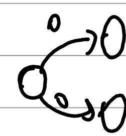
Thm concatenation ($\circ$) is closed under Regular Language.

proof by building an Non-deterministic Finite Automaton. (NFA)

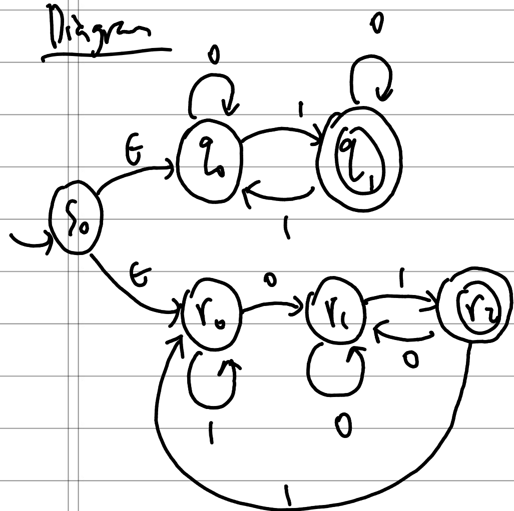
Def An NFA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where

1) Q is a finite set
2) Σ is a finite alphabet
* 3) $\delta : Q \times \Sigma_\epsilon \to P(Q)$ ← power set
4) $q_0 \in Q$ is the initial state
5) $F \subseteq Q$ is the set of accepting states.

① ϵ (empty) transition
② out-going edges of a state can have duplicate symbol 

Diagram

> example $\boxed{A \cup B}$



Symbolic

$$A = (Q, \Sigma, \delta_1, q_0, F_1)$$

$$B = (R, \Sigma, \delta_2, r_0, F_2)$$

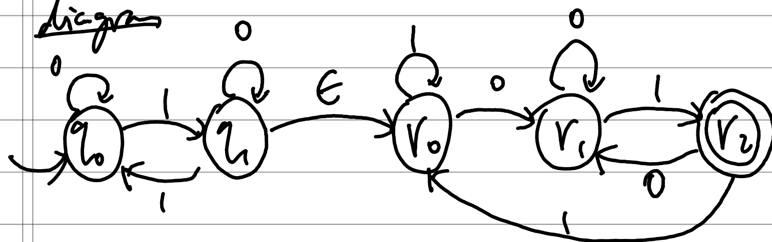
NFA of $A \cup B = (\{s_0\} \cup Q \cup R, \Sigma, \delta, s_0, F_1 \cup F_2)$

δ is derived by

$$\delta \triangleq \begin{cases} \delta(s_0, \epsilon) = \{q_0, r_0\} \\ \delta(s_0, a) = \emptyset \text{ (unaccepted)} & \forall a \neq \epsilon \\ \delta(q, a) = \delta_1(q, a) & \forall q \in Q \\ \delta(q, a) = \delta_2(q, a) & \forall q \in R \end{cases}$$

> example $A \circ B$

diagram



Symbolic

$A = \{Q, \Sigma, \delta_1, q_0, F_1\}$

$B = \{R, \Sigma, \delta_2, r_0, F_2\}$

start from A

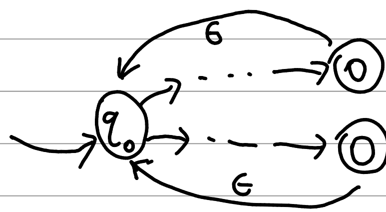
NFA for $A \circ B = \{Q \cup R, \Sigma, \delta, q_0, F_2\}$

end in B

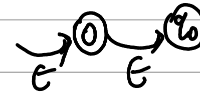
$$\delta(q,a) \triangleq \begin{cases} \delta_1(q,a) & \text{if } q \in Q / F_1 \\ \delta_1(q,a) & \text{if } q \in F_1 \wedge a \neq \epsilon \\ \delta_1(q,a) \cup \{r_0\} & \text{if } q \in F_1 \wedge a = \epsilon \\ \delta_2(q,a) & \text{if } q \in R \end{cases}$$

> example: A^*

δ is defined by connecting all the accepting states with the initial state by ϵ transition.

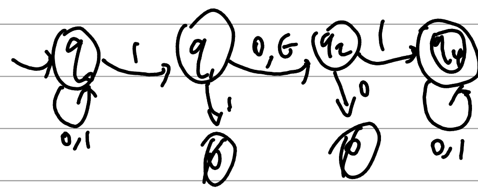


also, $\epsilon \in A^*$



---

example 010110



1) $q_0 q_0 q_0 q_0 q_0 q_0$

2) $q_0 q_0 q_0 q_0 q_1 q_2$

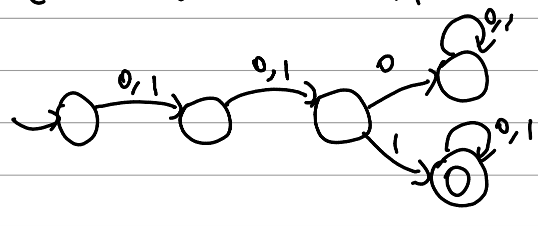
3) $q_0 q_0 q_0 q_0 q_1 q_4$

4) $q_0 q_0 q_0 q_1 \emptyset$

8) $q_0 q_0 q_0 q_1 q_2 q_4 q_4$

...

exercise Design a FA to recognize the language consist of all strings of 0's & 1's, that contain a 1 in the third position.

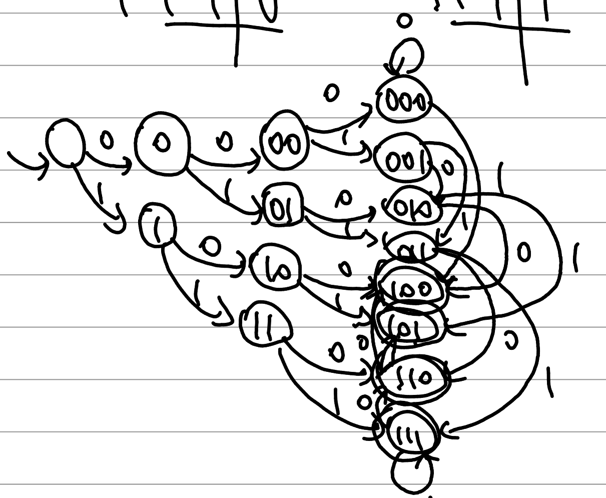


a 1 in the 3rd position from the end of the string.

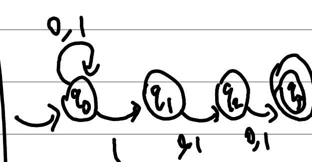
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

DFA



NFA

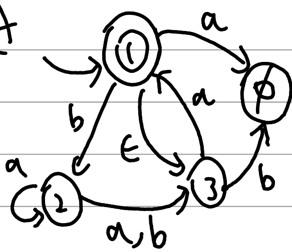


Thm If a language is recognized by a NFA, then it is recognized by a DFA.

* Moreover, the proof is constructive based on an algorithm for converting a NFA to a DFA.

> example

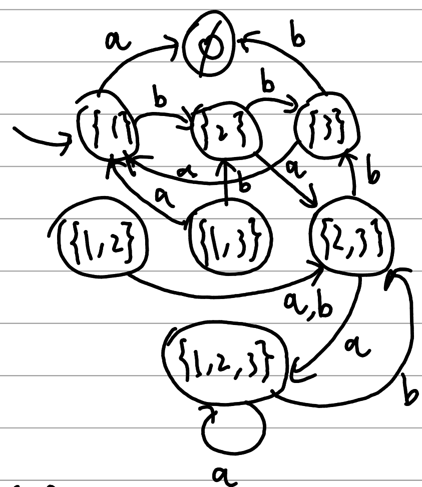
NFA



$\Longleftrightarrow$

DFA (ignoring ϵ transition)

$Q = P(\{1,2,3\})$



Strategy

$$\left.\begin{array}{l} 1 \xrightarrow{a} \emptyset \\ 2 \xrightarrow{a} \{2,3\} \end{array}\right\} \Rightarrow \{1\} \cup \{2\} \xrightarrow{a} \emptyset \cup \{2,3\}$$

$$\Leftrightarrow \{1,2\} \xrightarrow{a} \{2,3\}$$

denotes transitions from 1 or 2 lands at 2 or 3.

"proof" NFA: $N = \{Q, \Sigma, \delta, q_0, F\}$

DFA: $D = \{\mathcal{P}(Q), \Sigma, \delta', q_0', F'\}$

where $F' = \{R \in \mathcal{P}(Q) \mid R \cap F \neq \emptyset\}$,

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

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

union of all states connected by ϵ transition
from state $\{q_0\}$
and including $\{q_0\}$

$$E(q) = \{q\} \cup \delta(q, \epsilon)$$

Section 1.3 Regular Expression

build using regular operators | union $(\cdot \cup \cdot)$
concatenate $(\cdot \circ \cdot)$
Kleene star $(\cdot^*)$

precedence of operators: $(*) > (\circ) > (\cup)$

Def Given an alphabet Σ, R is a regular expression over Σ if
(recursive/constructive definition)

base cases
1) $R \in \Sigma$
2) $R = \epsilon$
3) $R = \emptyset$

induction steps
4) $R = R_1 \cup R_2$ where R_1 and R_2 are both regular expression
5) $R = R_1 \circ R_2$ where " "
6) $R = R_1^*$ where " "

7) Nothing else is a regular expression.

> example

1) 0^*10^*

strings of 0 and 1 with exactly one 1.

2) $\Sigma^*1\Sigma^*$

$(\{0,1\}^* 1 \{0,1\}^*)$

" " w/ <u>at least</u> one 1.

3) $(\Sigma\Sigma)^*$

$((\{0,1\}\{0,1\})^*)$

" " with even length.

5) Contain 001 as a substring

$$\Sigma^*(001)\Sigma^*$$

6) start & end with the same symbol

$$0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1$$

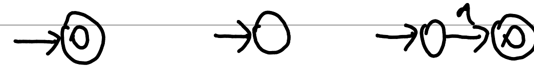
<u>Thm</u> If a language is regular (FA)

$\Longleftrightarrow$ iff there exists a regular expression that describes it.

(ReExp)

<u>Proof</u> 1) FA $\Rightarrow$ ReExp (hard)

2) FA $\Leftarrow$ ReExp (easy, by composition of "unit" modules)



$R=\varepsilon$ $\quad$ $R=\emptyset$ $\quad$ $R=a$