Non-Deterministic Finite Automata

-

Hemant Kumar

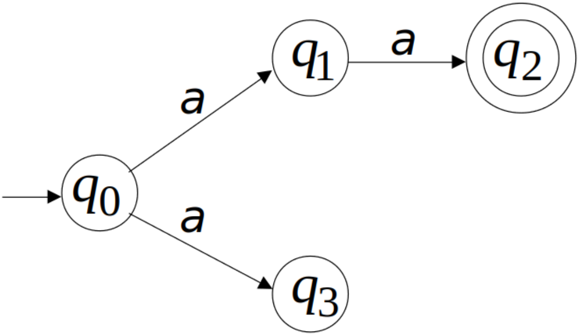
August 2, 2020

Outline

1. Non-Deterministic Finite Automata(NFA or NDFA)
2. Conversion NFA to DFA
3. ε-Non-Deterministic Finite Automata(ε-NFA)
    - Conversions ε-NFA to NFA

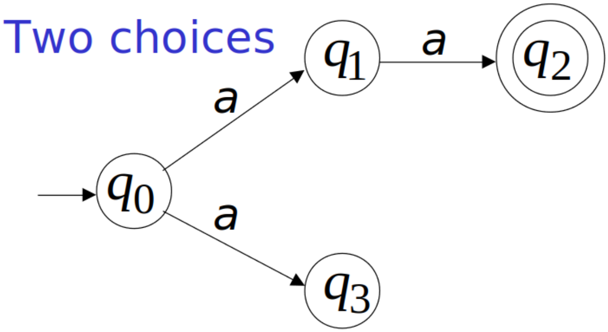
Non-Deterministic Finite Automata(NFA or NDFA)

Alphabet $=\{a\}$



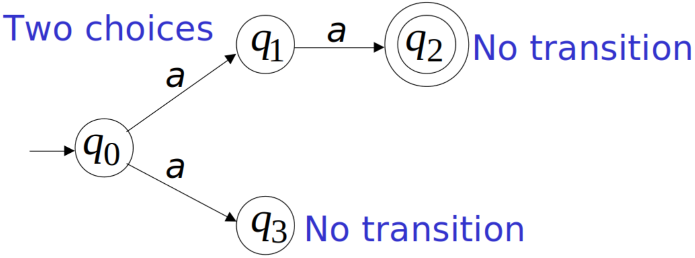
Non-Deterministic Finite Automata(NFA or NDFA)

Alphabet $=\{a\}$

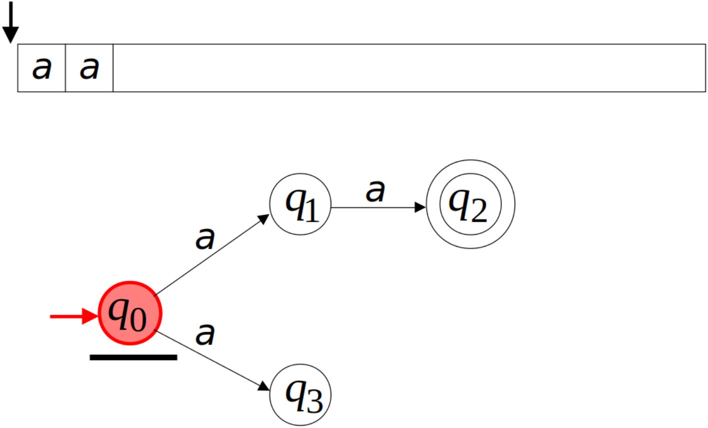


Non-Deterministic Finite Automata(NFA or NDFA)

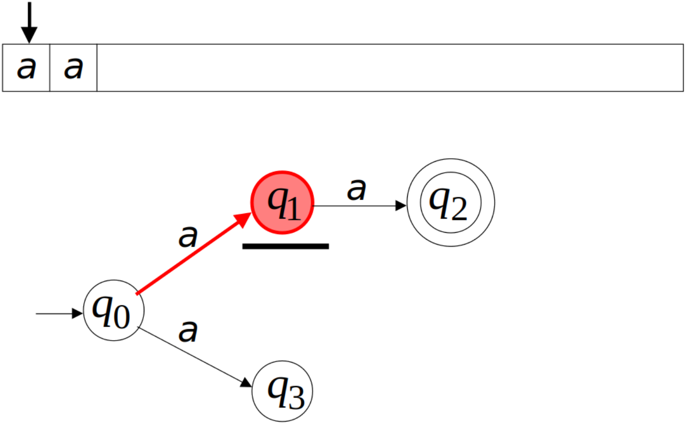
Alphabet $=\{a\}$



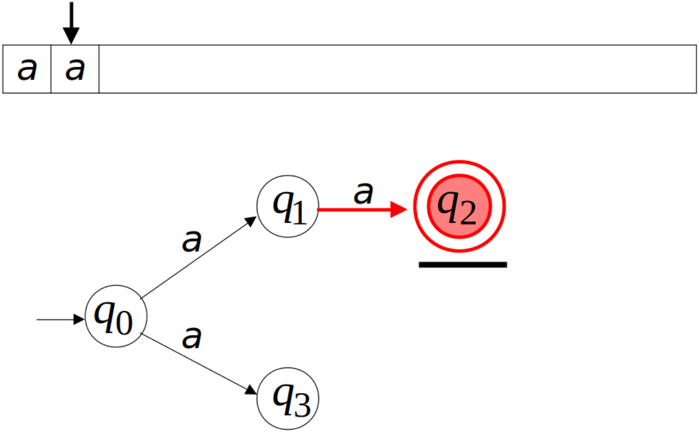
First Choice



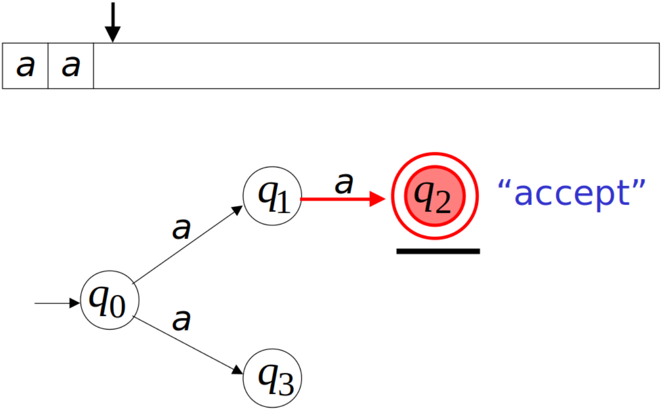
First Choice



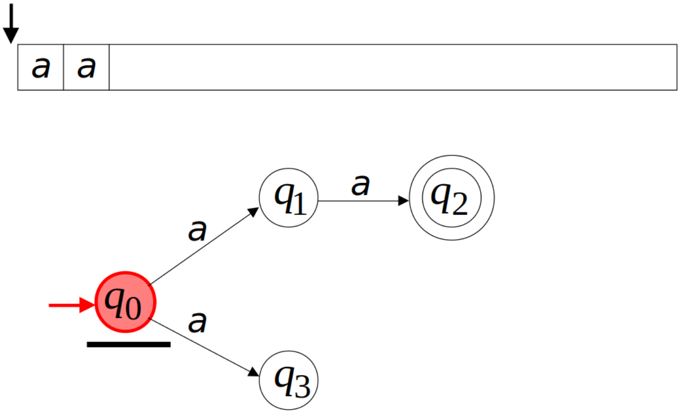
First Choice



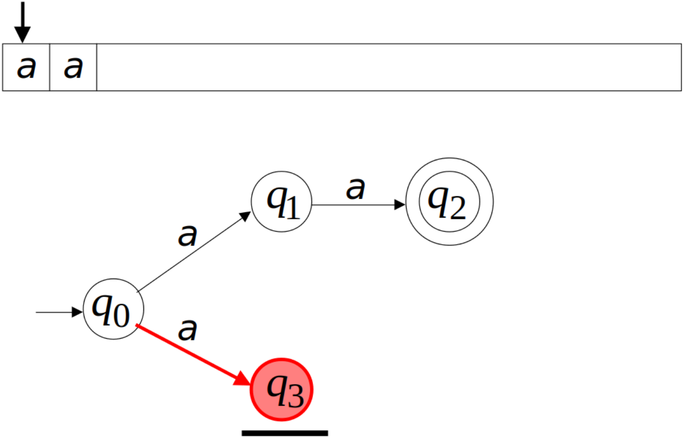
First Choice



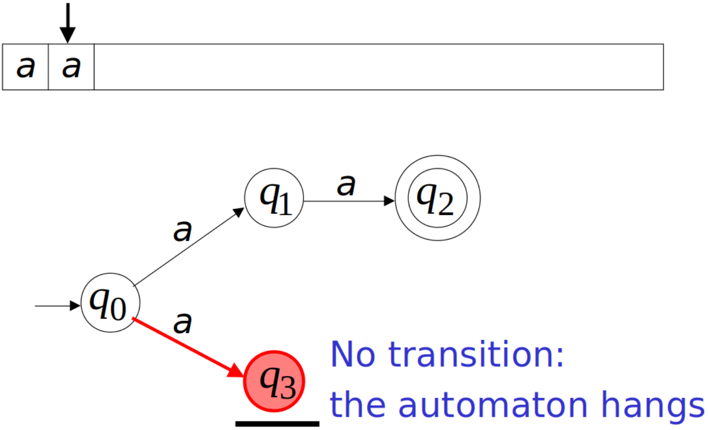
Second Choice



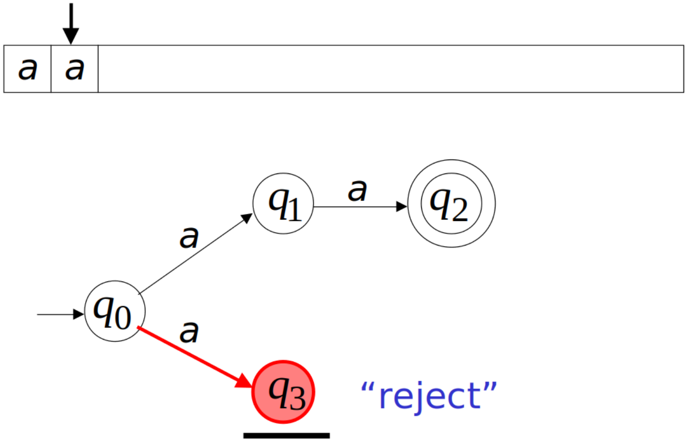
Second Choice



Second Choice



Second Choice



Non-Deterministic Finite Automata(NFA or NDFA)

Why do we need nfa's? NFA provides multiple options and are useful in solving problem easily.

Definition

The NFA contains five tuples in a

$$M = (Q, \Sigma, \delta, q_0, F)$$

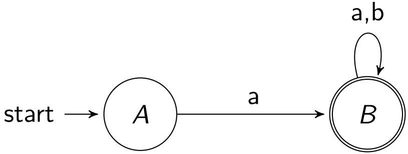
where,
Q is finite set of states
Σ is input alphabet
q_0 is start state $q_0 \in Q$
F is set of final states $Q \supseteq F$ (Q is superset of F)
δ is transition function $\delta : Q \times \Sigma \to 2^Q$

In NFA no need to draw dead/trap state.

Non-Deterministic Finite Automata(NFA or NDFA)

Draw NFA which accepts set of all strings start with 'a' over Σ.

Language $L = \{a, ab, aa, aaa, aba, \dots\}$



In NFA no need to draw dead/trap state.
In that example, $\delta(A, b)$ no such transition present here, it means NULL, then this situation is called **Dead Configuration**.

Conversion NFA to DFA

The steps to construct a DFA from a NFA are

1. choose initial state and apply transition function for input alphabet.
2. State obtaining from above step; those state is new state and apply transition function on new state and create new state.
3. Repeat steps.

Convert NFA to DFA

	input	
state	0	1
$\rightarrow p$	$\{p, q\}$	p
q	r	r
r	s	—
$*s$	s	s

Conversion NFA to DFA

Problem:

	input	
state	0	1
$\rightarrow p$	$\{p, q\}$	p
q	r	r
r	s	—
$*s$	s	s

Solution:

	input	
state	0	1
$\rightarrow [p]$	$[p, q]$	$[p]$

Conversion NFA to DFA

Problem:

state	input 0	input 1
$\rightarrow p$	$\{p, q\}$	p
q	r	r
r	s	—
$*s$	s	s

Solution:

state	input 0	input 1
$\rightarrow [p]$	$[p, q]$	$[p]$
$[p, q]$	$[p, q, r]$	$[p, r]$

Conversion NFA to DFA

Problem:

state	input 0	input 1
$\rightarrow p$	$\{p, q\}$	p
q	r	r
r	s	—
$*s$	s	s

Solution:

state	input 0	input 1
$\rightarrow [p]$	$[p, q]$	$[p]$
$[p, q]$	$[p, q, r]$	$[p, r]$
$[p, r]$	$[p, q, s]$	$[p]$

Conversion NFA to DFA

Problem:

state	input 0	input 1
$\rightarrow p$	$\{p, q\}$	p
q	r	r
r	s	—
$*s$	s	s

Solution:

state	input 0	input 1
$\rightarrow [p]$	$[p, q]$	$[p]$
$[p, q]$	$[p, q, r]$	$[p, r]$
$[p, r]$	$[p, q, s]$	$[p]$
$[p, q, r]$	$[p, q, r, s]$	$[p, r]$

Conversion NFA to DFA

Problem:

state	input 0	input 1
$\rightarrow p$	$\{p, q\}$	p
q	r	r
r	s	$-$
$*s$	s	s

Solution:

state	input 0	input 1
$\rightarrow [p]$	$[p, q]$	$[p]$
$[p, q]$	$[p, q, r]$	$[p, r]$
$[p, r]$	$[p, q, s]$	$[p]$
$[p, q, r]$	$[p, q, r, s]$	$[p, r]$
$*[p, q, s]$	$[p, q, r, s]$	$[p, r, s]$

Conversion NFA to DFA

Problem:

state	input 0	input 1
$\rightarrow p$	$\{p, q\}$	p
q	r	r
r	s	—
$*s$	s	s

Solution:

state	input 0	input 1
$\rightarrow [p]$	$[p, q]$	$[p]$
$[p, q]$	$[p, q, r]$	$[p, r]$
$[p, r]$	$[p, q, s]$	$[p]$
$[p, q, r]$	$[p, q, r, s]$	$[p, r]$
$*[p, q, s]$	$[p, q, r, s]$	$[p, r, s]$
$*[p, q, r, s]$	$[p, q, r, s]$	$[p, r, s]$

Conversion NFA to DFA

Problem:

state	input 0	input 1
$\rightarrow p$	$\{p, q\}$	p
q	r	r
r	s	—
$*s$	s	s

Solution:

state	input 0	input 1
$\rightarrow [p]$	$[p, q]$	$[p]$
$[p, q]$	$[p, q, r]$	$[p, r]$
$[p, r]$	$[p, q, s]$	$[p]$
$[p, q, r]$	$[p, q, r, s]$	$[p, r]$
$*[p, q, s]$	$[p, q, r, s]$	$[p, r, s]$
$*[p, q, r, s]$	$[p, q, r, s]$	$[p, r, s]$
$*[p, r, s]$	$[p, q, s]$	$[p, s]$

Conversion NFA to DFA

Problem:

state	input 0	input 1
$\rightarrow p$	$\{p, q\}$	p
q	r	r
r	s	—
$*s$	s	s

Solution:

state	input 0	input 1
$\rightarrow [p]$	$[p, q]$	$[p]$
$[p, q]$	$[p, q, r]$	$[p, r]$
$[p, r]$	$[p, q, s]$	$[p]$
$[p, q, r]$	$[p, q, r, s]$	$[p, r]$
$*[p, q, s]$	$[p, q, r, s]$	$[p, r, s]$
$*[p, q, r, s]$	$[p, q, r, s]$	$[p, r, s]$
$*[p, r, s]$	$[p, q, s]$	$[p, s]$
$*[p, s]$	$[p, q, s]$	$[p, s]$

Conversion NFA to DFA

Problem:

state	input 0	input 1
$\rightarrow p$	$\{p, q\}$	p
q	r	r
r	s	$-$
$*s$	s	s

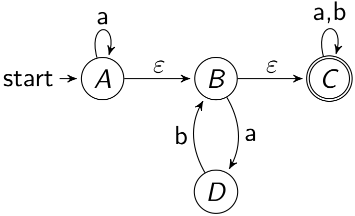
Solution:

state	input 0	input 1
$\rightarrow [p]$	$[p, q]$	$[p]$
$[p, q]$	$[p, q, r]$	$[p, r]$
$[p, r]$	$[p, q, s]$	$[p]$
$[p, q, r]$	$[p, q, r, s]$	$[p, r]$
$*[p, q, s]$	$[p, q, r, s]$	$[p, r, s]$
$*[p, q, r, s]$	$[p, q, r, s]$	$[p, r, s]$
$*[p, r, s]$	$[p, q, s]$	$[p, s]$
$*[p, s]$	$[p, q, s]$	$[p, s]$

ε-Non-Deterministic Finite Automata(ε-NFA)

Definition

The NFA contains five tuples in a
$M = (Q, \Sigma, \delta, q_0, F)$ where,
Q is finite set of states
Σ is input alphabet
q_0 is start state $q_0 \in Q$
F is set of final states $Q \supseteq F$ (Q is superset of F)
δ is transition function $\delta : Q \times \{\Sigma \cup \varepsilon\} \rightarrow 2^Q$



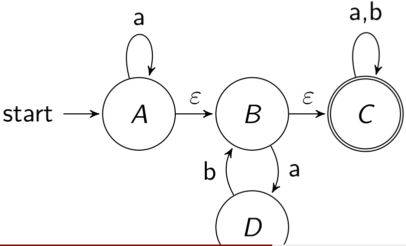
Conversions ε-NFA to NFA

Stpes for Conversions

1. Obtain $\varepsilon - closure$ of all the states.
2. Apply extended transition function for all input to all state
   $$\boxed{\delta'(q,\ input) = \varepsilon - closure(\delta(\varepsilon - closure(q), input))}$$
3. Repeat the step for all state and all input

1. $\varepsilon - closure$

The $\varepsilon - closure$ of state is a set of all states which are reachable from the given state using ε as input (and also include self state).



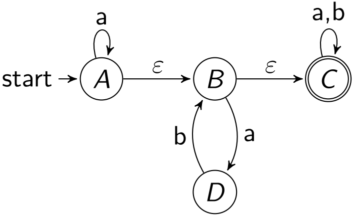
$$\varepsilon - closure(A) = \{A, B, C\}$$
$$\varepsilon - closure(B) = \{B, C\}$$
$$\varepsilon - closure(C) = \{C\}$$
$$\varepsilon - closure(D) = \{D\}$$

Conversions ε-NFA to NFA

2. Apply extended transition function(δ')

The $\varepsilon -$ *closure* of state is a set of all states which are reachable from the given state using ε as input (and also include self state).

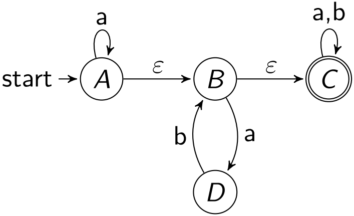
$$\begin{aligned}
\delta'(A,a) &= \varepsilon - closure\left(\delta\left(\varepsilon - closure\left(A\right), a\right)\right) \\
&= \varepsilon - closure(\delta(\{A, B, C\}, a)) \\
&= \varepsilon - closure(\delta(A,a) \ \cup \ \delta(B,a) \ \cup \ \delta(C,a)) \\
&= \varepsilon - closure(\{A, D, C\}) \\
&= \varepsilon - closure(A) \ \cup \ \varepsilon - closure(D) \ \cup \ \varepsilon - closure(C) \\
&= \{A, B, C\} \ \cup \ \{D\} \ \cup \ \{C\} \\
&= \{A, B, C, D\}
\end{aligned} \tag{1}$$



$$\begin{aligned}
\delta'(A,b) &= \varepsilon - closure\left(\delta\left(\varepsilon - closure\left(A\right), b\right)\right) \\
&= \varepsilon - closure(\delta(\{A, B, C\}, b)) \\
&= \varepsilon - closure(\delta(A,b) \ \cup \ \delta(B,b) \ \cup \ \delta(C,b)) \\
&= \varepsilon - closure(\{\phi \cup \phi \cup C\}) \\
&= \varepsilon - closure(C) \\
&= \{C\}
\end{aligned} \tag{2}$$

2. Apply extended transition function(δ')

$$\begin{aligned}\delta'(B,a) &= \varepsilon - closure\left(\delta\left(\varepsilon - closure\left(B\right), a\right)\right) \\ &= \{C, D\}\end{aligned} \tag{3}$$



$$\begin{aligned}\delta'(B,b) &= \varepsilon - closure\left(\delta\left(\varepsilon - closure\left(B\right), b\right)\right) \\ &= \{D\}\end{aligned} \tag{4}$$

$$\begin{aligned}\delta'(C,a) &= \varepsilon - closure\left(\delta\left(\varepsilon - closure\left(C\right), a\right)\right) \\ &= \{\phi\}\end{aligned} \tag{5}$$

$$\begin{aligned}\delta'(C,b) &= \varepsilon - closure\left(\delta\left(\varepsilon - closure\left(C\right), b\right)\right) \\ &= \{B, D\}\end{aligned} \tag{6}$$

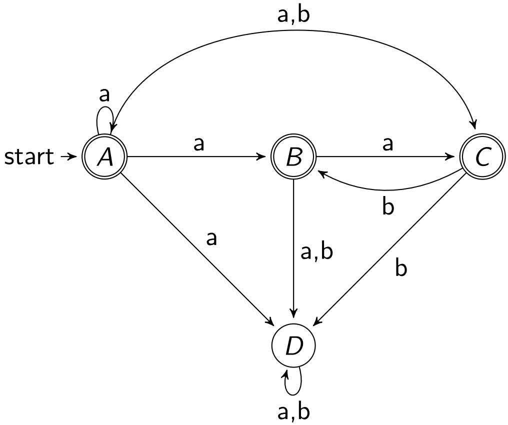
$$\begin{aligned}\delta'(D,a) &= \varepsilon - closure\left(\delta\left(\varepsilon - closure\left(D\right), a\right)\right) \\ &= \{D\}\end{aligned} \tag{7}$$

$$\begin{aligned}\delta'(D,b) &= \varepsilon - closure\left(\delta\left(\varepsilon - closure\left(D\right), b\right)\right) \\ &= \{D\}\end{aligned} \tag{8}$$

Now

Now, summarize all the δ' computed,

$\delta'(A,a) = \{A,B,C,D\}$, $\delta'(A,b) = \{C\}$, $\delta'(B,a) = \{C,D\}$,
$\delta'(B,b) = \{D\}$, $\delta'(C,a) = \{\phi\}$, $\delta'(C,b) = \{B,D\}$,
$\delta'(D,a) = \{D\}$, $\delta'(D,b) = \{D\}$



Here A, B, and C is a final state because $\varepsilon - closure(A)$, $\varepsilon - closure(B)$ and $\varepsilon - closure(C)$ contains final state C.

Conversion $\varepsilon-$NFA to DFA

Steps for conversions

1. Obtains $\varepsilon - closure$ of all states.
   Let $\varepsilon - closure(q) = \{p_1, p_2, p_3, \ldots, p_n\}$ then $[p_1, p_2, p_3, \ldots, p_n]$ becomes new states of DFA.
2. Apply given extended transition function on new state which is generated by step 1.
   We apply transition function to new states $[p_1, p_2, p_3, \ldots, p_n]$ for each input.

$$\delta'([p_1, p_2, \ldots, p_n],\ a) = \varepsilon - closure(\delta(p_1,\ a) \cup \delta(p_2,\ a) \cup \ldots \delta(p_n,\ a))$$

$$= \bigcup_{i=1}^{n} \varepsilon - closure(\delta(p_i,\ input))$$