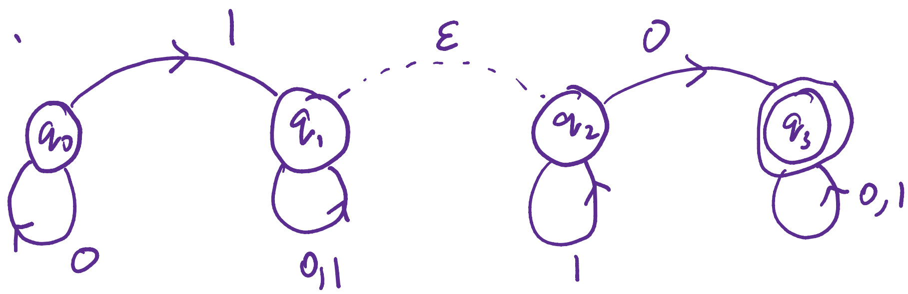
# Nondeterminism, Regular Languages

Tuesday, August 27, 2019 7:21 PM

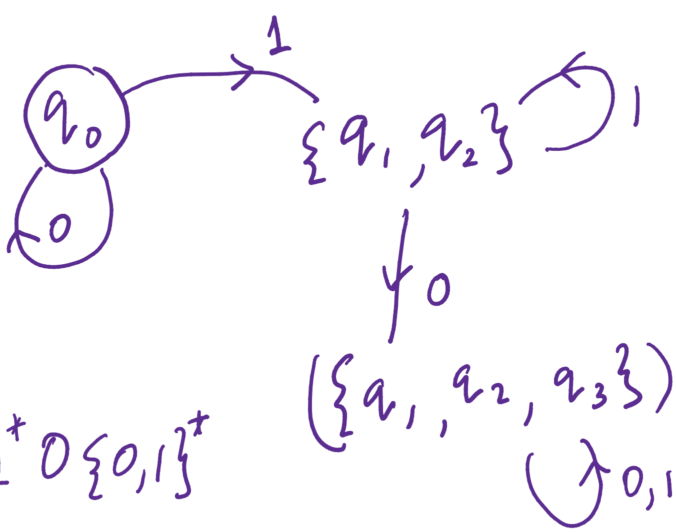
| DFA | NFA |
| --- | --- |
| $Q, \Sigma$ | $Q, \Sigma$ |
| $\delta(q,a) \to q'$ | $\delta(q,a) = \{q_1, q_2, \dots\}$ OR $\delta(q,a) \to q'$, $\delta(q,\varepsilon) \to q''$. |
| $q_0$ | $q_0$, $F \subseteq Q$. |
| $F \subseteq Q$ | |

e.g.




Thm. Every NFA can be converted to a DFA.

e.g.




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

---

**General construction.**

General construction:

Given $Q, \Sigma, \delta, q_0, F$

DFA: $Q' = 2^Q$

$R \subseteq Q$ $\quad \delta'(R, a) = \{q' \mid \exists q \in R \ \delta(q, a) = q'\}$

$q_0' = \{q_0\}$

$F' = \{R \subseteq Q \mid \exists q \in R \cap F\}$

To add $\varepsilon$ transitions, define

$$\mathcal{E}(R) = \{q \mid q \in R \text{ or } \exists q' \in R \text{ and } \delta(q', \varepsilon) = q\}$$

"$\varepsilon$-extension" of $R$.

$$\delta'(R) = \mathcal{E}(\{q \mid \exists q' \in R, \delta(q', a) = q\})$$

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

$$F' = \mathcal{E}(\{R \subseteq Q \mid \exists q \in R \cap F\})$$

---

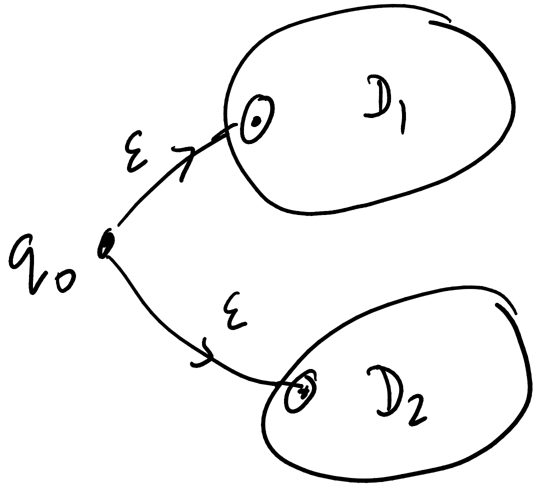
- Every valid sequence of transitions is allowed in the new DFA in the new DFA

- All accepting paths remain accepting paths.

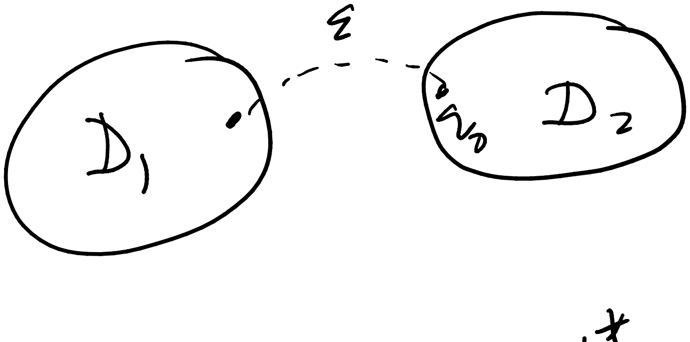
---

Non determinism is <u>convenient</u> but not more powerful.

e.g. NFA that accepts $L_1 \cup L_2$



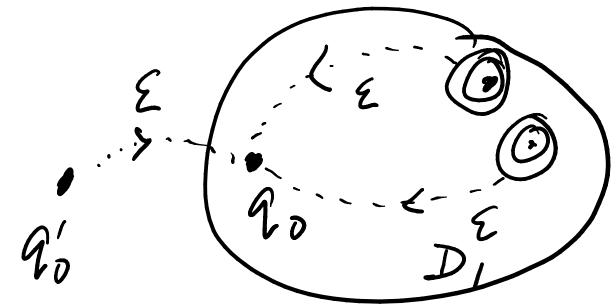

NFA that accepts $L_1 \cdot L_2 = \{ab \mid a \in L_1, b \in L_2\}$




NFA that accepts $L^*$

e.g. $L = \{aa, bc\}$ $L^* = \{\_, aa, bc, aaaa, aabc,$

(e.g. $L = \{aa, bc\}$ $L^* = \{\_, aa, bc, aaaa, \ldots,$
$bcbc, \ldots \}$ )




---

Regular expressions.

$R = \emptyset$ — empty language.

$R = a$ $\quad \forall a \in \Sigma$

$R = \{\varepsilon\}$ $\rightarrow$ language with empty string

$\emptyset$ vs $\{\emptyset\}$

$R = (R_1 \cup R_2)$

$R = R_1 \circ R_2$

$R = R_1^*$

---

Ex. Reg expression for binary strings with an even # 1's ?

$(0^*10^*1)^*0^*$

$(0^*10^*1)0^*$

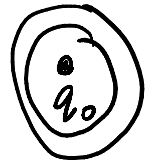
---

**Thm.** $L$ is regular $\Longleftrightarrow$ $\exists$ regular expression for $L$.

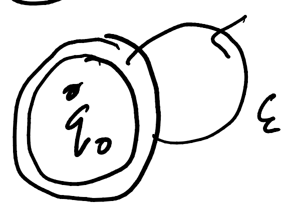
**Pf.** Regular expression $\Rightarrow$ NFA/DFA

Recall rules.

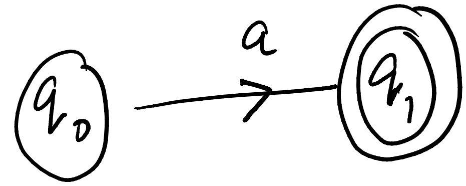
① $R = \phi$



② $R = \{\varepsilon\}$



③ $R = a$



④ $R = (R_1 \cup R_2)$ ✓

⑤ $R = R_1 \cdot R_2$ ✓

⑥ $R = R_1^*$ ✓

Other direction also elementary but more tedious. (see book).

---

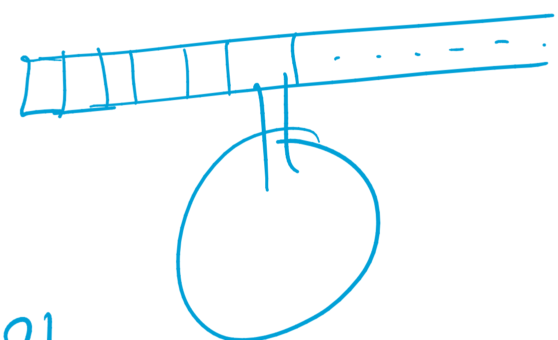
Regexp.

## What about NTMs?

$Q, \Sigma$

$\delta$ : list of allowed transitions
can be more than 1 for $(q, a)$

$q_0$



$F$.

More powerful than TMs?!

NTM accepts iff $\exists$ valid computation path leading to an accept state.

(P) $\exists$ path from $s$ to $t$ in $G = (V, E)$?

Deterministic: BFS, DFS

Deterministic : BFS, DFS

Non deterministic : Guess next vertex!
(all edges are valid transitions).

P $\exists$ Hamilton cycle (visits all vertices once) in $G$?

P $\exists$ TSP of length $\leq L$ given a graph (map) with edge lengths.

---

Languages recognized by TMs are called "Recursively Enumerable".