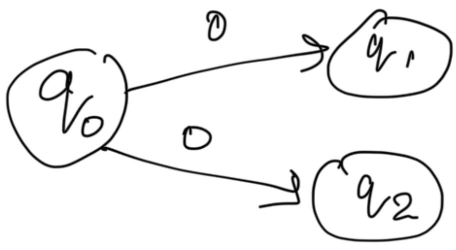
Non - deterministic Finite Automata

* For any particular input symbol, the machine can move to multiple state.

Previously, 

This is not the case fore NFA. For getting 0, there might be multiple transitions.

* At the same time, we don't have to show transition fore all input symbols.

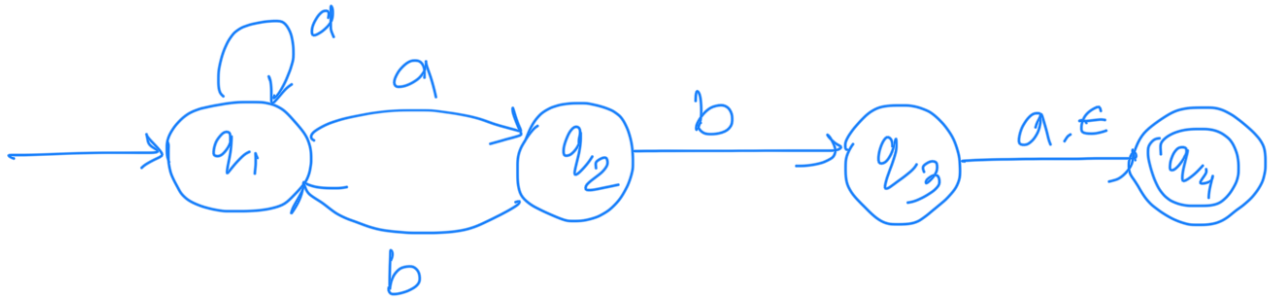


* We can go to other state without giving any input.

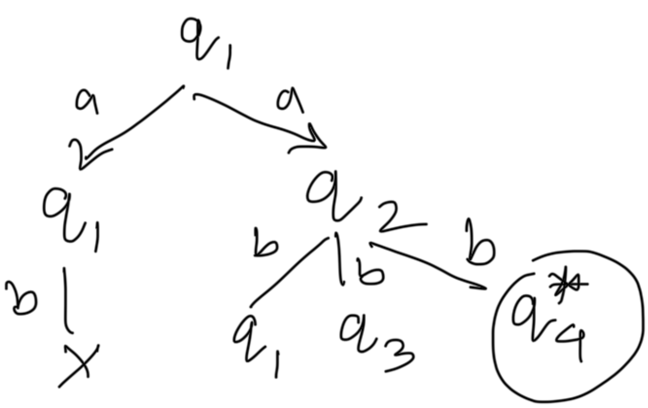
$$q_0 \longrightarrow q_1 , q_2$$

* However, NFA can't be used for machine.
* We use NFA for $\boxed{\text{design purpose}}$.
* To use this for machine we need to convert the NFA to DFA.

$\boxed{\text{Example}}$

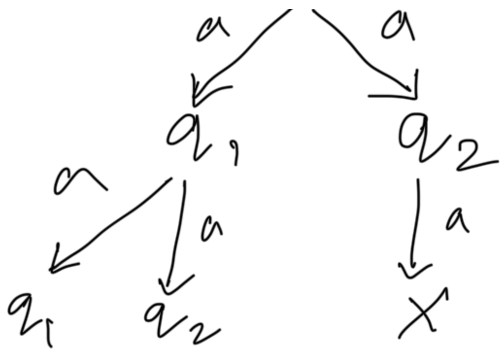


input: ab

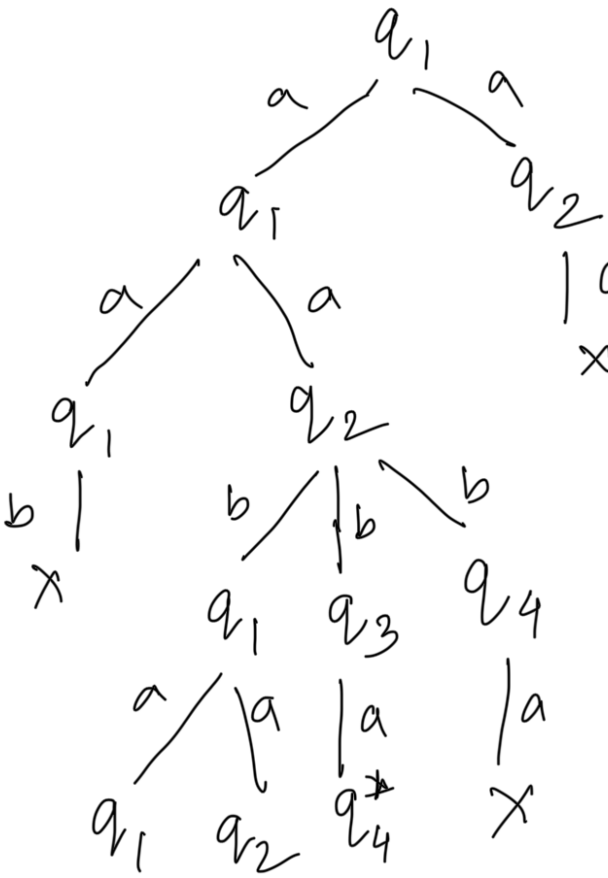


input: aa

q_1



input : aaba

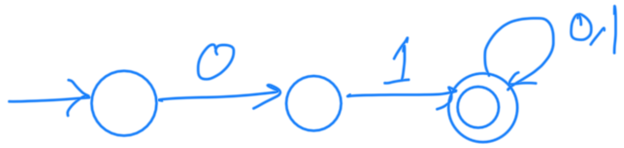


* NFA has some choices. From those choices it always selects or makes the right choice.

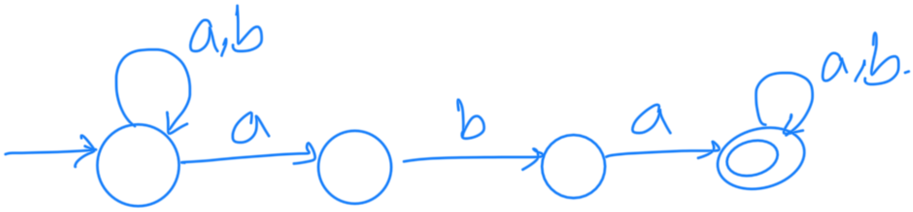
↳ You may ask, if there exists an accepting choice, how does the NFA makes the right choice? $\Longrightarrow$ It simples does that.

Some Examples

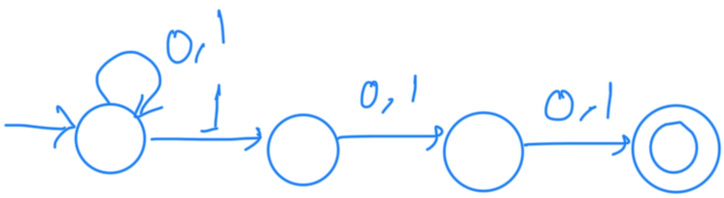
1. $L = \{ w \in \{0,1\}^* : w \text{ starts with } 01 \}$



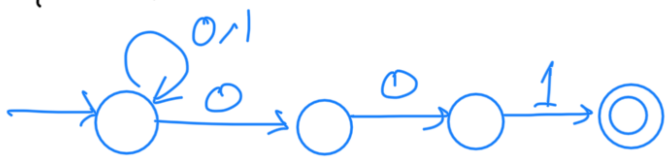
2. $L = \{ w \in \{a,b\}^* : w \text{ contains 'aba' as a substring} \}$



3. $L = \{ w \in \{0,1\}^* : \text{The 3red last symbol in } w \text{ is } 1 \}$

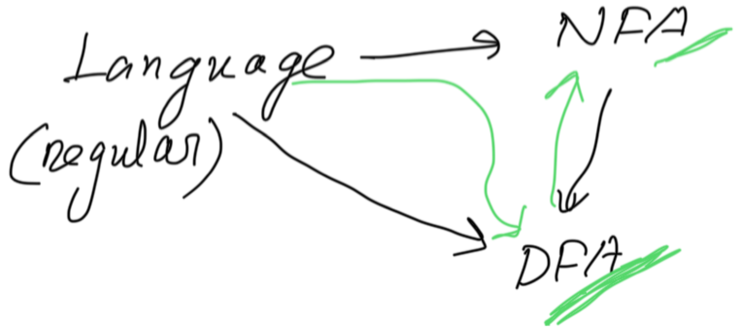


4. $L = \{ w \in \{0,1\}^* : w \text{ ends with } 001 \}$



Every NFA has an equivalent DFA.

NFA to DFA



power set

set $\rightarrow$ n elements

powerset $\rightarrow$ 2^n "

$\{\_\ \_\}$

$\{a, b\} \rightarrow 2^2 = 4$

$a < \begin{matrix} T \\ F \end{matrix}$

$b < \begin{matrix} T \\ F \end{matrix}$

FF	$\{F, F\}$	$\leftarrow \{\ \}$	$\rightarrow$	$\{\_, \_\}$
TF	$\{T, F\}$	$\leftarrow \{a\}$	$\rightarrow$	$\{a, \_\}$
FT	$\{F, T\}$	$\leftarrow \boxed{\{b\}}$	$\rightarrow$	$\{\_, b\}$
TT	$\{T, T\}$	$\leftarrow \{a, b\}$	$\rightarrow$	$\{a, b\}$

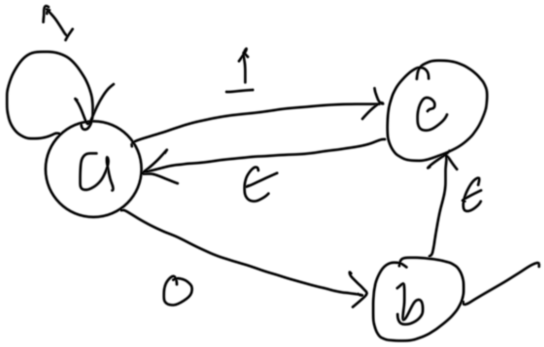
{T/F T/F

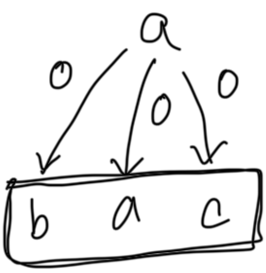
$2 \times 2 = 2^2$

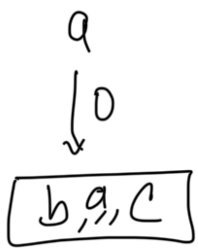
$\{a, b, c\} \longrightarrow \{a, c\} \rightarrow$ TFT

$2 \times 2 \times 2 = 2^3$

0/1 0/1 0/1





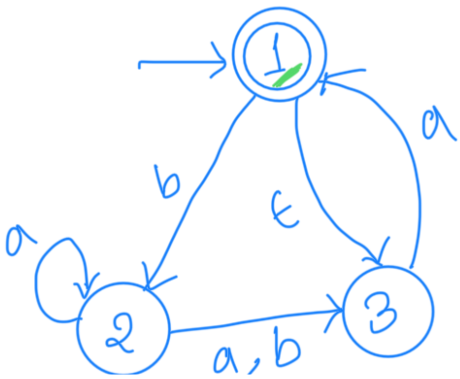


10

$a \xrightarrow{0} \boxed{b,c,a}$

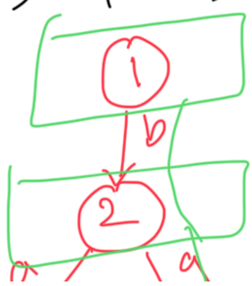
$a \xrightarrow{1} a,c$

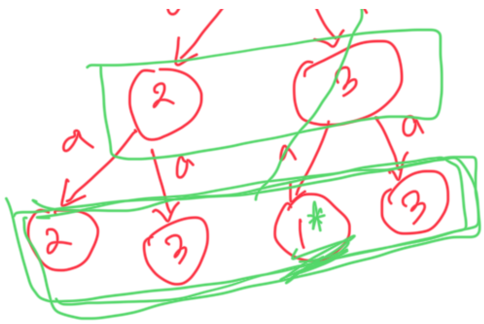
NFA



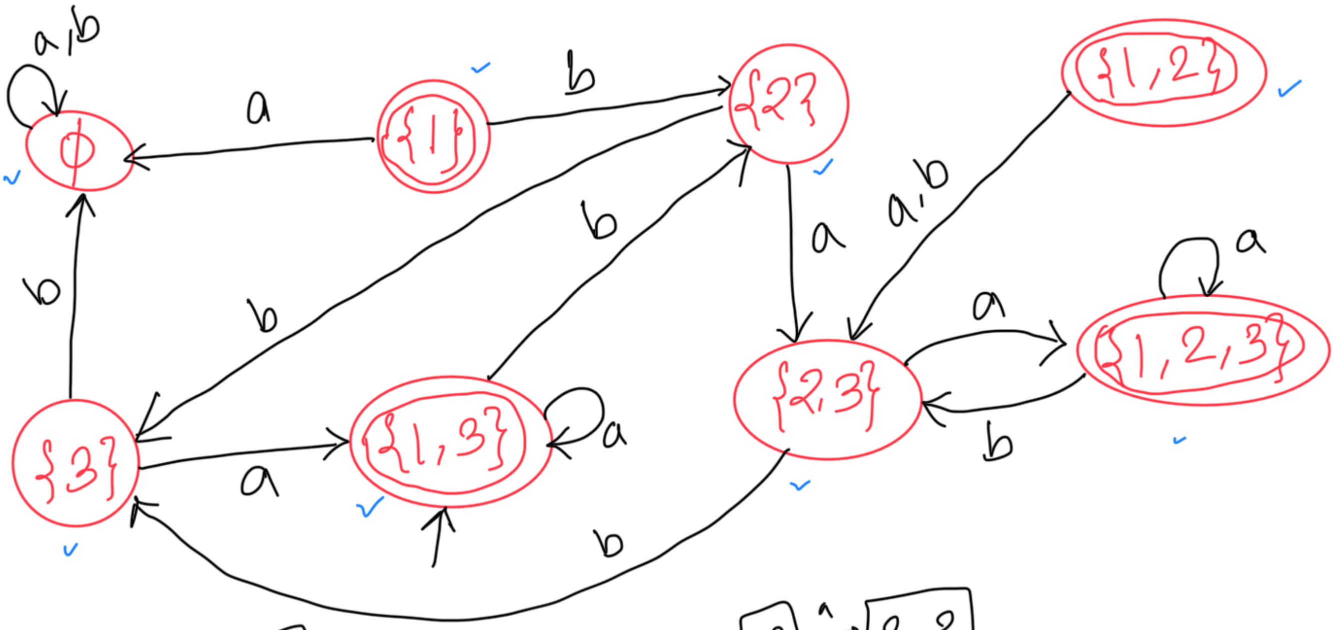
$E(1)=\{1,3\}$

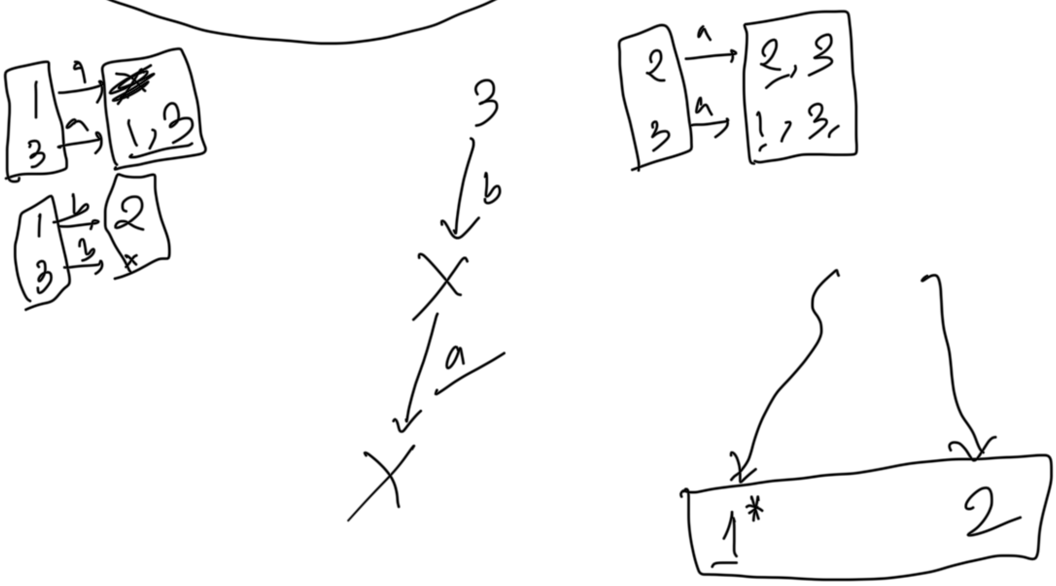
baa





Equivalent DFA



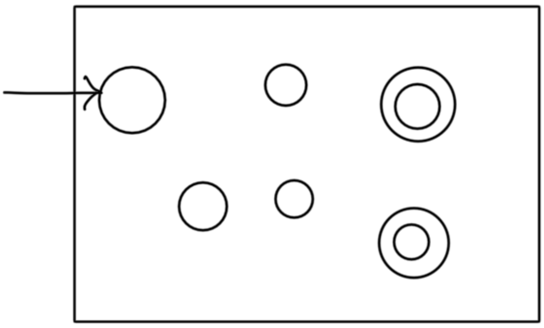


Regular Operations

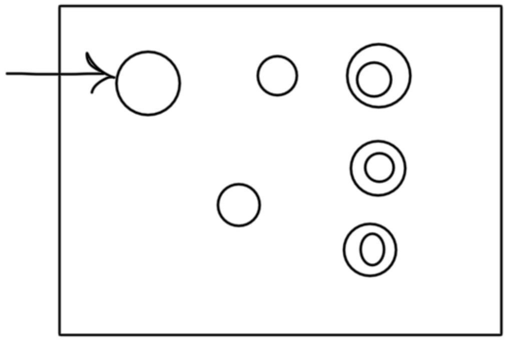
Union

$$L = L_1 \cup L_2$$

$$\begin{array}{ccc} \downarrow & \downarrow & \downarrow \\ N & N_1 & N_2 \end{array}$$

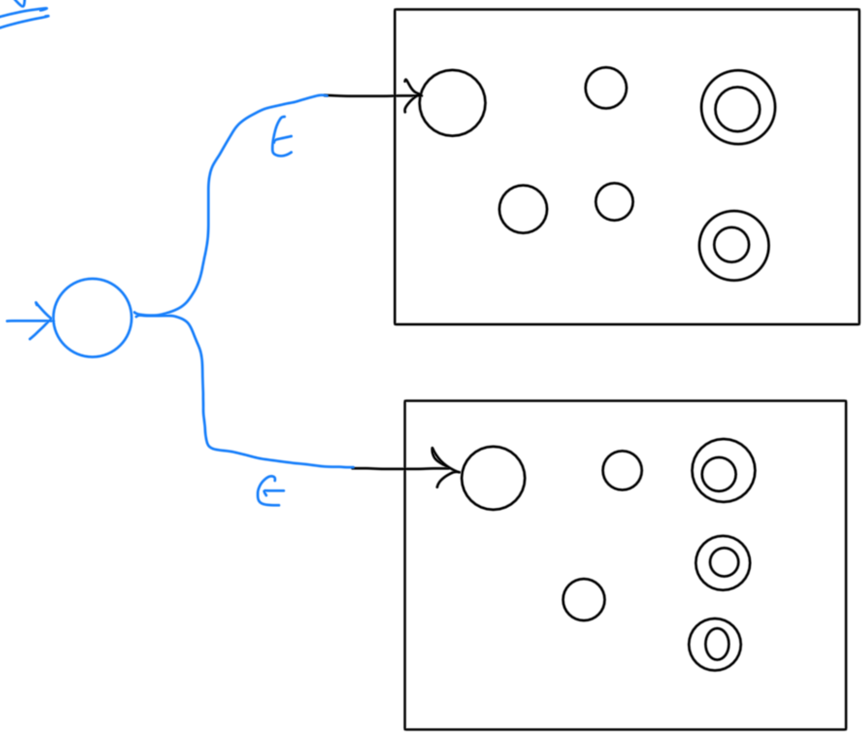


N_1



N_2

N

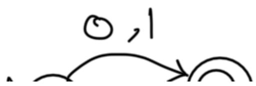


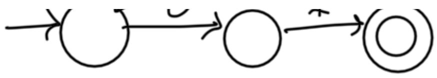
Example

$L_1 = \{ w \in \{0,1\}^* : w \text{ ends with } 01 \}$

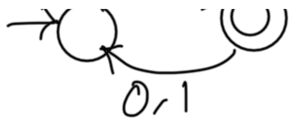
$L_2 = \{ w \in \{0,1\}^* : \text{length of } w \text{ is odd} \}$

$$\underset{\substack{\downarrow \\ N}}{L} = \underset{\substack{\downarrow \\ N_1}}{L_1} \cup \underset{\substack{\downarrow \\ N_2}}{L_2}$$



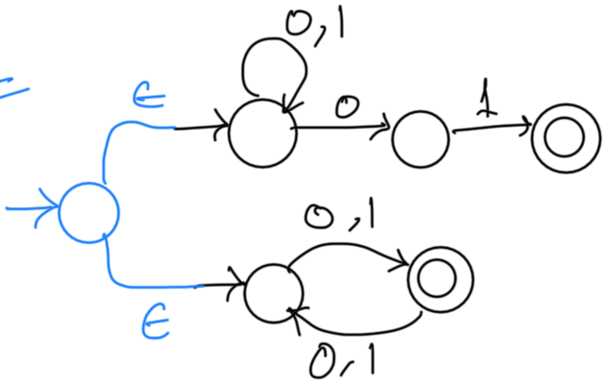


N_1



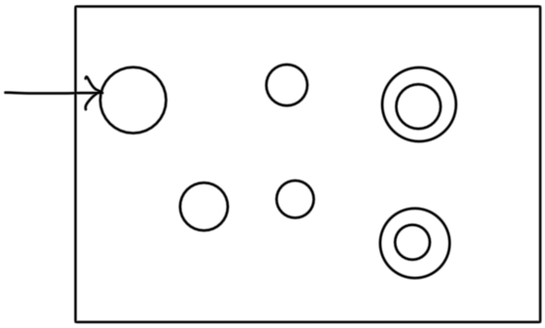
N_2

$\underline{\underline{N}}$

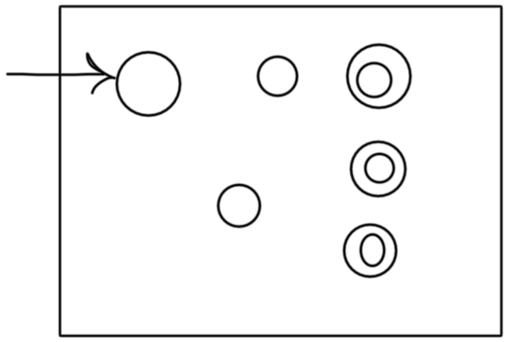


Concatenation

$$\begin{array}{ccccc} L & = & L_1 & \circ & L_2 \\ \downarrow & & \downarrow & & \downarrow \\ N & & N_1 & & N_2 \end{array}$$

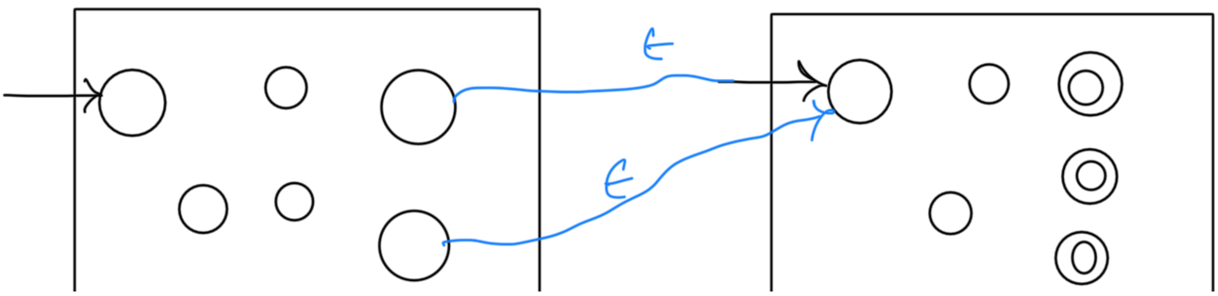


N_1



N_2

$\underline{\underline{N}}$

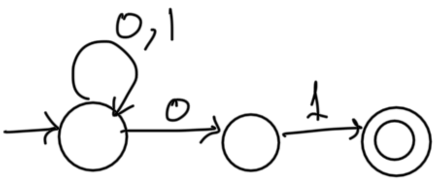


Example

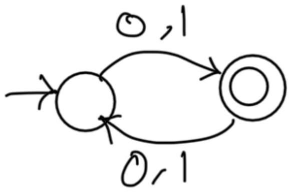
$$L_1 = \{ w \in \{0,1\}^* : w \text{ ends with } 01 \}$$

$$L_2 = \{ w \in \{0,1\}^* : \text{length of } w \text{ is odd} \}$$

$$\underset{\displaystyle N}{\underset{\downarrow}{L}} = \underset{\displaystyle N_1}{\underset{\downarrow}{L_1}} \circ \underset{\displaystyle N_2}{\underset{\downarrow}{L_2}}$$

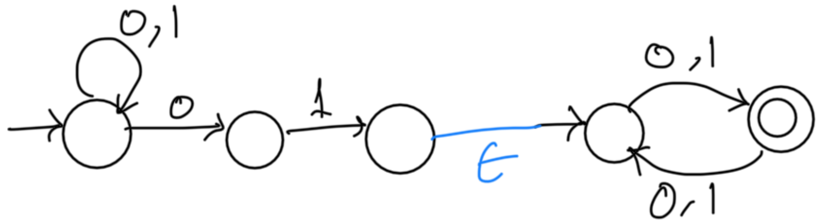


N_1



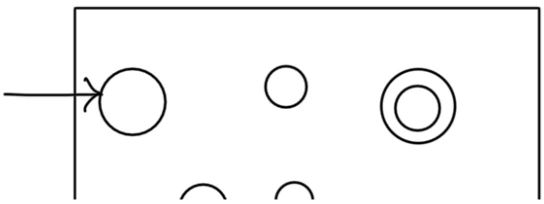
N_2

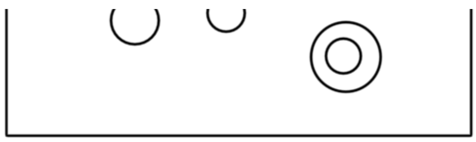
$\underline{\underline{N}}$



Star

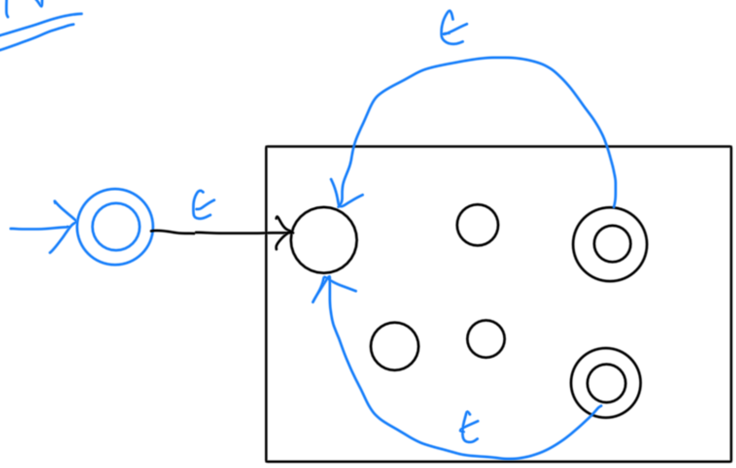
$$\underset{\displaystyle N}{\underset{\downarrow}{L}} = \underset{\displaystyle N_1}{\underset{\downarrow}{L_1}}{}^{*}$$





N_1

N



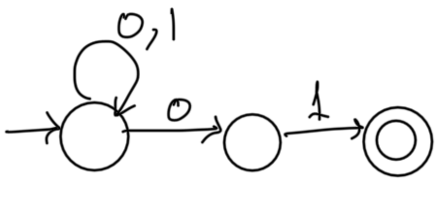
N_1

Example

$$L_1 = \{ w \in \{0,1\}^* : w \text{ ends with } 01 \}$$

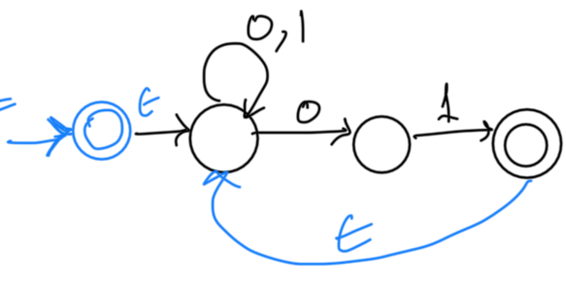
$$L = L_1^*$$

$L \to N$, $L_1 \to N_1$



N_1

N



Surprise Test

$L_1 = \{ w \in \{a,b\}^* :$ number of a's in w is one more than multiple of two $\}$

$L_2 = \{ w \in \{a,b\}^* :$ w contains "abb" as a substring $\}$

1. $L = L_1 \cup L_2$
2. $L = L_1 \circ L_2 = L_1 L_2$
3. $L = L_1^*$
4. $L = L_2^*$
5. $L = L_2 \circ L_1 = L_2 L_1$