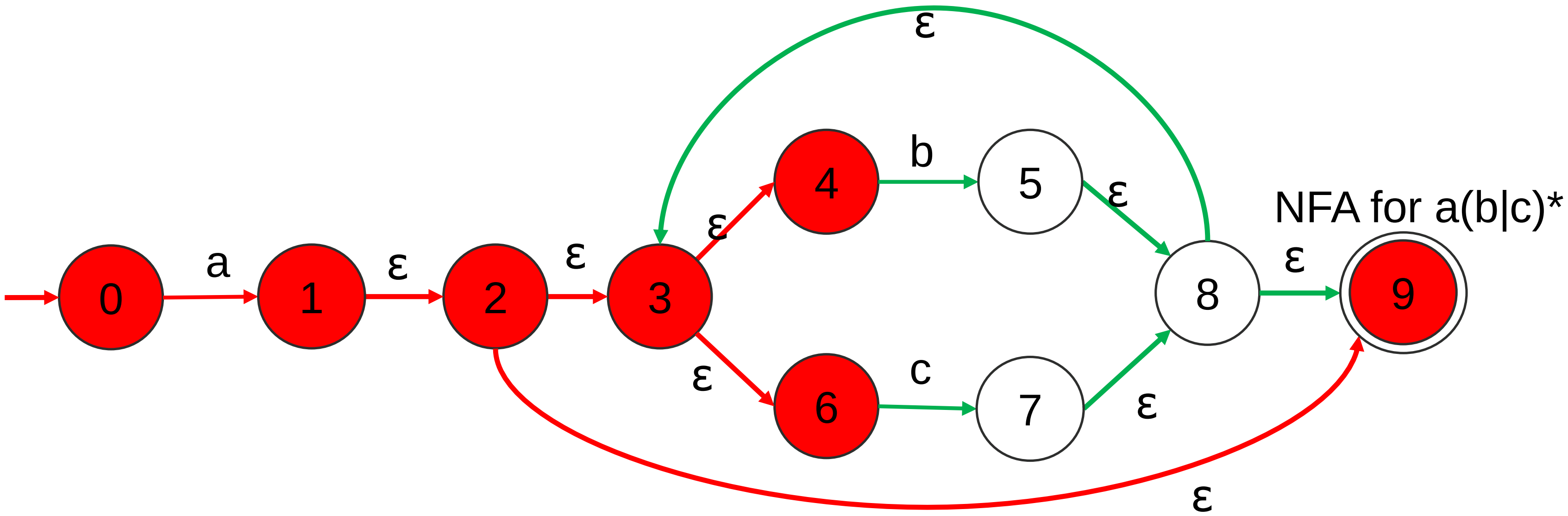
NFA to DFA
(Subset construction)

NFA to DFA(Continued)



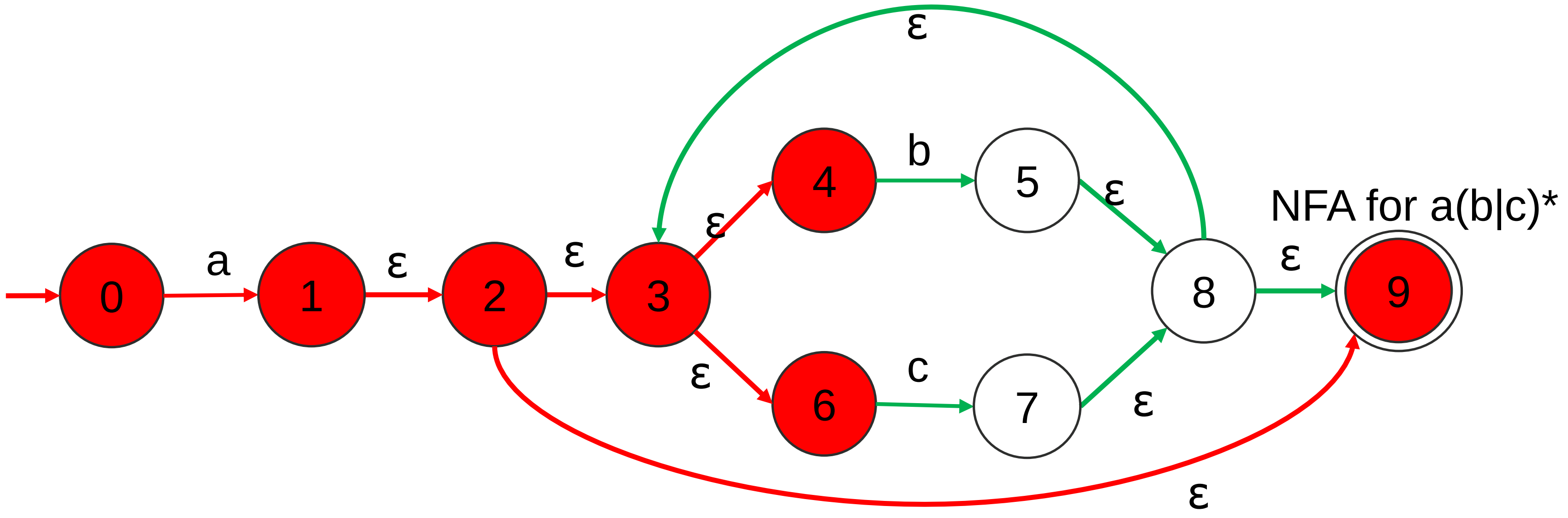
NFA State	a	b	c
d0 = n0	d1 = {1,2,3,4,6,9}	none	none

NFA to DFA(Continued)



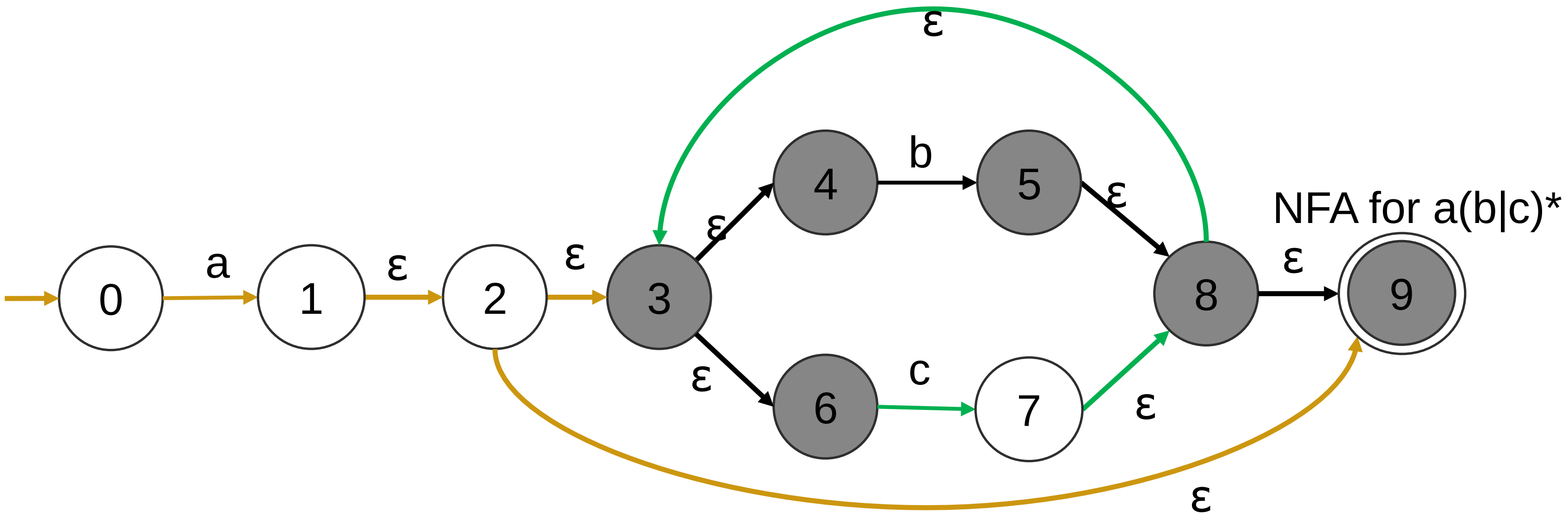
NFA State	a	b	c
d0 = n0	d1 = {1,2,3,4,6,9}	none	none
d1 = {1,2,3,4,6,9}	none	d2 = {3,4,5,6,8,9}	

NFA to DFA(Continued)



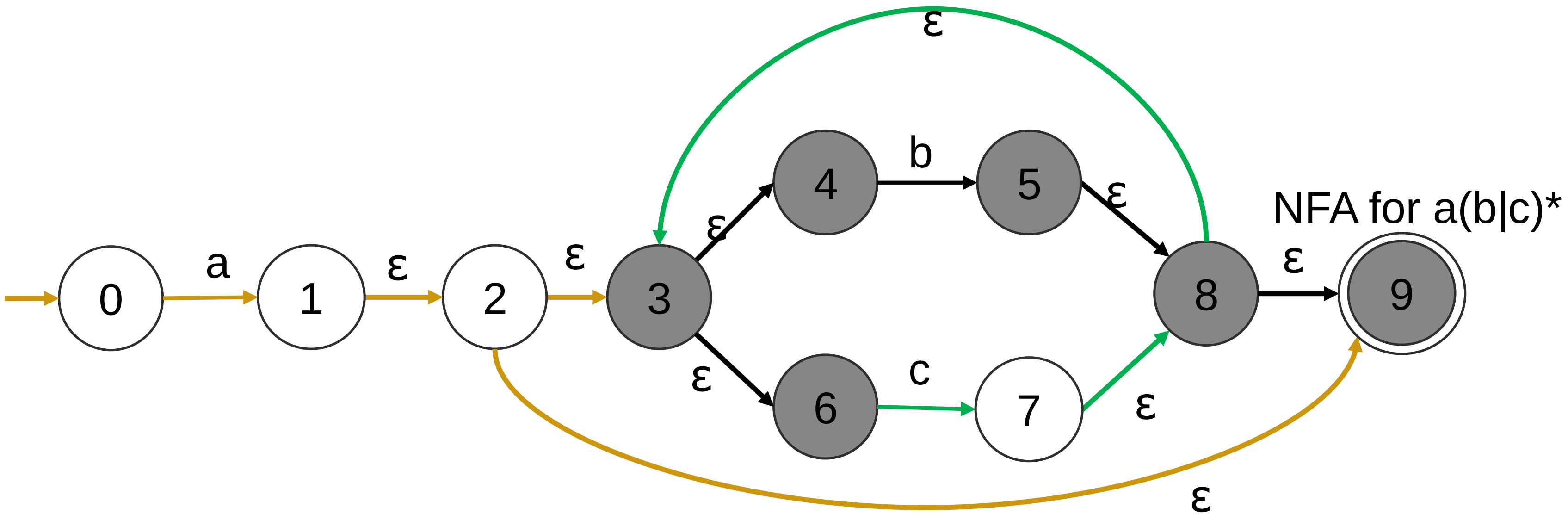
NFA State	a	b	c
d0 = n0	d1 = {1,2,3,4,6,9}	none	none
d1 = {1,2,3,4,6,9}	none	d2 = {3,4,5,6,8,9}	d3 = {3,4,6,7,8,9}

NFA to DFA(Continued)



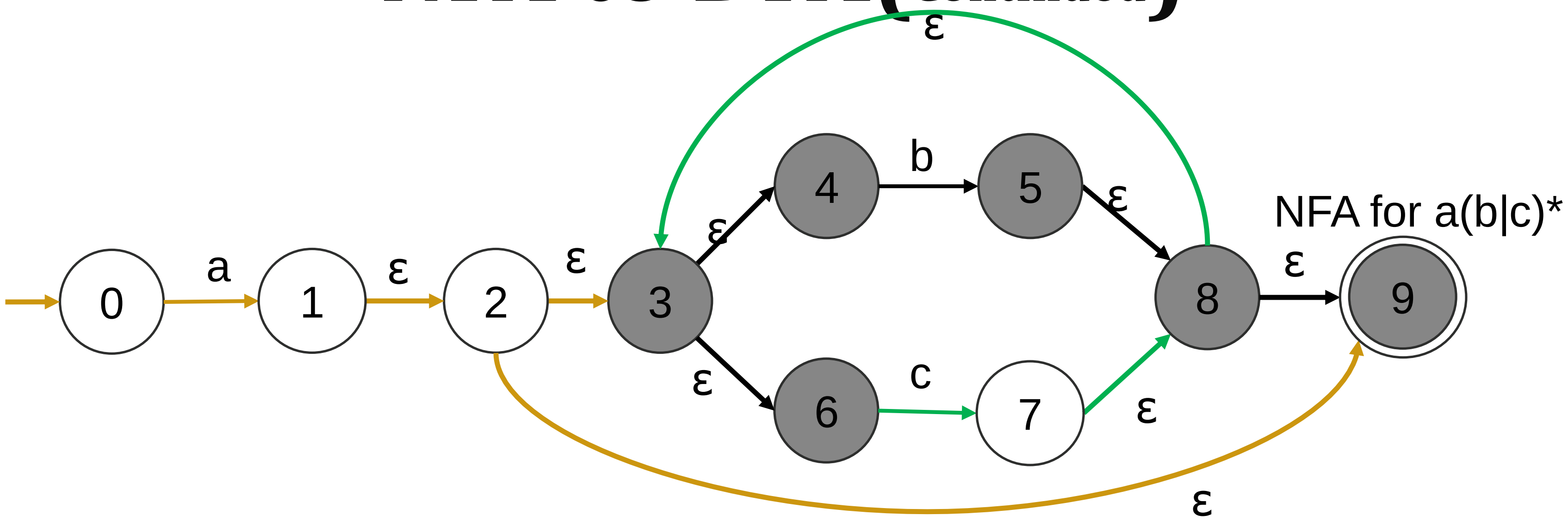
NFA State	a	b	c
d0 = n0	d1 = {1,2,3,4,6,9}	none	none
d1 = {1,2,3,4,6,9}	none	d2 = {3,4,5,6,8,9}	d3 = {3,4,6,7,8,9}
d2 = {3,4,5,6,8,9}	none	d2 = {3,4,5,6,8,9}	

NFA to DFA(Continued)



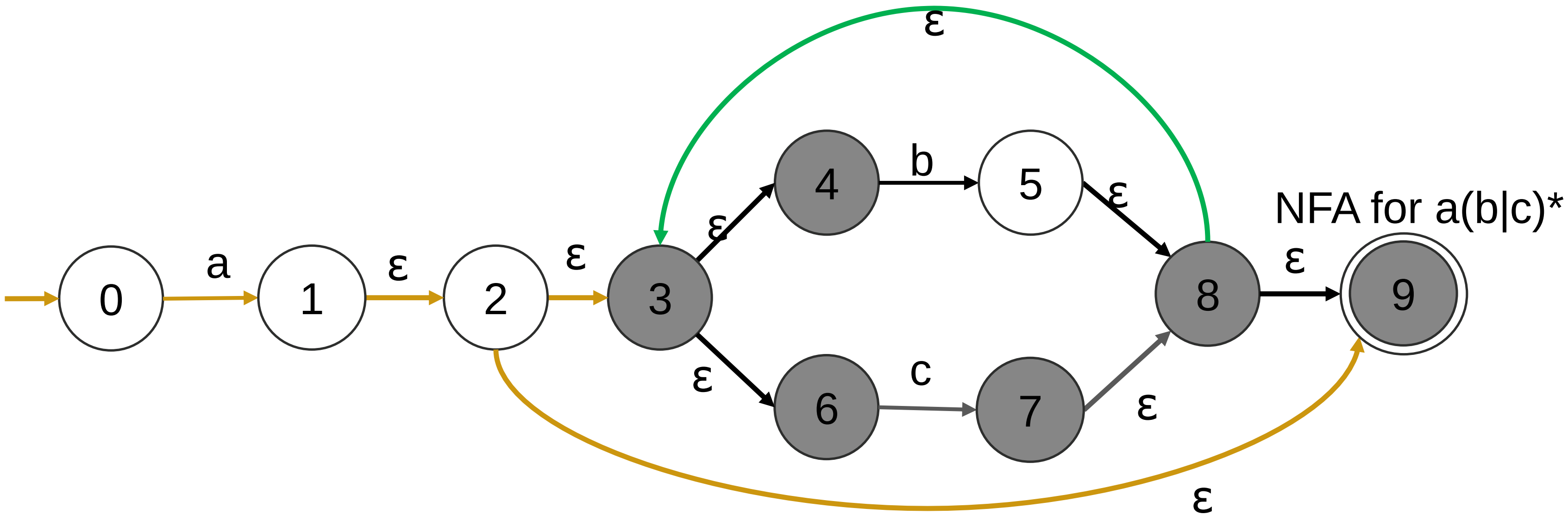
NFA State	a	b	c
d0 = n0	d1 = {1,2,3,4,6,9}	none	none
d1 = {1,2,3,4,6,9}	none	d2 = {3,4,5,6,8,9}	d3 = {3,4,6,7,8,9}
d2 = {3,4,5,6,8,9}	none	d2 = {3,4,5,6,8,9}	

NFA to DFA(Continued)



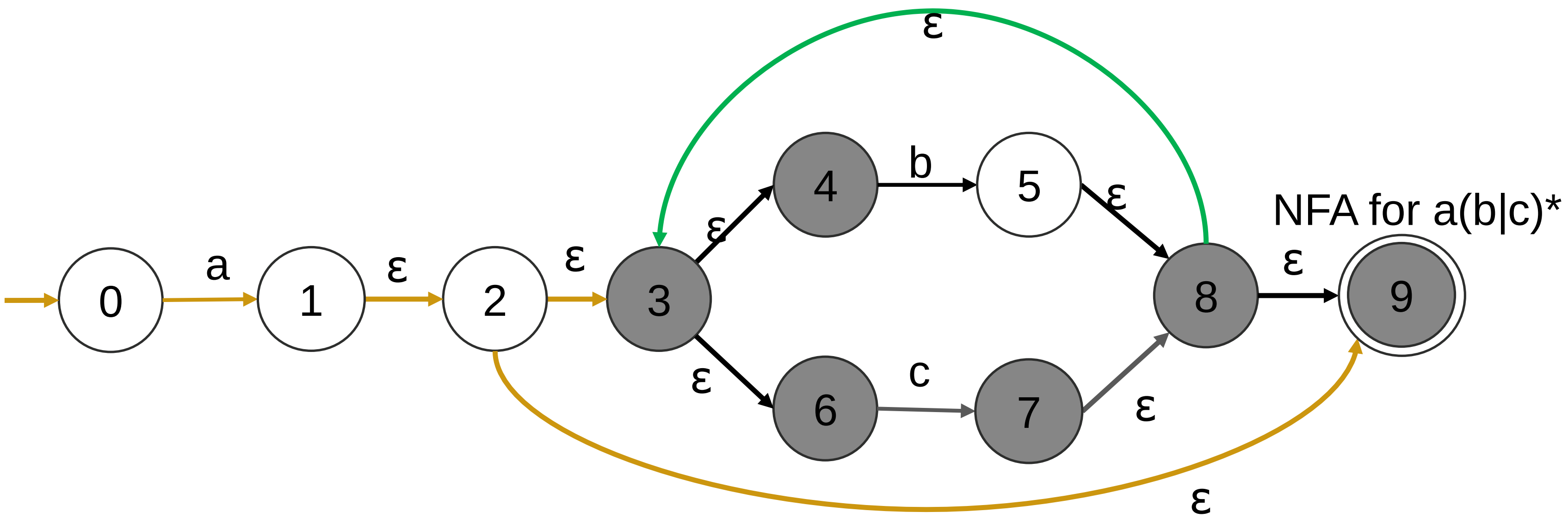
NFA State	a	b	c
d0 = n0	d1 = {1,2,3,4,6,9}	none	none
d1 = {1,2,3,4,6,9}	none	d2 = {3,4,5,6,8,9}	d3 = {3,4,6,7,8,9}
d2 = {3,4,5,6,8,9}	none	d2 = {3,4,5,6,8,9}	d3 = {3,4,6,7,8,9}

NFA to DFA(Continued)



NFA State	a	b	c
d0 = n0	d1 = {1,2,3,4,6,9}	none	none
d1 = {1,2,3,4,6,9}	none	d2 = {3,4,5,6,8,9}	d3 = {3,4,6,7,8,9}
d2 = {3,4,5,6,8,9}	none	d2 = {3,4,5,6,8,9}	d3 = {3,4,6,7,8,9}
d3 = {3,4,6,7,8,9}	none	d2 = {3,4,5,6,8,9}	

NFA to DFA(Continued)



NFA State	a	b	c
d0 = n0	d1 = {1,2,3,4,6,9}	none	none
d1 = {1,2,3,4,6,9}	none	d2 = {3,4,5,6,8,9}	d3 = {3,4,6,7,8,9}
d2 = {3,4,5,6,8,9}	none	d2 = {3,4,5,6,8,9}	d3 = {3,4,6,7,8,9}
d3 = {3,4,6,7,8,9}	none	d2 = {3,4,5,6,8,9}	d3 = {3,4,6,7,8,9}

NFA to DFA(Continued)

NFA State	a	b	c
d0	d1	-	-
*d1	-	d2	d3
*d2	-	d2	d3
*d3	-	d2	d3

DFA for a(b|c)*