계산이론

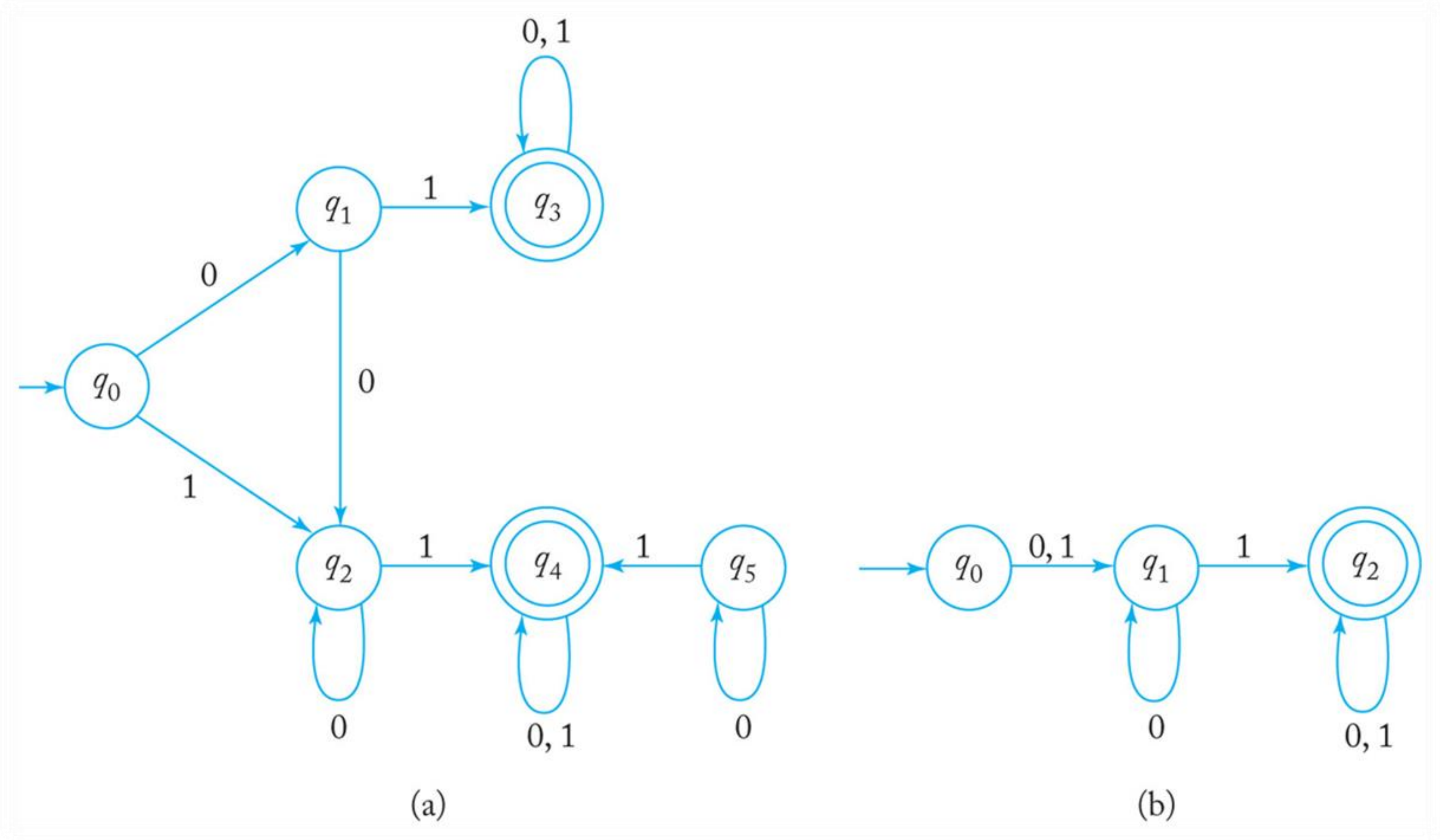
2022년 1학기

이은주

2장 유한 오토마타
유한 오토마타에서의 상태의 수 축소*
출력이 있는 유한 오토마타

유한 오토마타에서의 상태의 수 축소*

- 예제 2.14(70 page) : 기억 공간의 효율성



유한 오토마타에서의 상태의 수 축소*

- [정의 2.8] (75 page)
  - 임의의 dfa에서의 두 상태 p와 q에 대해, 이들이 모든 문자열 w∈Σ*에 대해 다음 조건을 만족하는 경우 : 구분불가능(indistinguishable)

    $$\delta^*(p, w) \in F \text{ implies } \delta^*(q, w) \in F,$$

    $$\delta^*(p, w) \notin F \text{ implies } \delta^*(q, w) \notin F.$$

  - 다음 조건을 만족하는 문자열 w ∈ Σ*가 존재하는 경우 : 상태 p와 q는 문자열 w에 의해 구분가능(distinguishable)

    $$\delta^*(p, w) \in F \text{ implies } \delta^*(q, w) \notin F,$$

    $$\delta_*(p, w) \notin F \text{ implies } \delta_*(q, w) \in F.$$

유한 오토마타에서의 상태의 수 축소*

- 구분불가능성은 동치관계 (equivalence relation)
  - 상태 p와 q가 구분불가능이고, 상태 q와 r 또한 구분불가능이면, 상태 p와 r도 구분불가능 : 세 상태 모두 구분불가능
- 임의의 dfa에서 상태의 수를 줄이는 한 가지 방법
  - 구분불가능 상태들을 찾아내어 이들을 병합하는 것

procedure : mark
(모든 구분가능한 쌍들을 마크하는 알고리즘)

1. 모든 도달 불가능 상태들 제거
   - dfa의 그래프에서 초기 상태로부터의 모든 단순 경로들 열거
     - 경로 상에 나타나지 않는 상태들은 도달 불가능한 것
2. 모든 상태 쌍 (p, q)에 대해 $p \in F$ and $q \notin F$ or $p \notin F$ and $q \in F$인 경우 :

   (p, q)는 구분가능 마크
3. 이전에 마크되지 않은 상태 쌍이 더 이상 마크되지 않을 때까지 다음을 계속
   - 모든 쌍 (p, q)와 모든 $a \in \Sigma$에 대해, $\delta(p, a) = p_a$와 $\delta(q, a) = q_a$를 계산
     - (p_a, q_a) 쌍이 구분가능으로 마크되어 있으면, (p, q)를 구분가능으로 마크

DFA의 최소화

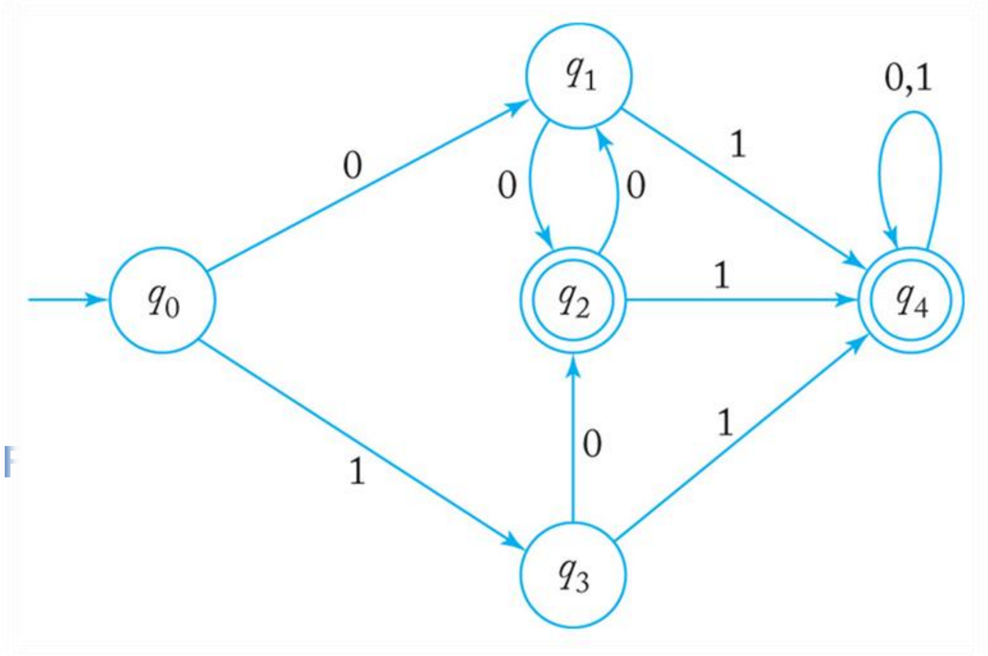
- 상태 q_i와 q_j가 길이 n인 문자열에 의해 구분가능하기 위한 필요충분 조건
  - 어떤 $a \in \Sigma$에 대해 다음과 같은 전이가 존재하고,

$$\delta(q_i, a) = q_k$$
$$\delta(q_j, a) = q_l$$

  - 이때, q_k와 q_l이 길이가 $n - 1$인 문자열들에 의해 구분가능해야 함

DFA의 최소화

- 예제 2.15 (77 page) : 프로시저 mark의 단계 2에서, 상태들의 집합을 승인 상태와 비승인 상태로 분할
  - 두 개의 동치 부류 $\{q_0, q_1, q_3\}$와 $\{q_2, q_4\}$ 를 얻음
  - 다음 단계에서 다음을 계산하면, $\delta(q_0, 0) = q_1$, $\delta(q_1, 0) = q_2$
  - q_0와 q_1는 구분가능함 : 서로 다른 동치 부류에 속하게 됨
    - $\{q_0, q_1, q_3\}$이 $\{q_0\}$ 과 $\{q_1, q_3\}$ 로 분리
  - $\delta(q_2, 0) = q_1$과 , $\delta(q_4, 0) = q_4$이므로, $\{q_2, q_4\}$는 $\{q_2\}$와 $\{ q_4\}$로 분리
  - 나머지 계산에서 더 이상 분리가 없음



2. 모든 상태 쌍 (p, q)에 대해 p ∈ F and q ∉ F or p ∉ F and q ∈ F

(p, q)는 구분가능 마크

DFA의 최소화

- [정의] DFA $M = (Q, \sum, \delta, q_0, F)$

q, q_1, q_2, q_f : 상태, x : 입력스트링

- $(q_1,x) \vdash^* (q,\lambda)$ and $(q_2,x) \vdash^* (q_f,\lambda)$, $q \notin F$, $q_f \in F$

  입력스트링 x가 상태 q_1과 q_2를 구분(distinguish)

- $q_1 \equiv^k q_2$,

  x ($|x| \leq k$)는 q_1과 q_2를 구분할 수 없음

- $q_1 \equiv q_2$,

  어떤 x도 q_1과 q_2를 구분할 수 없음.(q_1과 q_2는 동치(equivalent))

DFA의 최소화

- [정의] **<u>도달할 수 없는</u>**<u>(in-accessible)</u> 상태: 초기상태에서 도달 불가능
- [정의] **<u>최적화 된(최소 상태의) DFA M</u>** : 도달할 수 없는 상태가 없고, 어떤 두 개의 상태도 구분할 수 있음
- [알고리즘] **DFA의 최적화**

1. 도달할 수 없는 상태 제거
2. 동치 클래스(equivalent class) 구하기

   $\equiv^0, \equiv^1, \ldots$ 구하기 until $\equiv^{k+1} = \equiv^k$.

   Let $\equiv$ to be $\equiv^k$.

3. 최적화된 DFA $M' = (Q', \Sigma, \delta', q_0', F')$ 구성

(a) Q' : 동치 클래스들의 집합

[p] : Q' 의 원소(element)

(b) $\delta'([p], a) = [q]$ if $\delta(p, a) = q$.

(c) q_0' is $[q_0]$

(d) $F' = \{ [q] \mid q \in F \}$

DFA의 최소화

- [ex]
  1. **F, G : 도달할 수 없는 상태, 제거**

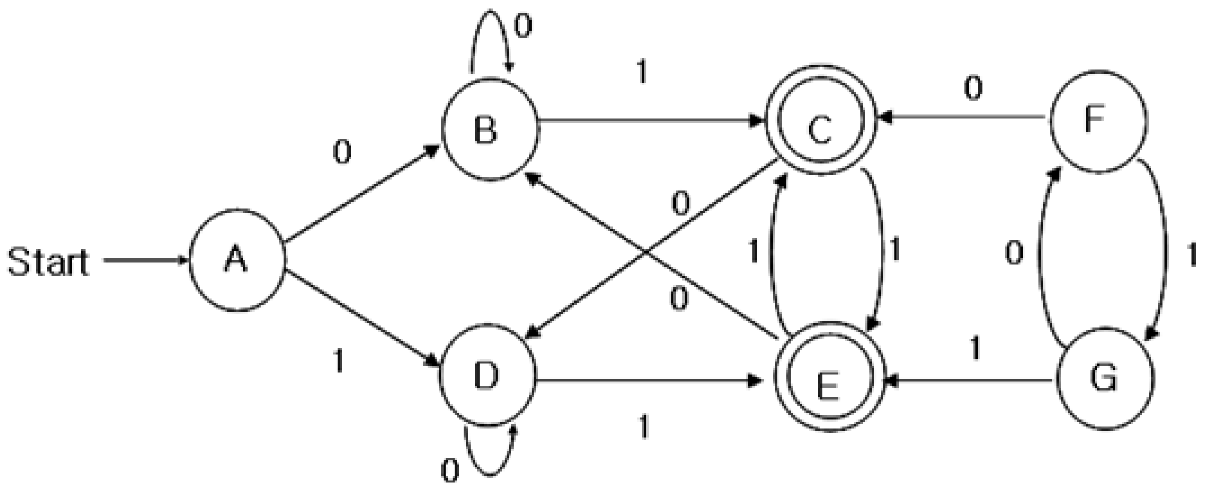
  2. **동치 클래스 구하기**

  For $\equiv^0$: (A, B, D) (C, E)

  For $\equiv^1$: (A) (B, D) (C, E)

  For $\equiv^2$: (A) (B, D) (C, E) : 동치 클래스

  ∴ Q' = {(A), (B, D), (C, E)}



DFA의 최소화

3. 최적화된 DFA M' = (Q', ∑, δ', q_0', F') 구성

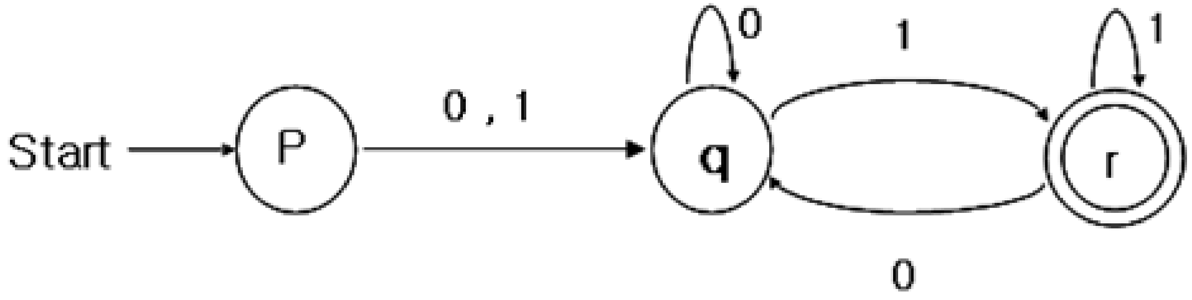
(A), (B, D), (C, E) 를 각각 p, q, r 바꾸면.

Q' = {p, q, r}

∑ = {0, 1}

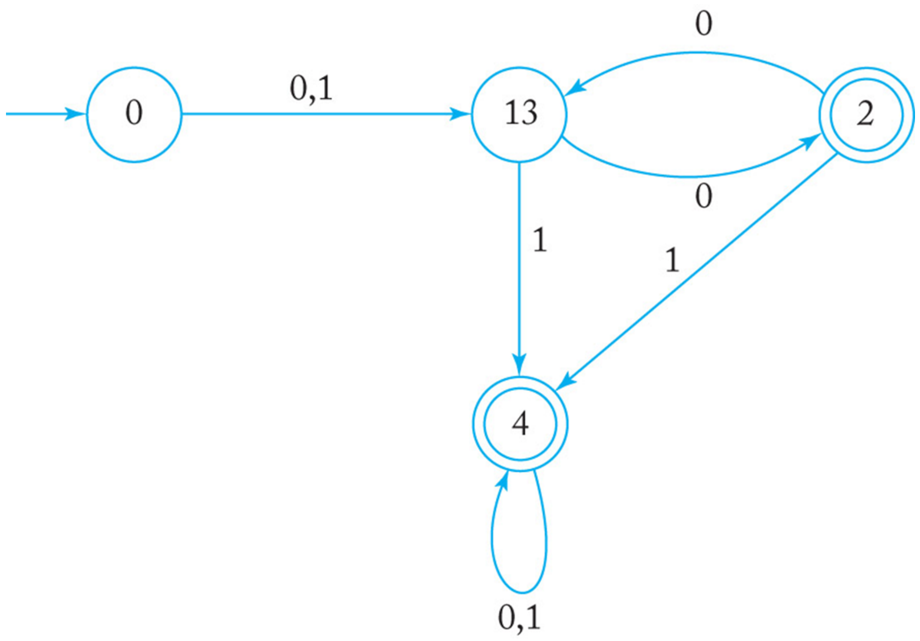
q_0' = (A) = p

F' = { (C,E) } = { r }



DFA의 최소화

- 예제 2.16(79 page)
  - 예제 2.15에 계속하여, 그림에서 보여주는 상태들이 생성됨
  - 예로서, 전이 $\delta(q_1, 0) = q_2$가 있기 때문에, 상태 13에서 상태 2로 가는 라벨이 0인 간선이 존재
  - 나머지 전이들도 쉽게 찾아지며, 최소 dfa를 얻게 됨



DFA의 최소화

- **[ex]**
  - 서브스트링 bbb를 포함하는 a, b의 string을 인식하는 최소의 상태(4개)를 갖는 DFA를 구하라.

nfa	a	b
초기 1	1	{1,2}
2	-	3
3	-	4
최종 4	4	4

  - 초기화 : λ-closure(1) = **{1} = A**
  - marking A : a-successor(A) = {1} = A

    b-successor(A) = **{1,2} = B**

DFA의 최소화

- marking B : a-successor(B) = {1} = A
  b-successor(B) = **{1,2,3} = C**
- marking C : a-successor(C) = {1} = A
  b-successor(C) = **{1,2,3,4} = D**
- marking D : a-successor(D) = **{1,4} = E**
  b-successor(D) = {1,2,3,4} = D
- marking E : a-successor(E) = {1,4} = E
  b-successor(E) = **{1,2,4} = F**
- marking F : a-successor(F) = {1,4} = E
  b-successor(F) = {1,2,3,4} = D

nfa	a	b
초기 1	1	{1,2}
2	-	3
3	-	4
최종 4	4	4

DFA의 최소화

- **[ex]** 서브스트링 aabb를 포함하는 a, b의 string을 인식하는 최소의 상태(5개)를 갖는 DFA를 구하라.

nfa	a	b
초기 1	{1,2}	1
2	3	-
3	-	4
4	-	5
최종 5	5	5

- 초기화 : λ-closure(1) = **{1} = A**
- marking A : a-successor(A) = **{1,2} = B**

  b-successor(A) = {1} = A

DFA의 최소화

- marking B : a-successor(B) = **{1,2,3} = C**
  b-successor(B) = {1} = A
- marking C : a-successor(C) = {1,2,3} = C
  b-successor(C) = **{1,4} = D**
- marking D : a-successor(D) = {1,2} = B
  b-successor(D) = **{1,5} = E**
- marking E : a-successor(E) = **{1,2,5} = F**
  b-successor(E) = {1,5} = E
- marking F : a-successor(F) = **{1,2,3,5} = G**
  b-successor(F) = {1,5} = E
- marking G : a-successor(G) = {1,2,3,5} = G
  b-successor(G) = **{1,4,5} = H**
- marking H : a-successor(H) = {1,2,5} = F
  b-successor(H) = {1,5} = E

nfa	a	b
초기 1	{1,2}	1
2	3	-
3	-	4
4	-	5
최종 5	5	5

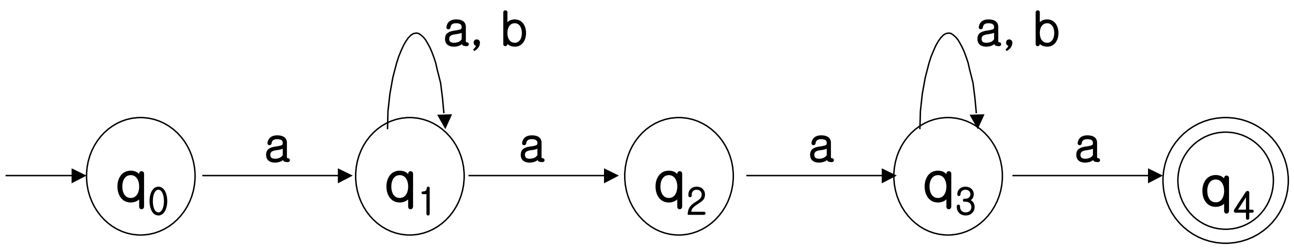
DFA의 최소화

dfa	a	b
초기 A	B	A
B	C	A
C	C	D
D	B	E
최종 E	~~F~~ E	E
최종 ~~F~~	~~G~~	~~E~~
최종 ~~G~~	~~G~~	~~H~~
최종 ~~H~~	~~F~~	~~E~~

최소 상태의 dfa	a	b
초기 A	B	A
B	C	A
C	C	D
D	B	E
최종 E	E	E

DFA의 최소화

- **[ex]** L = { $\boldsymbol{a w_1 a a w_2 a}$ | $w_1, w_2 \in \{a,b\}^*$ }를 인식하는 최소상태의 dfa M을 만들어라.
  - nfa :



  - 초기화 : λ-closure(q_0) = **{q_0} = A**
  - marking A : a-successor(A) = **{q_1} = B**
  - marking B : a-successor(B) = **{q_1, q_2} = C**

    b-successor(B) = {q_1} = B
  - marking C : a-successor(C) = **{q_1, q_2, q_3} = D**

    b-successor(C) = {q_1} = B

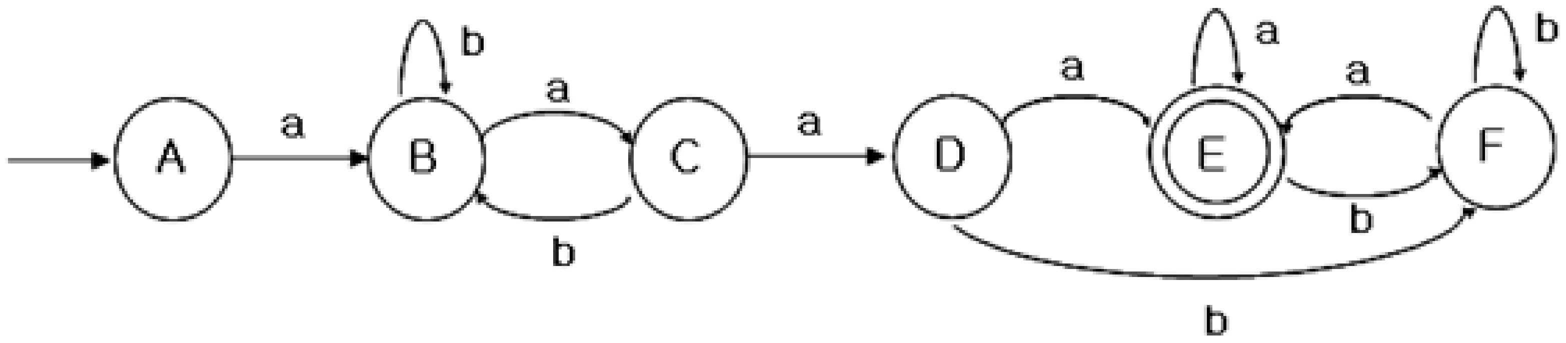
DFA의 최소화

- marking D : a-successor(D) = **$\{q_1, q_2, q_3, q_4\}$ = E**

  b-successor(D) = **$\{q_1, q_3\}$ = F**
- marking E : a-successor(E) = $\{q_1, q_2, q_3, q_4\}$ = E

  b-successor(E) = $\{q_1, q_3\}$ = F
- marking F : a-successor(F) = $\{q_1, q_2, q_3, q_4\}$ = E

  b-successor(F) = $\{q_1, q_3\}$ = F



출력이 있는 유한오토마타

- [정의] **밀리기계**(Mealy Machine, Transition-assigned FSM)

  $M = (Q, \Sigma, \Delta, \delta, \lambda, q_o)$

  Q : 상태의 집합

  ∑ : 입력 기호의 집합

  Δ : 출력 기호의 집합

  δ : 전이함수 $Q \times \Sigma \rightarrow Q$

  λ : 출력함수 $Q \times \Sigma \rightarrow \Delta$

  q_o : 초기 상태

  - 상태가 전이될 때마다 출력함수에 의해 기호가 출력됨.

    $\rightarrow (p) \xrightarrow{a/b} (q) \rightarrow$

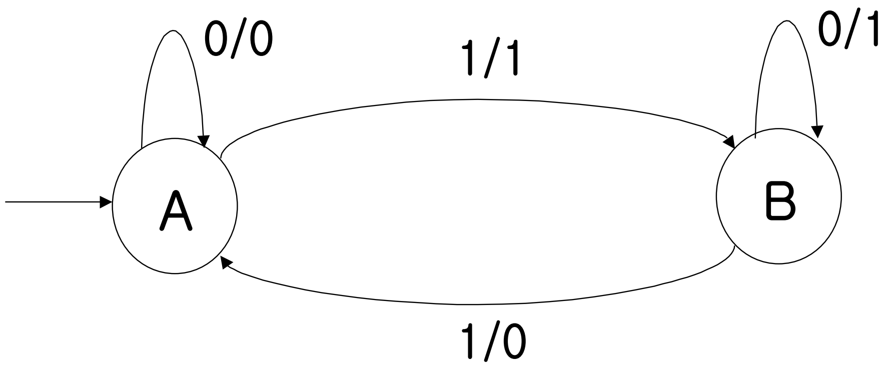
  - 상태 전이 때 : 입력 a, 출력 b

    $\delta(p, a) = q$

    $\lambda(p, a) = b$

출력이 있는 유한오토마타

- **[ex]** Parity Checker



밀리 M	0	1
초기 A	A, 0	B, 1
B	B, 1	A, 0

- 밀리기계 M = ({A,B}, {0,1}, {0,1}, δ, λ, A)

$\delta : \delta(A, 0) = A$ $\quad\quad$ $\lambda : \lambda(A, 0) = 0$

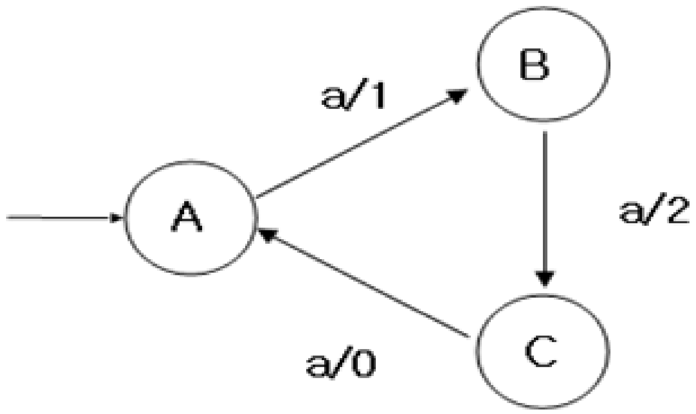
$\delta(B, 0) = B$ $\quad\quad$ $\lambda(B, 0) = 1$

$\delta(A, 1) = B$ $\quad\quad$ $\lambda(A, 1) = 1$

$\delta(B, 1) = A$ $\quad\quad$ $\lambda(B, 1) = 0$

출력이 있는 유한오토마타

- **[ex]** Mod-3 Counter



- 입력스트링 aaaa ,

  → Ⓐ $\rightarrow^{a/1}$ Ⓑ $\rightarrow^{a/2}$ Ⓒ $\rightarrow^{a/0}$ Ⓐ $\rightarrow^{a/1}$ Ⓑ

출력이 있는 유한오토마타

- [정의] **무어기계**(Moore Machine, State-assigned FSM)

  $M = (Q, \sum, \Delta, \delta, \lambda, q_o)$

  λ : 출력함수 $Q \rightarrow \Delta$

  나머지는 Mealy기계와 동일.

  - 출력 기호가 각 상태에 의해 결정됨.

    $\rightarrow q \rightarrow^{a} q' \rightarrow$

    output b b'

    $\delta(q, a) = q'$

    $\lambda(q) = b$

    $\lambda(q') = b'$

  - 입력스트링의 길이 = n
  - 출력스트링의 길이 = n+1

    (초기상태에서 입력에 관계없이 출력)

출력이 있는 유한오토마타

- **[ex]** 2진 스트링으로 표현된 10진수의 mod-3 값을 출력하는

  무어기계 $M = (\{q_0, q_1, q_2\}, \{0,1\}, \{0,1,2\}, \delta, \lambda, q_0)$

  $\delta : \delta(q_0, 0) = q_0$

  $\delta(q_1, 0) = q_2$

  $\delta(q_2, 0) = q_1$

  $\delta(q_0, 1) = q_1$

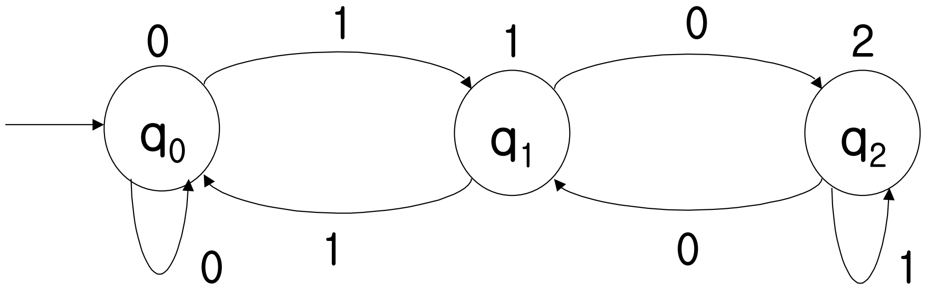
  $\delta(q_1, 1) = q_0$

  $\delta(q_2, 1) = q_2$

  $\lambda : \lambda(q_0) = 0$

  $\lambda(q_1) = 1$

  $\lambda(q_2) = 2$

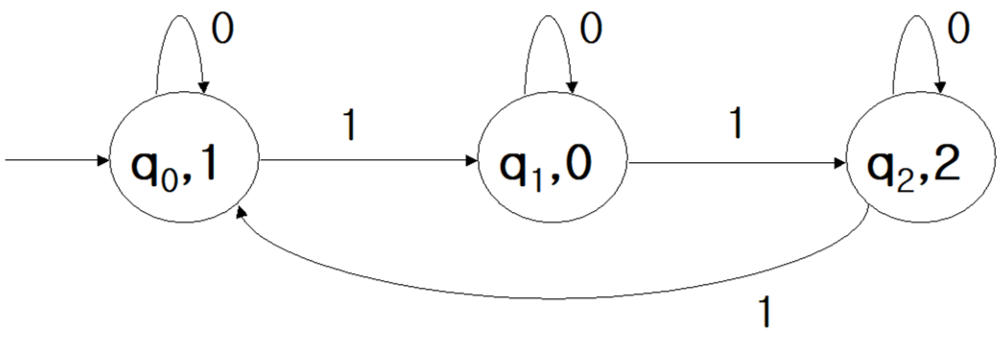


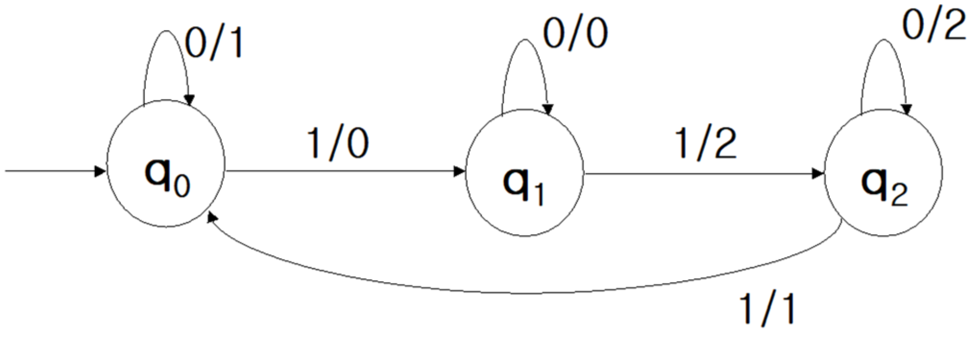
무어 M	0	1	output
초기 q_0	q_0	q_1	0
q_1	q_2	q_0	1
q_2	q_1	q_2	2

입력스트링 1010, q0 →1 q1 →0 q2 →1 q2 →0 q1
output 0 1 2 2 1

출력이 있는 유한오토마타

- [정리] 무어기계와 유사한 밀리기계가 항상 존재
- **[ex]** 무어기계: 밀리기계:





입력스트링 100 ,

무어기계 : $q_0 \rightarrow^1 q_1 \rightarrow^0 q_1 \rightarrow^0 q_1$

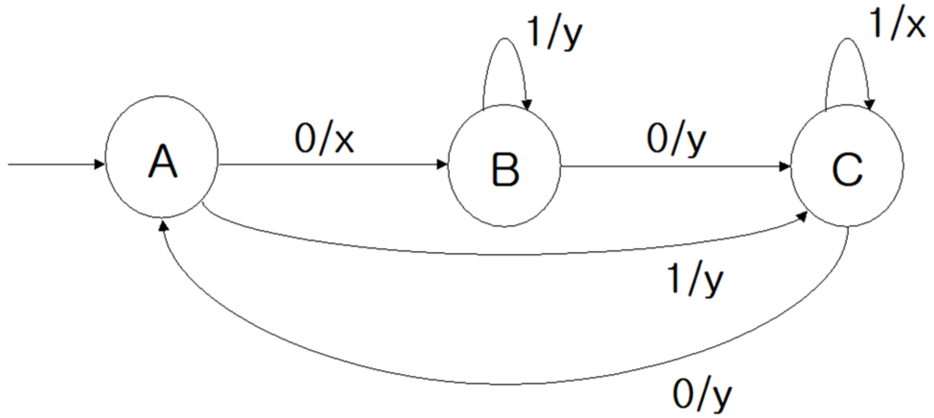
output 1 0 0 0

밀리기계 : $q_0 \rightarrow^{1/0} q_1 \rightarrow^{0/0} q_1 \rightarrow^{0/0} q_1$, output= 000

출력이 있는 유한오토마타

- [정리] 밀리기계와 유사한 무어기계가 항상 존재

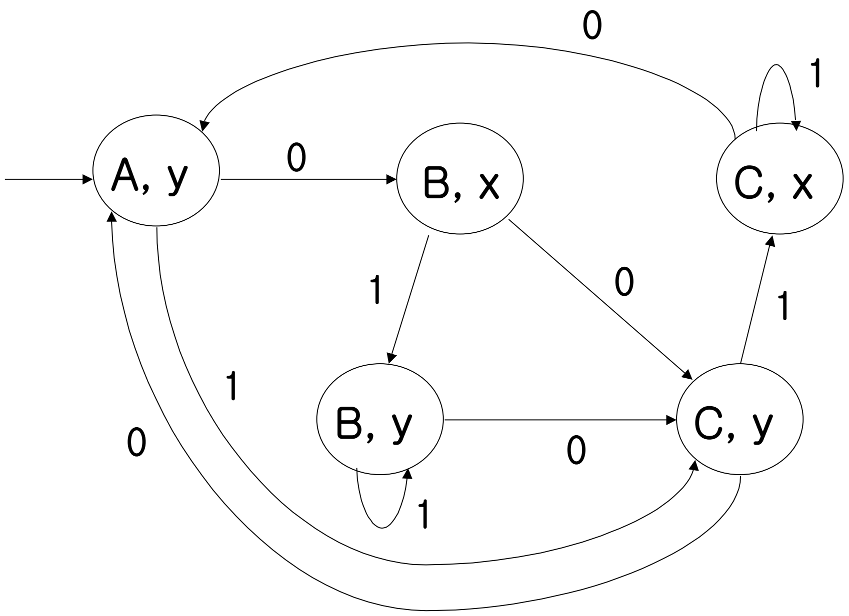
- **[ex]** 밀리기계 M을 동치인 무어기계 M'로 바꿔라.
  (Table 이용)
  밀리기계:



밀리 M	0	1
초기 A	B, x	C, y
B	C, y	B, y
C	A, y	C, x

출력이 있는 유한오토마타

무어기계:



밀리 M	0	1
초기 A	B, x	C, y
B	C, y	B, y
C	A, y	C, x

무어 M'	0	1	출력
초기 Ay	Bx	Cy	y
Bx	Cy	By	x
By	Cy	By	y
Cx	Ay	Cx	x
Cy	Ay	Cx	y

출력이 있는 유한오토마타

- **[ex]** 밀리기계 M을 동치인 무어기계 M'로 바꿔라.

밀리 M	a	b
초기 P	Q, x	R, y
Q	R, x	Q, z
R	P, z	Q, y

무어 M'	a	b	출력
초기 Pz	Qx	Ry	z
Qx	Rx	Qz	x
Qy	Rx	Qz	y
Qz	Rx	Qz	z
Rx	Pz	Qy	x
Ry	Pz	Qy	y

출력이 있는 유한오토마타

- **[ex]** 밀리기계 M을 동치인 무어기계 M′ 으로 바꿔라.

밀리 M	0	1
초기 q_0	q_0, 1	q_1, 1
q_1	q_1, 0	q_0, 0

무어 M'	0	1	출력
초기 A (q_0, 0)	**B**	**D**	**0**
B (q_0, 1)	**B**	**D**	**1**
C (q_1, 0)	**C**	**A**	**0**
D (q_1, 1)	**C**	**A**	**1**