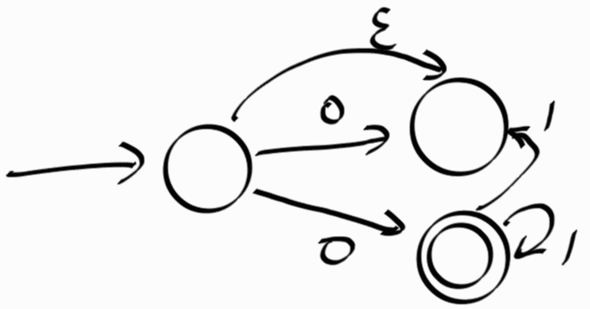
09/09/2025

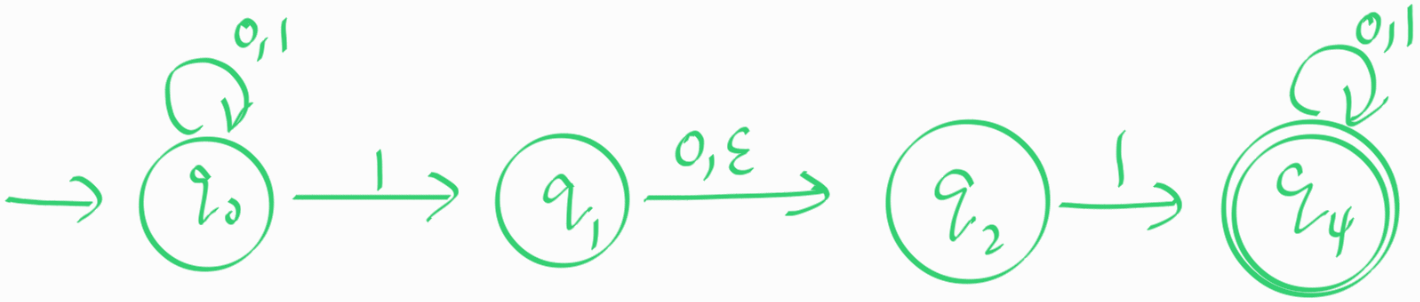
Deterministic: Given the current state and the input symbol we know exactly which state to reach to:



$\Sigma = \{0,1\}$

Not a DFA

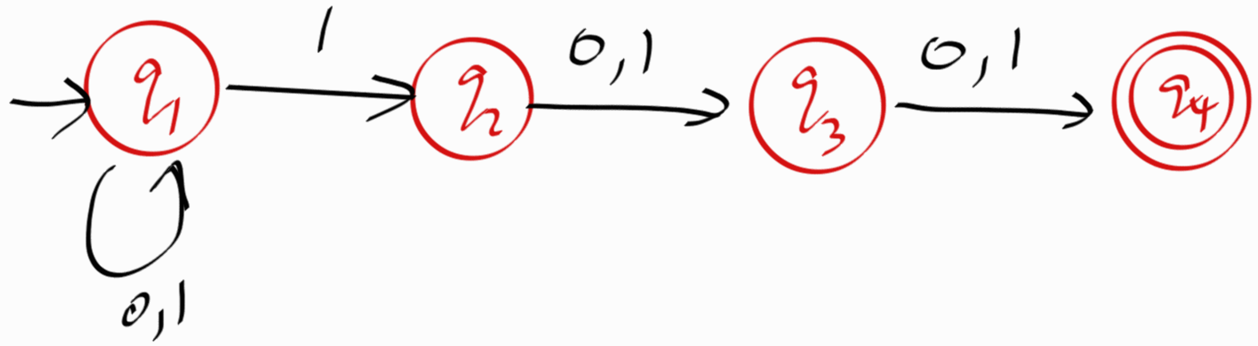
Nondeterministic



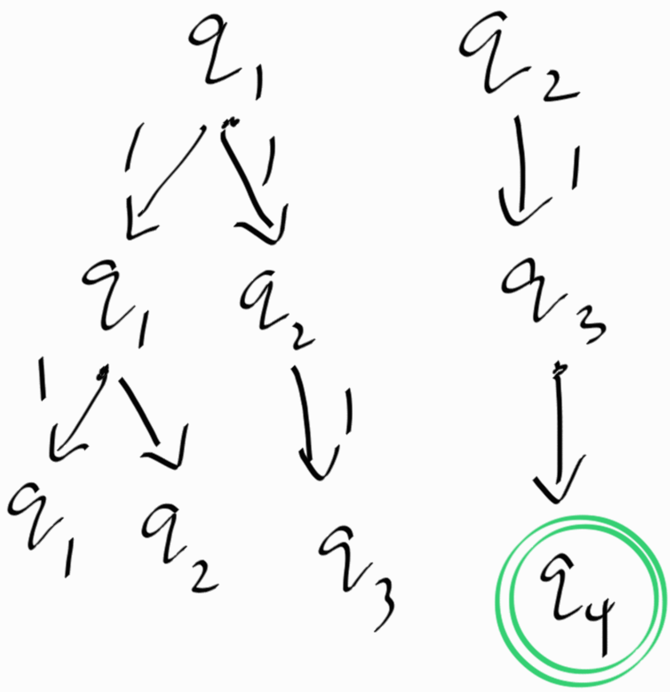
1. Many arrow from a state on the same symbol.
2. Transitions on empty string

How does NFA compute?

- If any of the accept states can be reached by reading the input x in any way, we say NFA accepts x.

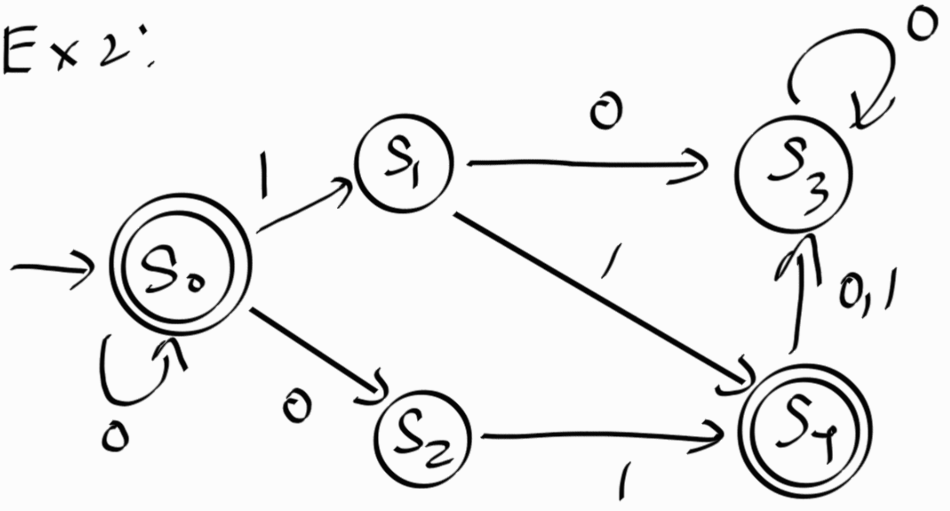


x = 111



We say 111 is accepted by this machine.

Ex 2:



0^* ✓

$0^* 1 0$ ✗

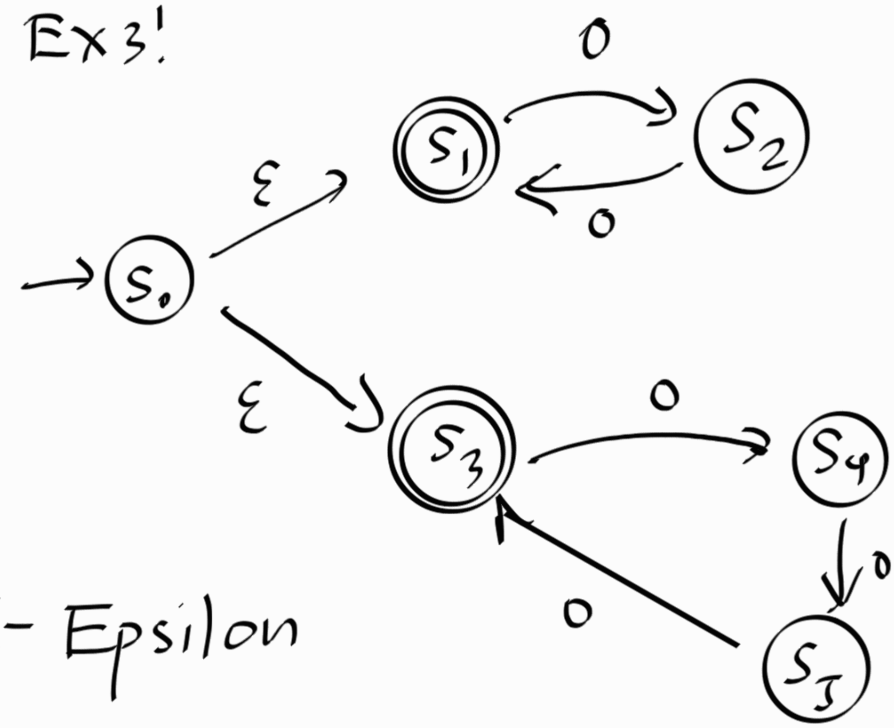
$0^* 1 1$ ✓

$0^* 1 1 (0|1)$ ✗

$0^+ 0 1$

Ex3!

$\Sigma = \{0, 1\}$



$X = 1001$

ε- Epsilon

ε = ""

$S_0 \quad S_1 \quad S_3$

X X X

$X = \underline{0}\,\underline{0}\,\underline{0}$ ✓

S_0: X

$S_1 \xrightarrow{0} S_2 \xrightarrow{0} S_1 \xrightarrow{0} S_2$

$S_3 \xrightarrow{0} S_4 \xrightarrow{0} S_5 \xrightarrow{0} S_3$

Definition of NFA.

NFA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$
s.t

1. Q is the set of states
2. Σ is a finite alphabet
3. $\delta : Q \times \Sigma_\varepsilon \longrightarrow P(Q)$

$$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$$

4. $q_0 \in Q$ is the start state.
5. $F \subseteq Q$ is the accepting states.

$$\delta(s_0, \varepsilon) = \{s_1, s_2\}$$

How does NFA accepts a string?

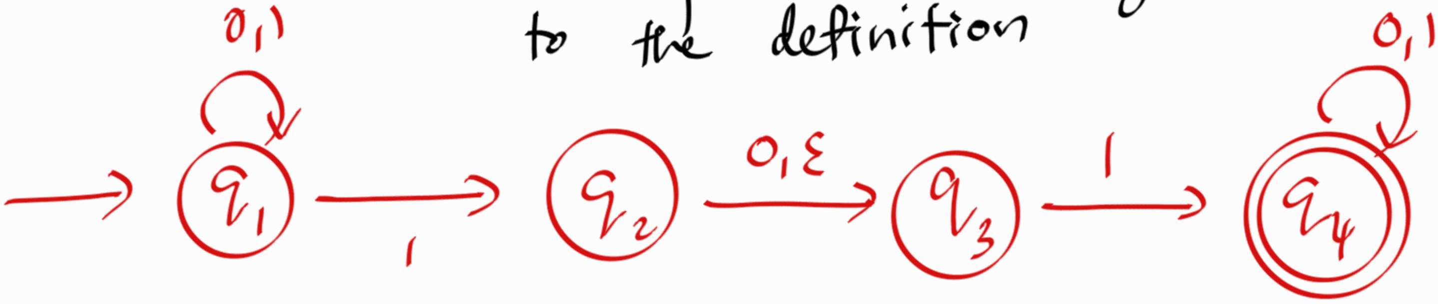
Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA, and $w = \langle w_1, w_2, w_3, \ldots, w_n \rangle$ $w_i \in \Sigma$

N accepts w if

$\exists\ r_0, r_1, r_2 \ldots\ldots, r_n \in Q$ s.t

1. $r_0 = q_0$
2. $r_i \in \delta(r_i, w_{i+1})\ \forall i \in [0 \ldots n-1]$
3. $r_n \in F$

Let's map the following NFA to the definition



1. $Q = \{q_1, q_2, q_3, q_4\}$

2. $\Sigma = \{0, 1\}$, $\Sigma_\varepsilon = \{0, 1, \varepsilon\}$

3. δ is given as follows.

	ε	0	1
q_1	$\emptyset$	$\{q_1\}$	$\{q_1, q_2\}$
q_2	$\{q_3\}$	$\{q_3\}$	$\emptyset$
q_3	$\emptyset$	$\emptyset$	$\{q_4\}$
q_4	$\emptyset$	$\{q_4\}$	$\{q_4\}$

4. $q_1 \in Q$ is the start state

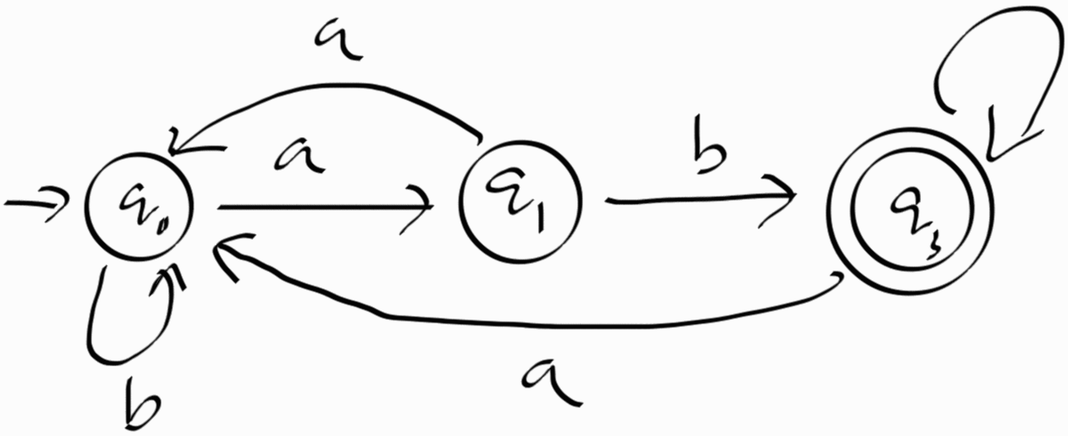
5. $F = \{q_4\}$ is the accepting states.

Let's do some examples

1. Design a DFA that accepts the following language.

$C = \{ w \mid w \text{ has an odd \# of a's, and ends with } b \}$

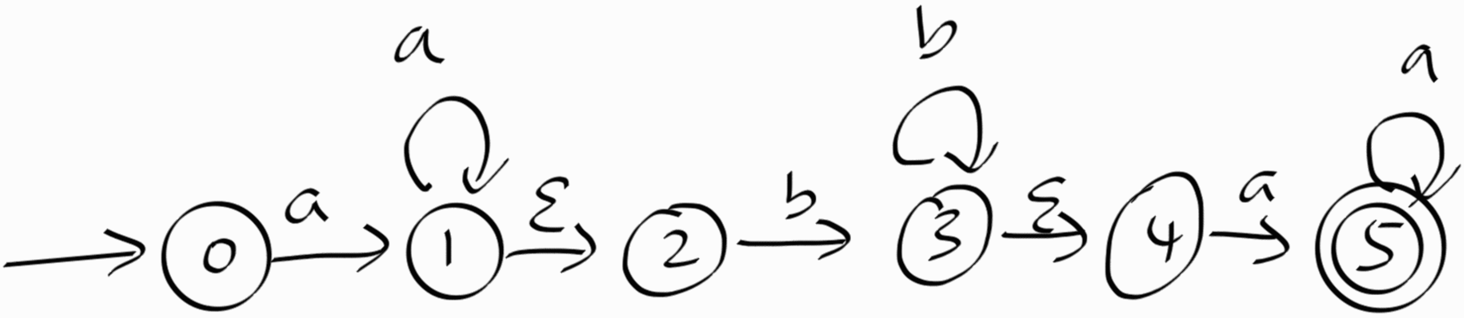
$\Sigma = \{a, b\}$

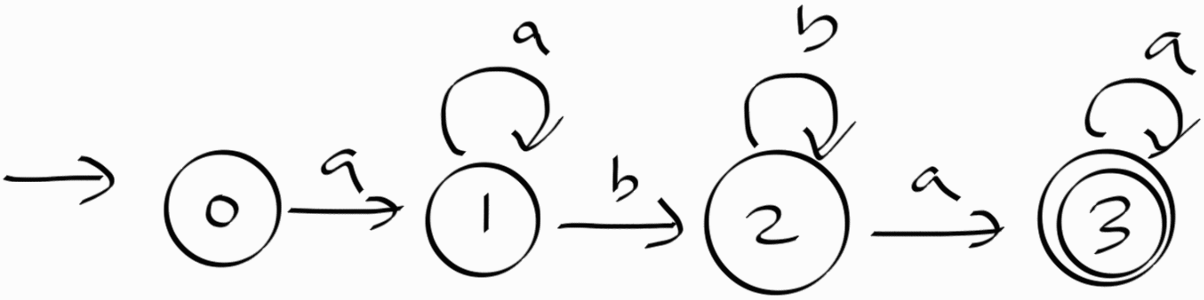


aaa

Ex2: Design an NFA for

$D = \{a^+ b^+ a^+\}$ $\Sigma = \{a, b\}$

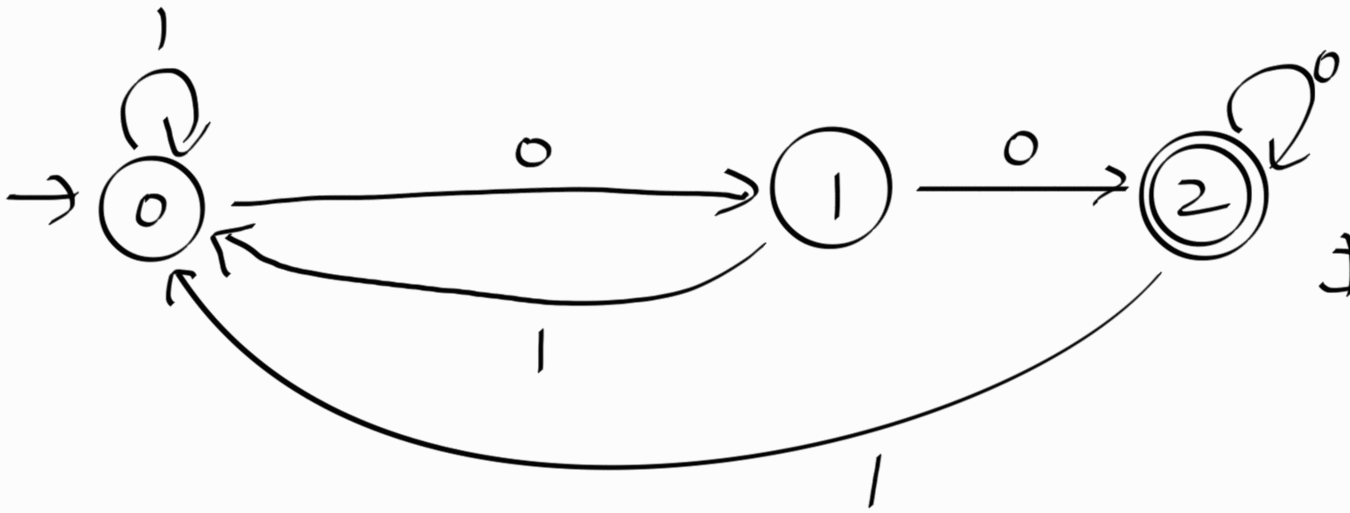




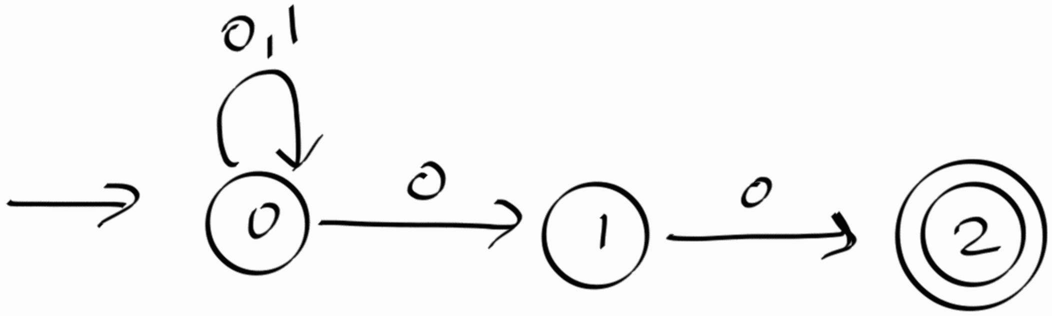
Ex5: Design an NFA for

$E = \{ w \mid w \text{ ends with } 00 \}$

$\Sigma = \{0, 1\}$



DFA



000

010111 00

1001

Definition

Two machines are equivalent if they accept the same language.

Theorem 1.39

Every NFA has an equivalent DFA.

Proof: Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA that recognize language A.

We'll construct a DFA M, $M = (Q', \Sigma, \delta', q_0', F')$

a) No ε- transitions in N.

1. $Q' = P(Q)$
2. $\Sigma = \Sigma$
3. $\forall R \in Q', a \in \Sigma$, $\delta'(R,a) = \bigcup_{r \in R} \delta(r,a)$
4. $q_0' = \{q_0\}$
5. $F' = \{R \in Q' \mid R \text{ contains an accept state in } N\}$

The basic idea for using the powerset is to simulate every possible execution path an NFA can take.

b) what if NFA N has ε-arrows?

we define $E(R)$ for $R \subseteq Q$ as the collection of states that we can reach from R by going along ε arrows, including members of R as well.

$E(R)$ is called ε-closure.

with this updated notation, δ function should be redefined.

$$\delta'(R,a) = \{ q \in Q \mid q \in E(\delta(r,a)) \text{ for some } r \in R \}$$

Also, we need to change the start state as well.

$$q_0' = E(\{q_0\})$$

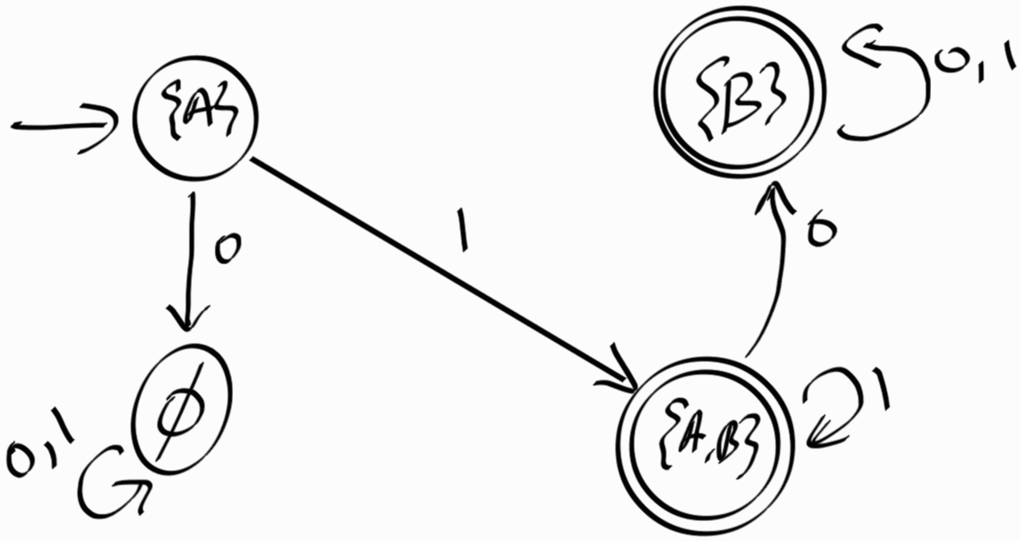
Let's look at an example

NFA N:

An example w.o ε-transitions

DFA M: $Q' = P(Q)$

$Q' = \{\emptyset, \{A\}, \{B\}, \{A,B\}\}$



$$\delta'(\{A\},0) = \bigcup_{r\in\{A\}} \delta(r,0) = \emptyset$$

$$\delta'(\{A\},1) = \bigcup_{r\in\{A\}} \delta(r,1)$$

$$\delta'(\{B\},0) = \bigcup_{r\in\{B\}} \delta(r,0) = \{B\}$$

$$\delta'(\{B\},1) = \bigcup_{r\in\{B\}} \delta(r,1) = \{B\}$$

$$\delta'(\{A,B\},0) = \bigcup_{r \in \{A,B\}} \delta(r,0) = \{\} \cup \{B\}$$

$$= \{B\}$$

$$\delta'(\{A,B\},1) = \bigcup_{r \in \{A,B\}} \delta(r,1) = \{A,B\} \cup \{B\}$$

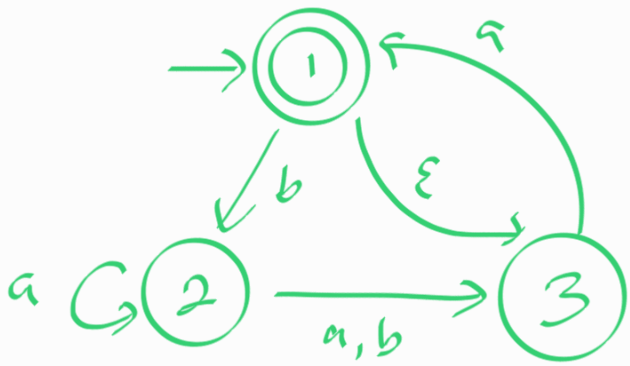
$$= \{A,B\}$$

$$\delta(\phi,0) = \bigcup_{r \in \phi} \delta(r,0) = \phi$$

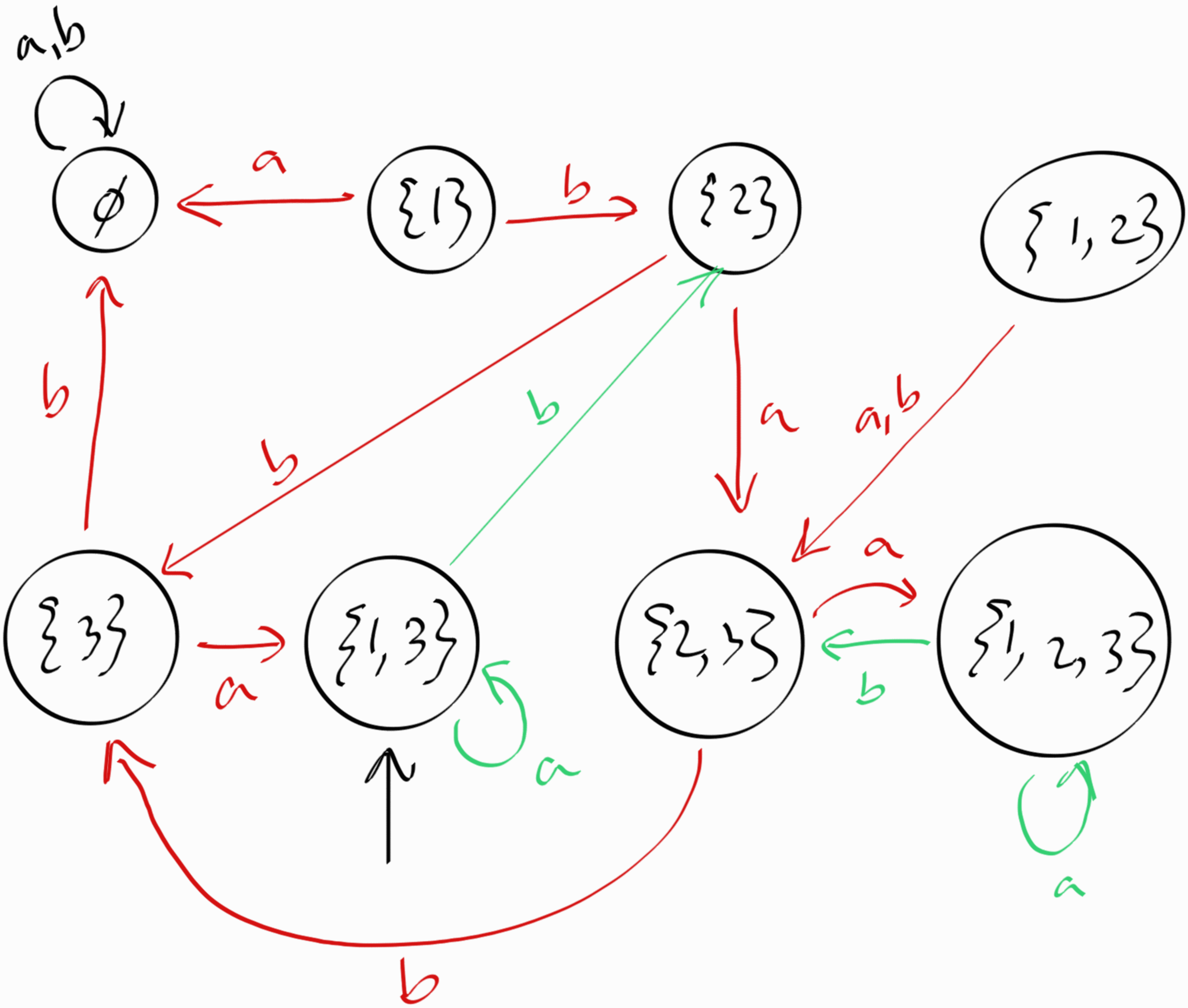
$$\delta(\phi,1) = \bigcup_{r \in \phi} \delta(r,1) = \phi$$

Let's try to understand $E(R)$

NFA N:



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

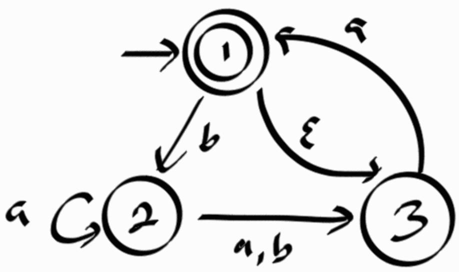


$$\delta'(R,a) = \{ q \in Q \mid E(\delta(r,a)) \text{ for } r \in R \}$$

or

$$\delta'(R,a) = \bigcup_{r \in R} E(\delta(r,a))$$

We will do only some calculations.



$$R = \{1, 3\}$$

$$\begin{aligned} \delta'(R,a) &= E(\delta(1,a)) \cup E(\delta(3,a)) \\ &= E(\emptyset) \cup E(\{1\}) \\ &= \emptyset \cup \{1,3\} = \{1,3\} \end{aligned}$$

$$\begin{aligned} \delta'(R,b) &= E(\delta(1,b)) \cup E(\delta(3,b)) \\ &= E(\{2\}) \cup E(\emptyset) \\ &= \{2\} \cup \emptyset = \{2\} \end{aligned}$$

$R = \{1, 2, 3\}$

$$\delta'(R, a) = E(\delta(1,a)) \cup E(\delta(2,a)) \cup E(\delta(3,a))$$
$$= \phi \cup E(\{2,3\}) \cup E(\{1\})$$
$$= \{1,2,3\}$$

$$\delta'(R, b) = E(\delta(1,b)) \cup E(\delta(2,b)) \cup E(\delta(3,b))$$
$$= E(\{2\}) \cup E(\{3\}) \cup E(\phi))$$
$$= \{2,3\}$$

You need to calculate these for each state.

Red transitions are the one's that I did not calculate here.