Chap. 2
Finite Automata

유한

Agenda of Chapter 2

- Deterministic Finite Accepters (DFA)
- Nondeterministic Finite Accepters (NDFA)
- Equivalence of DFA & NDFA

Deterministic Accepters & Transition Graph (1/3)

- Deterministic finite accepter (dfa)

모든 오토마타의 설명

$$\underset{\text{Machine}}{M} = (Q, \Sigma, \delta, q_0, F)$$

  - Q : a finite set of internal state (유한한 state) $Q = \{q_0, q_1\}$
  - Σ : a finite set of symbols, input alphabet $\Sigma = \{0, 1\}$
  - $\delta : Q \times \Sigma \rightarrow Q$: a total function, transition function

    모든 들어오는 값에 출력이 정해져 있다.

    입력을 받아서 어떻게 움직이나.
  - $q_0 \in Q$: initial state
  - $F \subseteq Q$: a set of final states $F = \{q_1\}$

domain. $Q \times \Sigma$

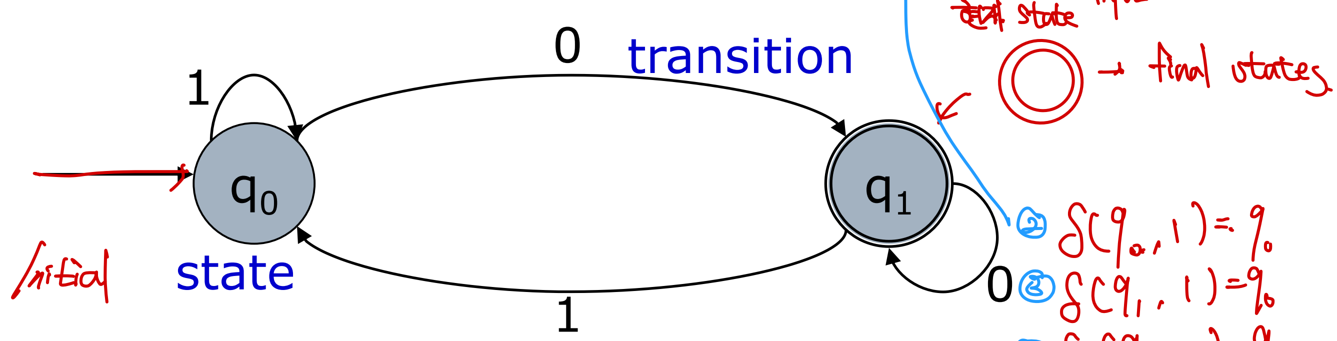
① $\delta(q_0, 0) = q_1$ (현재 state, input) range. co domain Q

② $\delta(q_0, 1) = q_0$

③ $\delta(q_1, 1) = q_0$

④ $\delta(q_1, 0) = q_1$

- Transition graph



Deterministic Accepters & Transition Graph (2/3)

Ex 2.1]$M = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_1\})$

finite state set

$\delta(q_0,0) = q_0, \quad \delta(q_0,1) = q_1,$

$\delta(q_1,0) = q_0, \quad \delta(q_1,1) = q_2,$

$\delta(q_2,0) = q_2, \quad \delta(q_2,1) = q_1.$

Transition

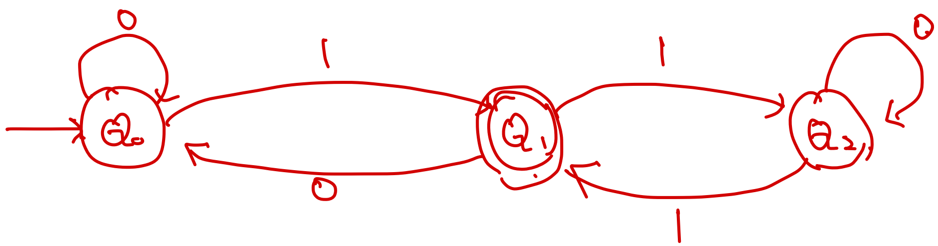


Table representation

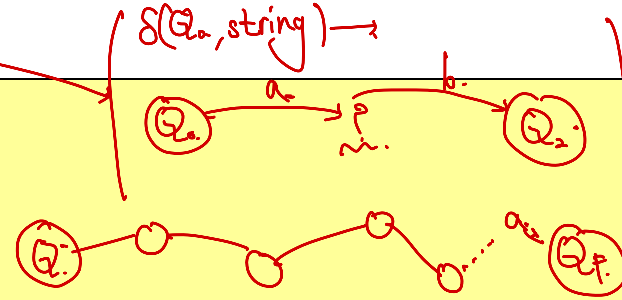
	0	1
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_2	q_1

Deterministic Accepters & Transition Graph (3/3)

- Extended transition function δ^* : $Q \times \Sigma^* \rightarrow Q$

string

Ex) $\delta^* (q_0, ab) = q_2$



[Recursive definition of δ^*]

1) $\delta^* (q, \lambda) = q$ basis

2) $\delta^* (q, wa) = \delta (\delta^* (q, w), a)$ (w는 string)

Ex) $\delta^* (q_0, ab) =$

$\delta(\delta^*(q_0, a), b) = \delta(\delta(\delta^*(q_0, \lambda), a), b) = \delta(\delta(q_0, a), b)$

by 1) $\therefore q_0$

$\delta^*(q, wa) = \delta(\delta^*(q, w), a)$

[THEOREM 2.1] Extended transition of DFA & strings

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

G_M : associated transition graph

For every $q_i, q_j \in Q$ and $w \in \Sigma^+$, $\delta^* (q_i, w) = q_j$

extended transition

$\Leftrightarrow$ There is in G_M a walk with label w from q_i to $q_j \in Q$

Languages and Dfa's (1/2)

"오토마타에 의해 accept 된 결과."

- Language
  - Set of all the strings accepted by the automaton.

$\{Q_1\}$

- Language accepted by dfa $M = (Q, \Sigma, \delta, q_0, F)$ finite state.
  - Set of all strings on accepted by M
  - $L(M) = \{w \in \Sigma^* \mid \boxed{\delta^*(Q_0, w) \in F} \}$

$L(G) \rightarrow$ Granmer 에 의해 생성된 /

$\rightarrow Q_0 \xrightarrow{w} Q_1$

$\delta^*(Q_0, w) \in F$

$\{Q_1\}$

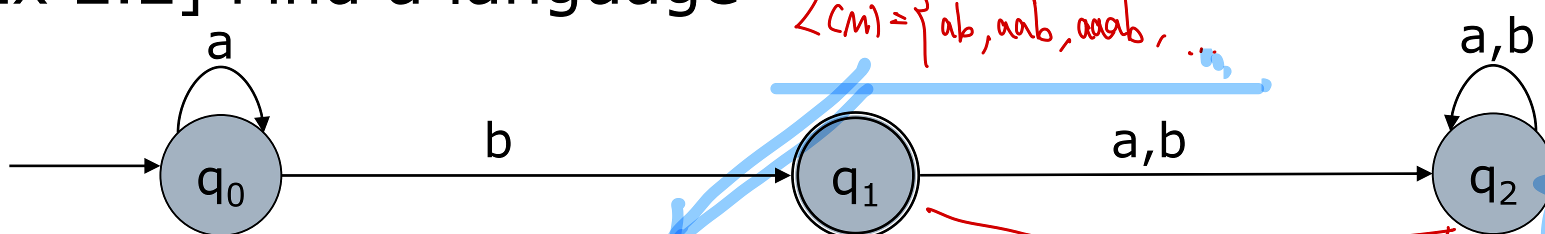
- Nonacceptance
  - $(L(M))^c = \{w \in \Sigma^* \mid \boxed{\delta^*(Q_0, w) \notin F} \}$

↳ reject.

$F = \{Q_1\}$.

Ex 2.2] Find a language

$L(M) = \{ab, aab, aaab, \ldots$



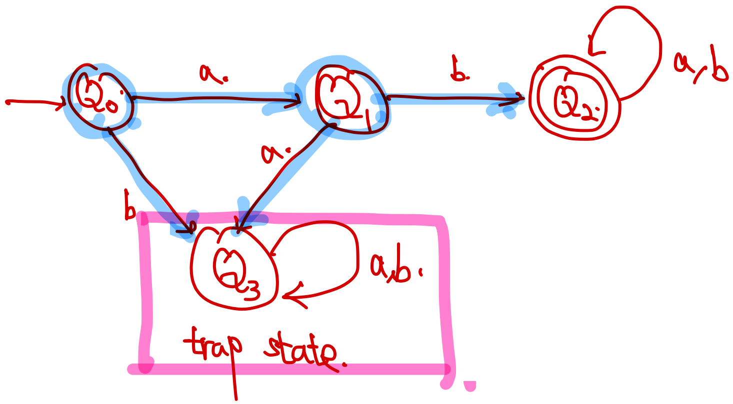
$L(M) = \{a^n b \mid n \geq 0\}$.

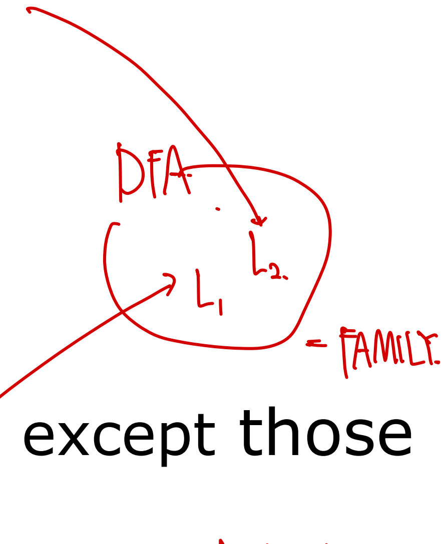
Languages and Dfa's (2/2)

δ, total function

Ex2.3] A dfa recognizing L={abw| w∈ Σ* }, Σ={a,b}

total function

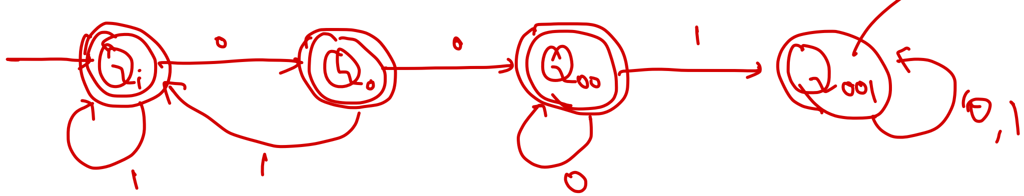




Ex2.4] A dfa accepting all strings on {0,1} except those containing 001

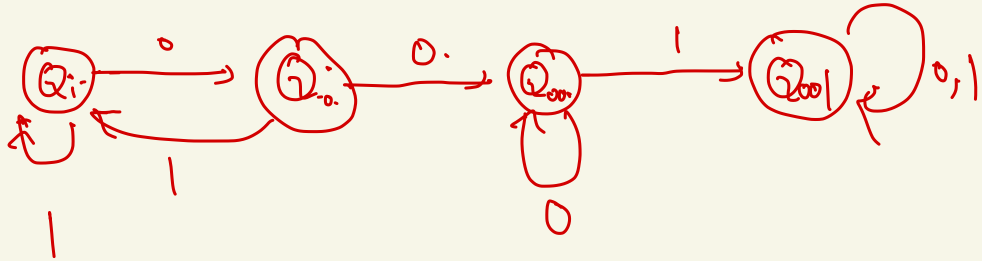
→ 001 만 포함하면 다 accept.

$M = (\{Q_i, Q_0, Q_{00}, Q_{001}\}, \{0,1\}, \delta, Q_i, \{Q_i, Q_0, Q_{00}\})$



0을 처리하면 된다.

001



Regular Languages

(DFA ⟺ Regular Language.)

- Family
  - a set of languages with a common characteristics
  - a set of languages accepted by a set of automata

nfa → dfa

[Definition] Regular Language (Family) DFA가 존재하는 Language.

- A language L is regular iff There exist some dfa M.
- L = L(M)

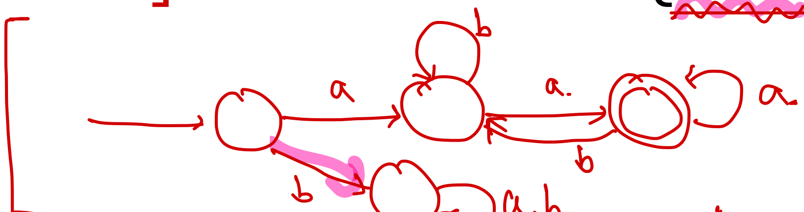
→ DFA로 그릴 수 있으면. dfa.

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

앞 a 끝 a.

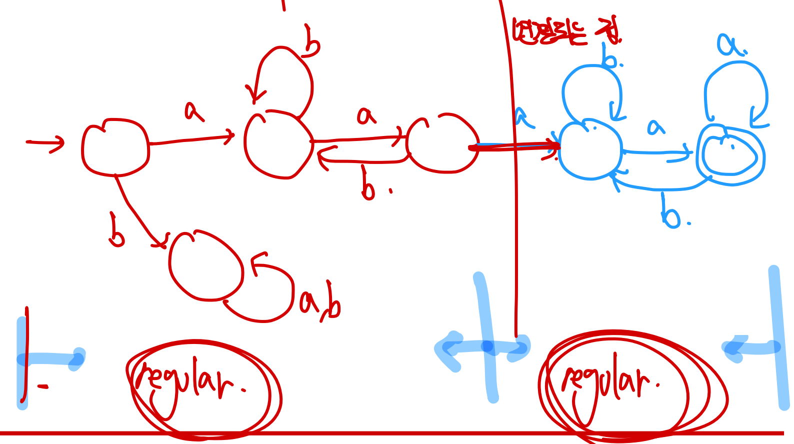
Ex2.5] Show that $L=\{awa \mid w\in\{a,b\}^*\}$ is regular

└> 얘를 accept 하는 DFA를 찾으면 됨.



Ex2.6] Show that L^2 is regular, [2번 concate]

$L^2=\{aw_1aaw_2a \mid w_1\in\Sigma^*, w_2\in\Sigma^*\}$



Definition of a Nondeterministic Accepter (1/4)

[Definition] Nondeterministic finite aceppter (NFA)

$\delta : Q \times (\Sigma \cup \{\lambda\}) \rightarrow 2^Q$

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

- Q : a finite set of internal state
- Σ : a finite set of symbols, input alphabet
- $q_0 \in Q$: initial state
- $F \subseteq Q$: a set of final states

DFA 와 동일.

(멱집합)

"Q의 부분집합의 집합."

- $\delta : Q \times (\Sigma \cup \{\lambda\}) \rightarrow 2^Q$: transition function, partial funciton

$\delta(Q_0, a) = \{Q_i, Q_j\}$

↳ 구체적인 symbol 이 없어도 전이가 될 수 있다.

$\delta : Q \times \Sigma \rightarrow Q$: DFA

그 다음 state가 정확히 결정되어있지 않다.

□ Characteristics of NFA

- Range of δ : not a single element, but a subset of Q
- λ : can be a second argument of δ ← input 없이 transition 가능
- $\delta(q_i, a)$: may be empty (partial function)

↳ 치역 x

"$\delta(Q_i, a) = \emptyset$ 이 가능."

□ A string is accepted by an nfa

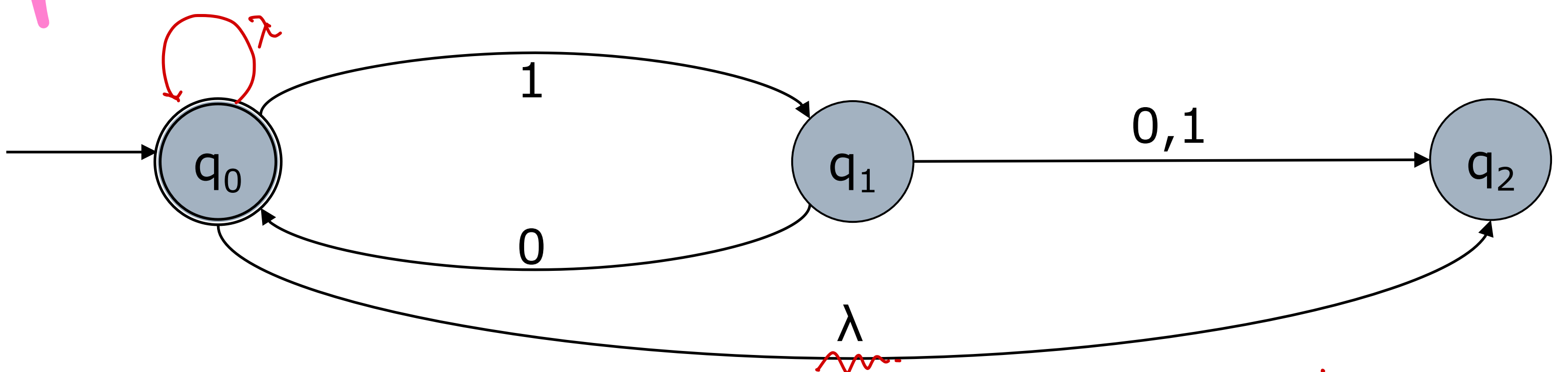
- if there is some sequence of possible moves that ends in a final state

하나가 accept 되면

무시

Definition of a Nondeterministic Accepter(2/4)

Ex2.8] An nfa shown as a transition graph



$Q_0 \rightarrow Q_2$도 있어서 조심해야 함.

$\delta(q_0, \lambda) = \{Q_2, Q_0\}$. $\quad \delta(q_0, 0) = \emptyset \quad \delta(q_0, 1) =$

$\delta(q_1, \lambda) = \quad \delta(q_1, 0) = \{Q_2, Q_0\} \quad \delta(q_1, 1) =$

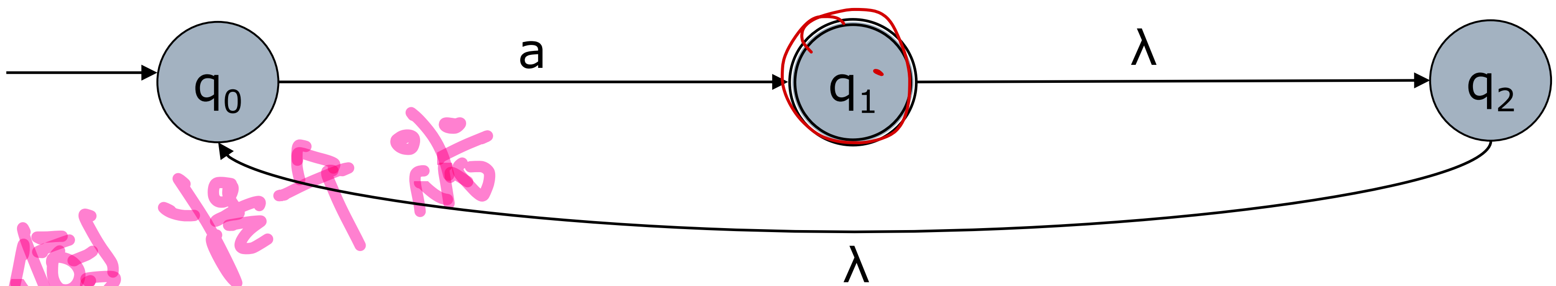
$\delta(q_2, \lambda) = \quad \delta(q_2, 0) = \quad \delta(q_2, 1) =$

Definition of a Nondeterministic Accepter(3/4)

- Extended transition
  - $\delta^* (q_i, w)$ contains q_j iff there is a walk from q_i to q_j labeled w

어떤 집합이 된다.

Ex2.9] An nfa shown as a transition graph



λλaλλa.

- $\delta^* (q_1, a) =$ {Q2, Q1, Q0}.
- $\delta^* (q_2, \lambda) =$ {Q2, Q0}. → 자기 자신도 포함해야 함.
- $\delta^* (q_2, aa) =$ {Q1, Q2, Q0}

(λaλλa.) Q1 자동으로 Q2, Q0

Definition of a Nondeterministic Accepter(4/4)

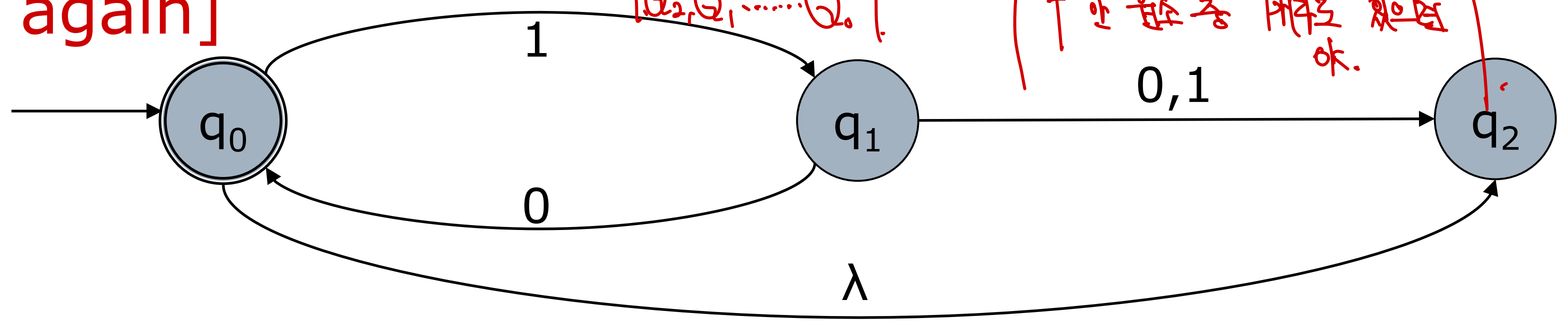
NFA: final에 1개라도 final 1번 accept

- A string is accepted by an NFA
  - if there is some sequence of possible moves that ends in a final
- Language L accepted by an NFA $M = (Q, \Sigma, \delta, q_0, F)$
  - $L(M) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \cap F \neq \emptyset\}$

$\{Q_2, Q_1, \ldots\ldots Q_0\}$

F 안 원소 중 1개라도 있으면 ok.

Ex2.8, again]



$\delta^*(q_0, 1) = \{Q_1\}$: reject

$\delta^*(q_0, \lambda) = \{Q_0\}$: accept.

자신 포함

accept strings: λ, 1010

reject strings: 1, 110

$L = \{(10)^n \mid n \geq 0\}$

⇒ 7↑E accept.

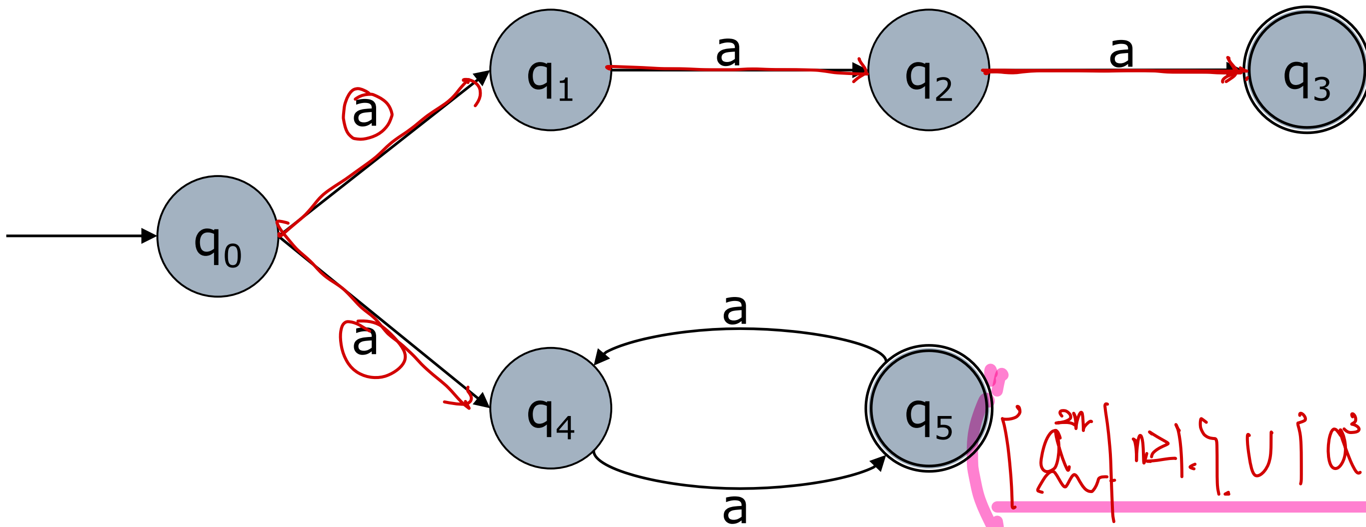
$\delta^*(q_0, 1010) = \{Q_0, Q_2\}$

$\delta^*(q_0, 110) = \emptyset$: reject

Why nondeterminism?

- An effective mechanism for describing some complicated languages concisely

→ "복잡한 언어를 정확하게 구사할 수 있다."



$\{a^{2n} \mid n \geq 1\} \cup \{a^3\}$

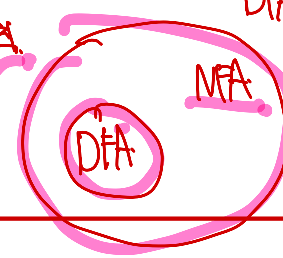
  - Language accepted by the nfa :

- Relation between and dfa & nfa?

→ 더 느슨한 규칙

$\delta(Q_1, a) = Q_0$

DFA ⊂ NFA



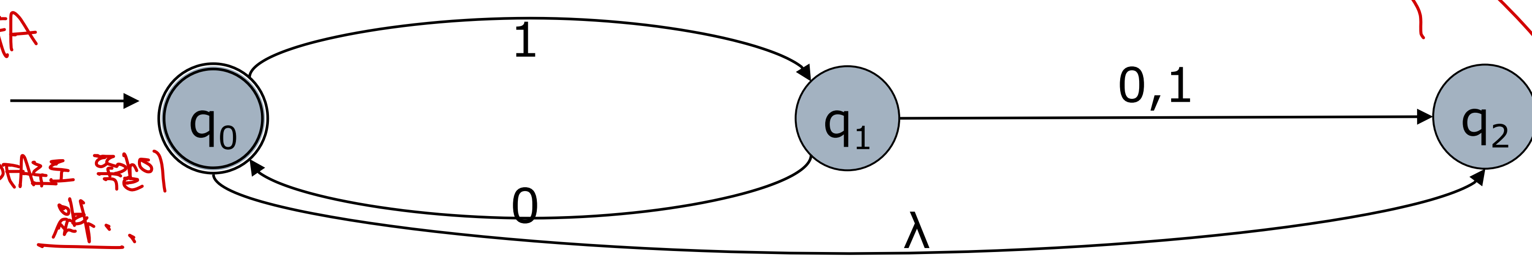
Equivalence of DFA & NFA (1/4)

- Two finite accepters M_1 and M_2 are equivalent if
  - $L(M_1) = L(M_2)$ 모양은 다르지만 accept 하는 language는 같다.
  - They both accept the same language

Ex2.11]

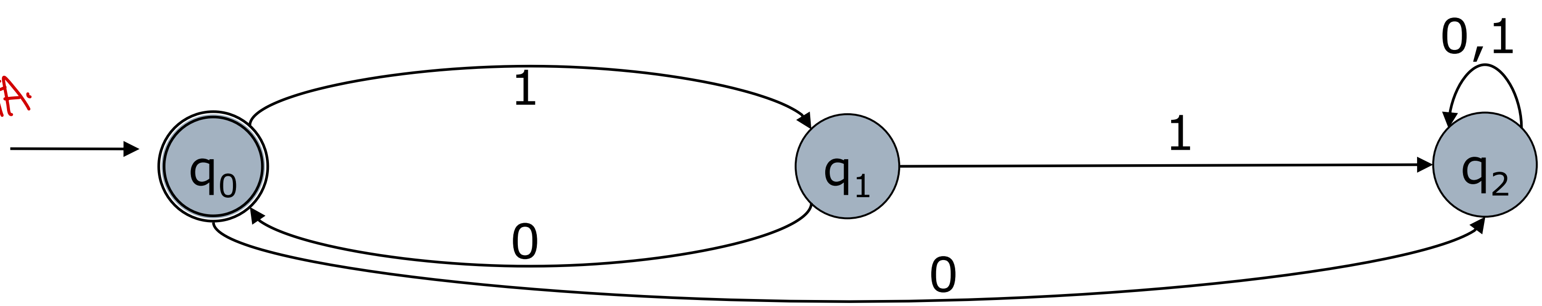
- nfa and dfa accepting $\{(10)^n : n \geq 0\}$

NFA



DFA로도 똑같이 함.

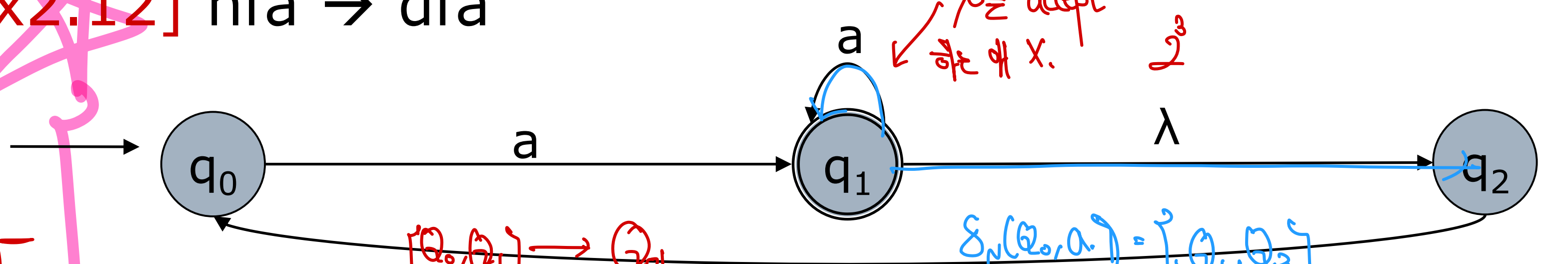
DFA.



Equivalence of DFA & NFA (2/4)

- The classes of DFA's and NFA's are equally powerful
  - possible to convert any nfa → an equivalent dfa
- How to convert
  - Correspond a set $\{q_i, q_j, \ldots, q_k\}$ in nfa → a state $q_{ij\ldots k}$ in dfa
  - For a set of $|Q|$ states, dfa can have states less than $2^{|Q|}$

Ex2.12] nfa → dfa



→ equivalent 한 dfa 어떻게 찾나!

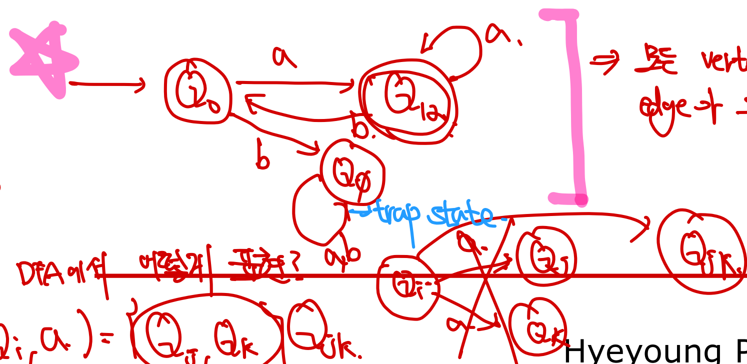
$\delta_N(Q_0, a) = \{Q_1, Q_2\}$.

$\delta_N(Q_0, b) = \emptyset$

$\delta_D(Q_{12}, a) = \{Q_1, Q_2\}$ NFA $\emptyset$ $= \delta_N(Q_1, b) \cup \delta_N(Q_2, b)$

Q_1에서 a 결과 Q_2에서 a 결과 union

$\delta_D(Q_{12}, b) = \{Q_0\}$.



⇒ 모든 vertex에 edge가 2개질 때까지

DFA DFA regular language

DFA에서 어떻게 표현?

$\delta_N(Q_i, a) = \{Q_j, Q_k\}$ Q_{jk}.

Equivalence of DFA & NFA (3/4)

[THEOREM 2.2]

L : language accepted by nfa $M_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$
then a dfa $M_D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ exists such that $L = L(M_D)$

Proof) Find a procedure nfa to dfa

$Q_{ij\ldots k} \xrightarrow{a}$

$\delta^*(Q_i,a) \cup \delta^*(Q_j,a)$
$\cup \delta^*(Q_k,a)$

1. Create a graph G_D with initial vertex $\{q_0\}$
2. Repeat until no more edges are missing
   - Take a vertex $[q_i,q_j,\ldots,q_k]$ of G_D with no outgoing edge for some *a*
   - Compute $\delta^*(q_i,a), \delta^*(q_j,a), \ldots, \delta^*(q_k,a)$
   - Form the union of all δ^* yielding the set "$\{q_l,q_m,\ldots,q_n\}$"
   - Create new vertex $[q_l,q_m,\ldots,q_n]$ if it does not "already exist"
   - Add an edge from $[q_i,q_j,\ldots,q_k]$ to $[q_l,q_m,\ldots,q_n]$ and label it with *a*
3. Identify final vertex
   - Every state of G_D with any $q_f \in F_N$
4. If M_N accepts "λ," the vertex $\{q_0\}$ is a final vertex

특별한 case

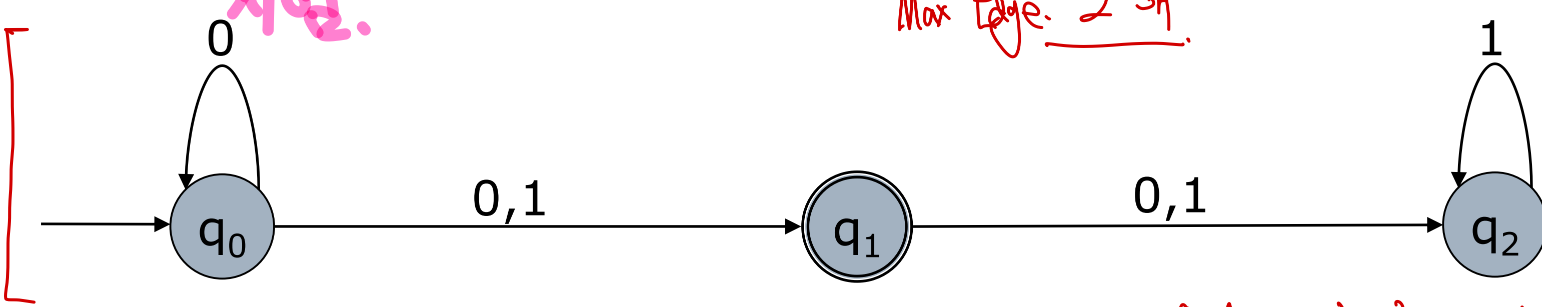
Equivalence of DFA & NFA (4/4)

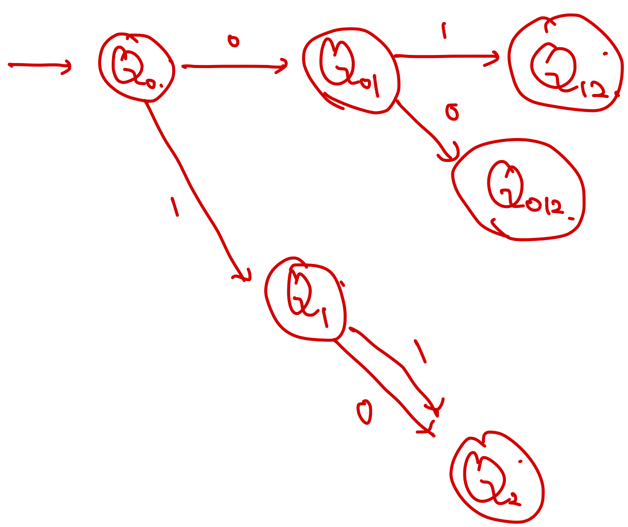
Ex 2.13] Convert nfa to an equivalent dfa

시험.

Max Edge. 2^3 개

NFA





$\delta_N(Q_0, 0) = \{Q_0, Q_1\}.$

$\delta_N(Q_0, 1) = \{Q_1\}.$

$\delta_D(Q_1, 0) = \{Q_2\}.$ $\delta_D(Q_1, 1) = \{Q_2\}$

$\delta_D(Q_{01}, 0) = \delta_N(Q_0, 0) \cup \delta_N(Q_1, 0) = \{Q_0, Q_1, Q_2\}.$

$\delta_D(Q_{01}, 1) = \delta_N(Q_0, 1) \cup \delta_N(Q_1, 1) = \{Q_1, Q_2\}.$

$\vdots$

더 이상 그릴 edge가 없을 때 까지