11. 오토마타

충북대학교

이 재 성

LaKE LAB

Language Knowledge Engineering

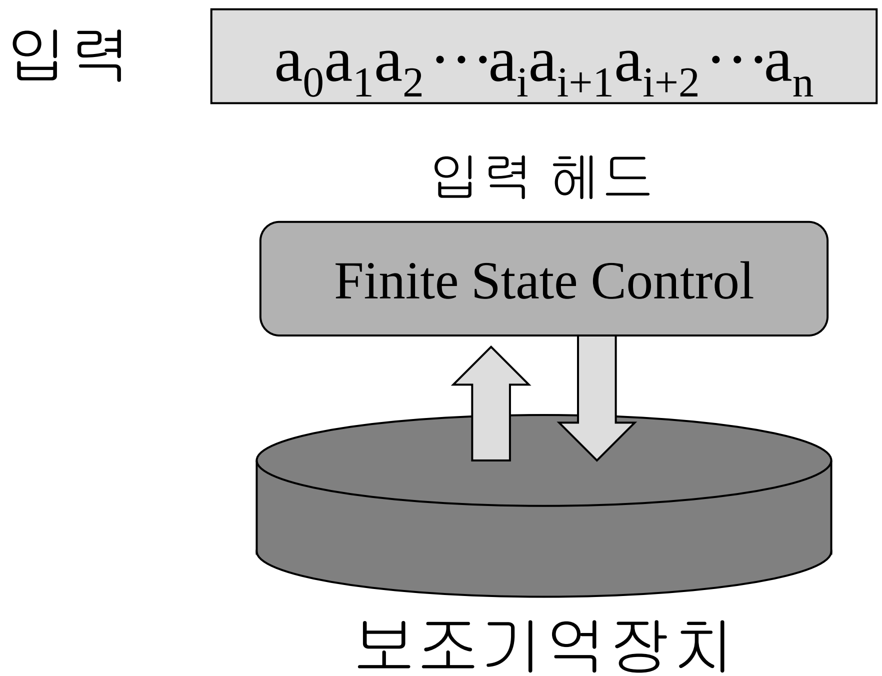
학습내용

- 유한 오토마타 정의
- 결정적 유한 오토마타(DFA)
- 비결정적 유한 오토마타(NFA)
- 정규표현을 NFA로 변환하는 방법
- NFA를 DFA로 변환하는 방법

인식기 (Recognizer)

인식기

- 입력으로 스트링을 받아 스트링이 그 언어의 문장이면 "yes", 아니면 "no"를 출력하는 프로그램



- 종류
  - 튜링 머신
  - A에 선형 종속
  - 푸쉬다운 오토마타
  - 유한 오토마타

유한 오토마타 (Finite Automata)

정의 : FA(Finite Automata)

알파벳 Σ에 대한 유한 오토마타 $M = (Q, \Sigma, \delta, q_0, F)$

여기서, Q : 상태(state)들의 유한 집합;

Σ : 입력 알파벳의 유한 집합;

δ : 사상 함수;

$q_0 \in Q$: 시작 상태

$F \subseteq Q$: 종결 상태의 집합

사상함수 $\delta : Q \times \Sigma \rightarrow 2^Q$

i,e. $\delta(q,a) = \{p_1, p_2, \ldots, p_n\}$

$G = (V_N, V_T, P, S)$

re : f, e, a, +, •, *

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

결정적 유한 오토마타 (DFA)

결정적

- $\delta(q,a)$가 한 상태만을 갖는 경우
- $\delta(q,a) = \{p\}$ 대신에 "$\delta(q,a) = p$ " 로 표기
- $\delta(q,a)$가 항상 한 개의 다음 상태를 가질 때, M이 완전히 명시

δ 함수의 확장

- $Q \times \sum \Rightarrow Q \times \sum^*$
- $\delta(q, \varepsilon) = q$

  $\delta(q,xa) = \delta(\delta(q,x),a)$, 여기서 $x \in \sum^*$ 이고 $a \in \sum$.

한 문장 x 의 인식

- 만약 $\delta(q_0, x) = p$인 경우, p가 종결 상태에 속할 때($p \in F$).
- M에 의해 인식된 언어
  - $L(M) = \{ x \mid \delta(q_0,x) \in F \}$

DFA 예

ex) $M = (\{p, q, r\}, \{0, 1\}, \delta, p, \{r\})$

$\delta : \delta(p,0) = q \qquad \delta(p,1) = p$

$\delta(q,0) = r \qquad \delta(q,1) = p$

$\delta(r,0) = r \qquad \delta(r,1) = r$

– $1001 \in L(M)$?

$\delta(p,1001) = \delta(p,001) = \delta(q,01) = \delta(r,1) = r \in F$.

$\therefore\ 1001 \in L(M)$.

– $1010 \in L(M)$?

$\delta(p,1010) = \delta(p,010) = \delta(q,10) = \delta(p,0) = q \notin F$.

$\therefore\ 0110 \notin L(M)$.

- 전이테이블: δ 를 matrix 형태로 표시. ex)

δ	Input symbols	
	0	1
p	q	p
q	r	p
r	r	r

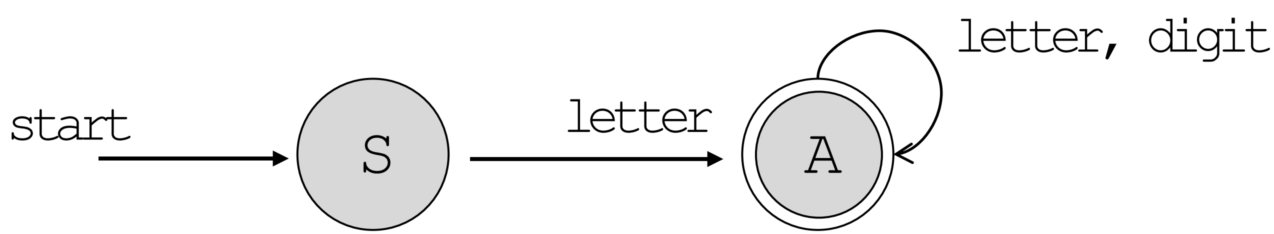
LaKE LAB

상태 전이도

■ 정의

- 노드: 유한 오토마타의 각 상태를 표시(p, q)
- 전이 상태: 상태 q에서 상태p로 가는 변화를 타나냄
- 레이블: 전이때의 입력 문자 (a)
- $\delta(q,a) = p$
- 종결 상태들은 이중 원으로 표시하고 시작 상태는 start 지시 선으로 표시

인식기 :



DFA 알고리즘

Algorithm : $w \in L(M)$ 결정.

```
assume M = (Q, ∑, δ, q0, F);
begin
  currentstate := q0;   (* start state *)
  get(nextsymbol);
  while not eof do
   begin  currentstate := δ(currentstate, nextsymbol);
              get(nextsymbol)
   end;
  if currentstate in F then write('Valid String')
    else write('Invalid String');
end.
```

비결정적 유한 오토마타 (NFA)

비결정적

- 상태 q에서 입력 심벌 a에 대해 갈 수 있는 상태가 여러개
- $\delta(q,a) = \{p1, p2, \ldots, pn\}$

예

- $M = (\{q0,q1,q2,q3,qf\}, \{0,1\}, \delta, q0, \{qf\})$
  - if $\delta(q,a) = f$, then $\delta(q,a)$ is undefined.

δ	**0**	**1**
$\mathbf{q_0}$	$\{q_1, q_2\}$	$\{q_1, q_3\}$
$\mathbf{q_1}$	$\{q_1, q_2\}$	$\{q_1, q_3\}$
$\mathbf{q_2}$	$\{q_f\}$	ϕ
$\mathbf{q_3}$	ϕ	$\{q_f\}$
$\mathbf{q_4}$	$\{q_f\}$	$\{q_f\}$

■ 정의 : $\delta(q_0, x)$의 상태 p 중 F의 상태를 포함하는 경우, 문장 x는 M에 의해 인식된다.

ex) $1011 \in L(M)$?

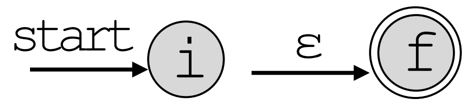
$$\delta(\{q_0\}, 1011) = \delta(\{q_1,q_3\}, 011) = \delta(\{q_1,q_2\},11)$$
$$= \delta(\{q_1,q_3\},1) = \{q_1,q_3,q_f\}$$
$$\therefore 1011 \in L(M) \ \ (\because \{q_1,q_3,q_f\} \cap \{q_f\} \neq \Phi)$$

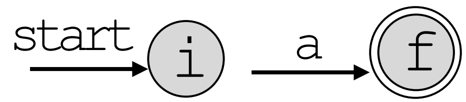
ex) $0100 \in L(M)$?

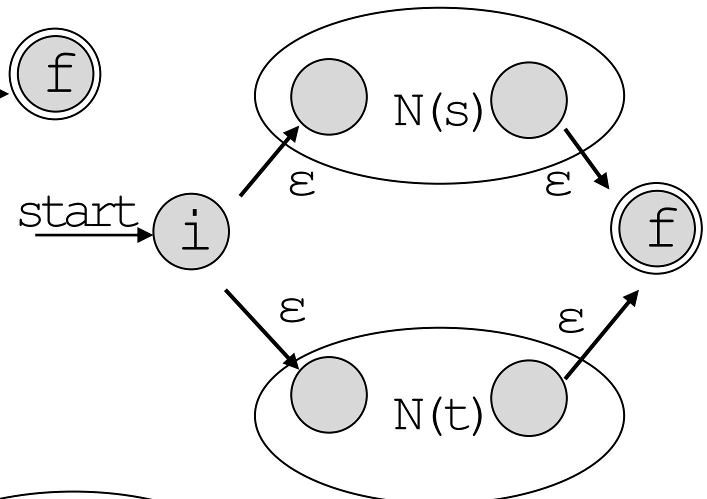
δ	0	1
q_0	$\{q_1, q_2\}$	$\{q_1, q_3\}$
q_1	$\{q_1, q_2\}$	$\{q_1, q_3\}$
q_2	$\{q_f\}$	ϕ
q_3	ϕ	$\{q_f\}$
q_4	$\{q_f\}$	$\{q_f\}$

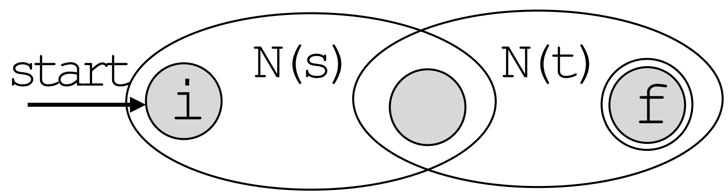
정규표현의 NFA로의 변환

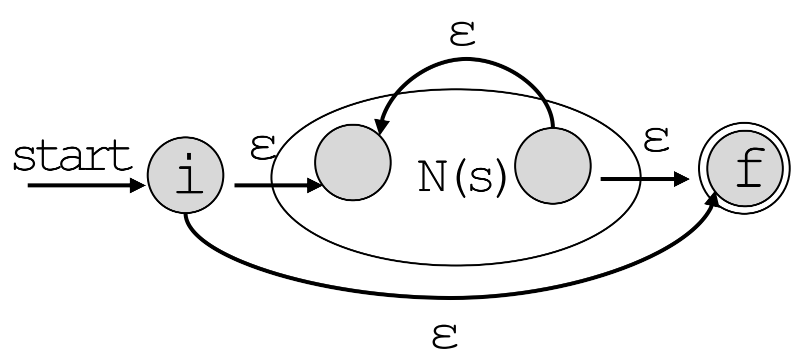
- 구성요소 분해
  - 정규표현을 구성요소 부분식으로 분리한 후 다시 규칙에 따라 조합
  - ε
  - a
  - s | t => N(s|t)
  - st => N(st)
  - s* => N(s*)
  - (s) => N(s)







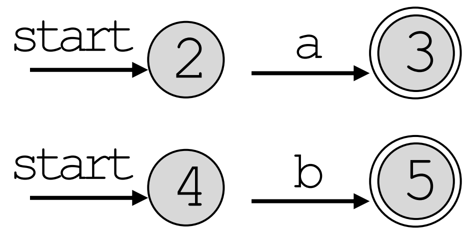


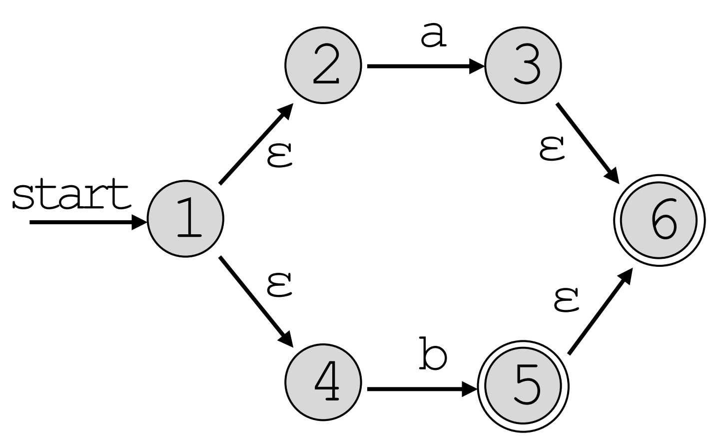


정규표현의 NFA로의 변환 예(1)

■ 정규표현 r=(a|b)*abb 의 변환순서

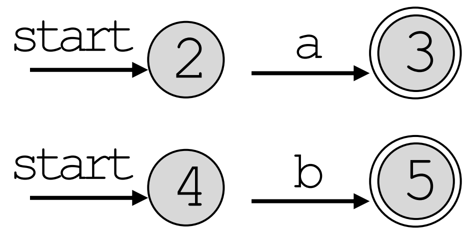
1. r1 =a
2. r2 = b
3. r3= r1|r2
4. r4=(r3)
5. r5=(r3)*
6. r6=a
7. r7=r5 r6
8. :

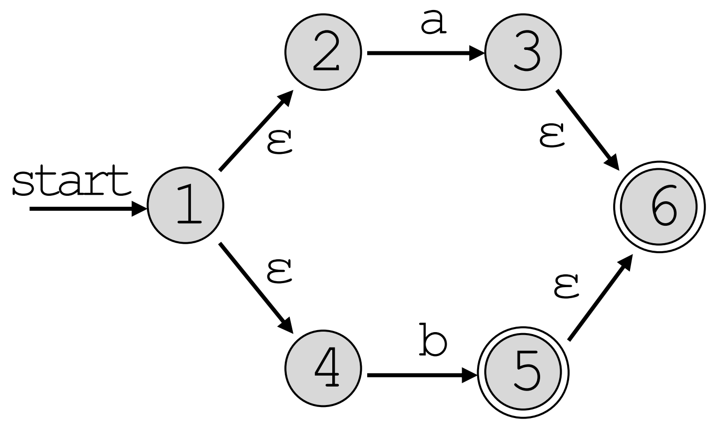




■ 정규표현 r=(a|b)*abb 의 변환순서

1. r1 =a
2. r2 = b
3. r3= r1|r2
4. r4=(r3)
5. r5=(r3)*
6. r6=a
7. r7=r5 r6
8. :

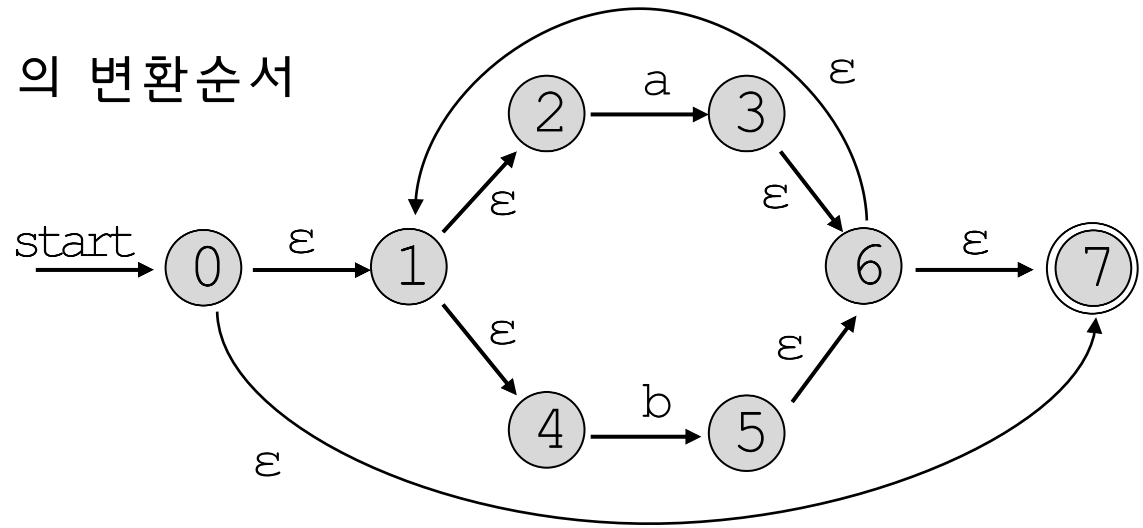


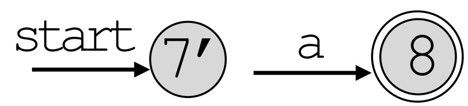


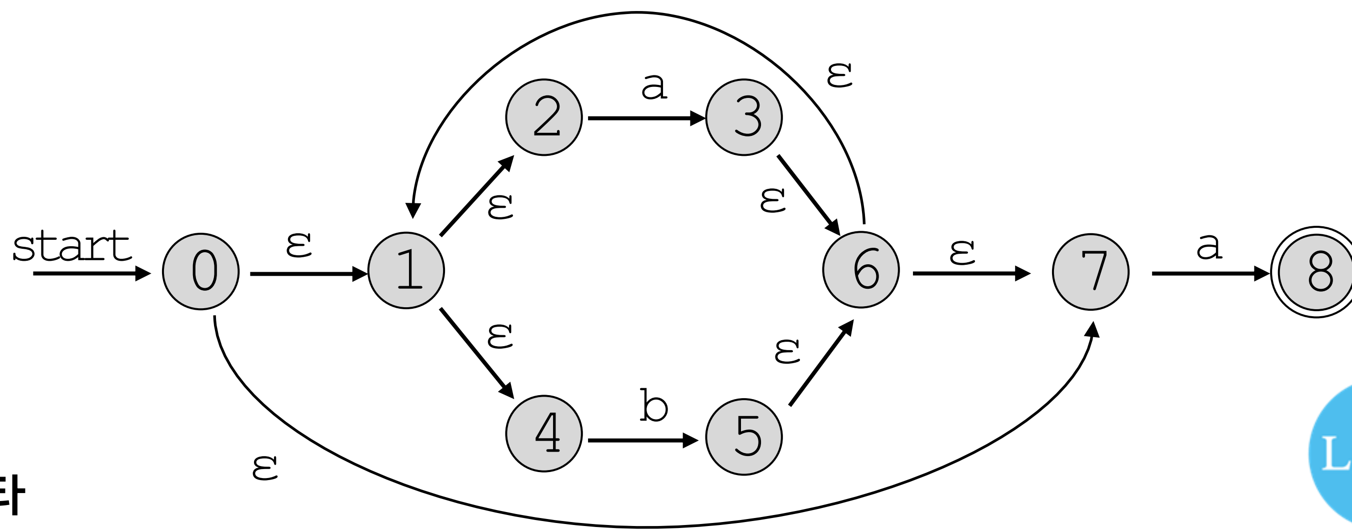
정규표현의 NFA로의 변환 예(2)

- 정규표현 r=(a|b)*abb 의 변환순서

1. r1 =a
2. r2 = b
3. r3= r1|r2
4. r4=(r3)
5. r5=(r3)*
6. r6=a
7. r7=r5 r6
8. :







NFA를 DFA로 전환

필요성

- NFA에서 ε전이에 대해 컴퓨터 처리가 어려움

부분집합 구성 알고리즘 용어

- ε-closure(s):
  - NFA상태 s에서 ε전이만으로 도달 가능한 상태집합
- ε-closure(T):
  - T에 포함된 어떤 상태 s에서 ε전이만으로 도달가능한 상태 집합
- move(T,a):
  - T에 포함되는 어떤 상태s에서 입력기호 a에 의해 전이되는 상태집합
- Dtran:
  - D에 대한 전이 테이블
- Dstates:
  - D의 상태들의 집합

부분집합 구성 알고리즘

- 입력: NFA N

- 출력: 동일한 언어를 수용하는 하나의 DFA D

- 방법

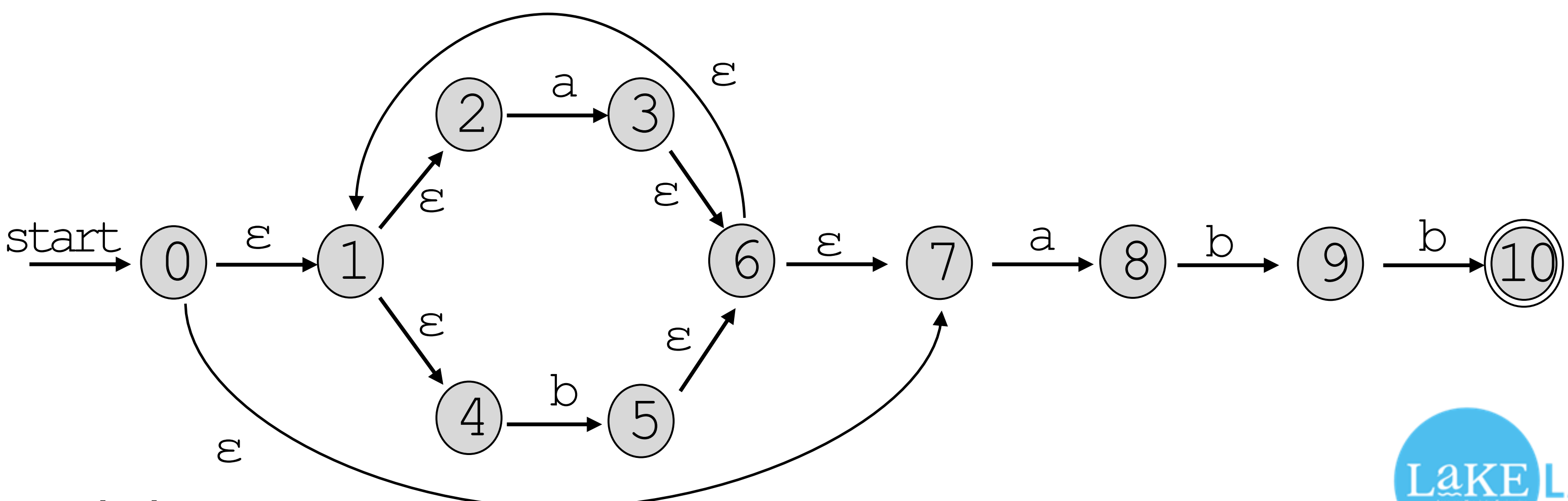
상태 ε-closure(s_0)만을 Dstates에 넣는다.
while Dstates 안에 마크가 안된 상태 T가 있다 **do begin**
    mark T;
    for 각 입력기호 a **do begin**
        U := ε-closure(move(T,a));
        if U가 Dstates안에 없다 **then**
            U를 마크하지 않고 Dstates에 추가
        Dtran[T, a] := U
    end
end

ε-closure의 계산

```
T안의 상태를 모두 스택에 푸시
ε-closure(T)를 T로 초기 설정
while 스택이 비어있지 않음 do begin
    스택에서 최상단 요소 t를 제거
    for t에서 ε레이블로 연결 가능한 상태 u do
        if u가 ε-closure(T)안에 없음 then do begin
            u를 ε-closure(T)에 추가
            u를 스택에 푸시
        end
    end
```

부분집합 구성 알고리즘 예(1)

- 시작상태 ε-closure(0) A={0,1,2,4,7}
  - 입력 알파벳 {a, b}
- ε-closure(move(A, a)) = ε-closure(move{0,1,2,4,7}, a))
  = ε-closure({3,8}) = {1,2,3,4,6,7,8} -> B
  - Dtran[A,a] = B
- ε-closure(move(A, b)) = ε-closure(move{0,1,2,4,7}, b))
  = ε-closure({5}) = {1,2,4,5,6,7} -> C
  - Dtran[A,b] = C

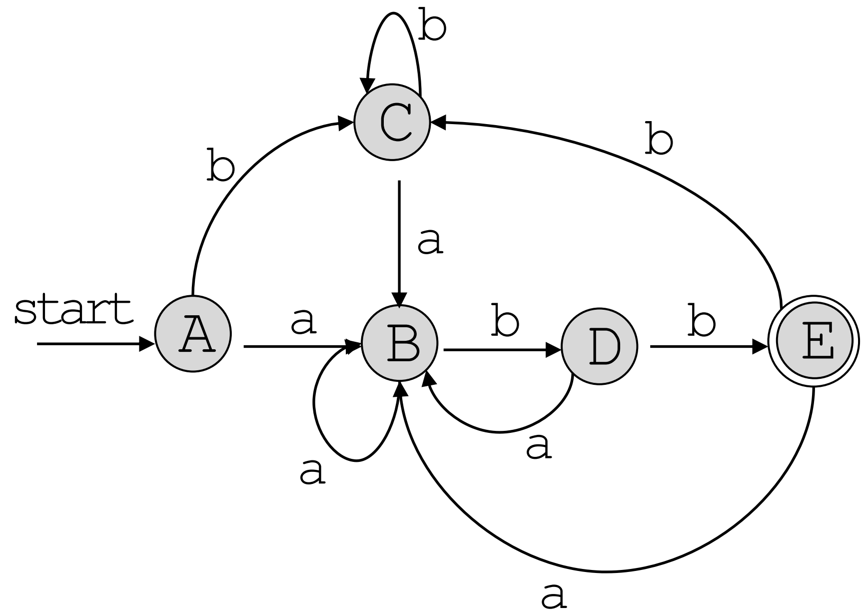


부분집합 구성 알고리즘 예(2)

- 구성된 부분집합
  - A={0,1,2,4,7} B={1,2,3,4,6,7,8} C={1,2,4,5,6,7}
  - D={1,2,4,5,6,7,9} E={1,2,4,5,6,7,10}

- DFA를 위한 전이 테이블

상태	입력 기호	
	a	b
A	B	C
B	B	D
C	B	C
D	B	E
E	B	C



FA에서 상태수를 줄이는 방법

<step 1> 모든 도달 불가능한 상태는 삭제;
<step 2> 동치 관계를 만듬;
<step 3> fa $M' = (Q', \sum, \delta', q_0', F')$를 만듬;

(a) Q' : $\equiv$부분 안의 동치류의 집합
[p]는 $\equiv$부분 안의 p상태의 동치류가 된다.

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

(c) $q_0' = [q_0]$.

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

정의

만약 (1) Q에 속하지 않은 상태는 도달 불가능하고
(2) Q의 구별되지 않은 상태들은 구별 불가능하면,
M은 축약되었다고 한다

ex) 유한 오토마톤 M = ({A,B,C,D,E,F}, {0,1}, δ, A, {E,F})에 의해 상세화된 언어를 위해 최소 상태 유한 오토마톤을 찾는다.

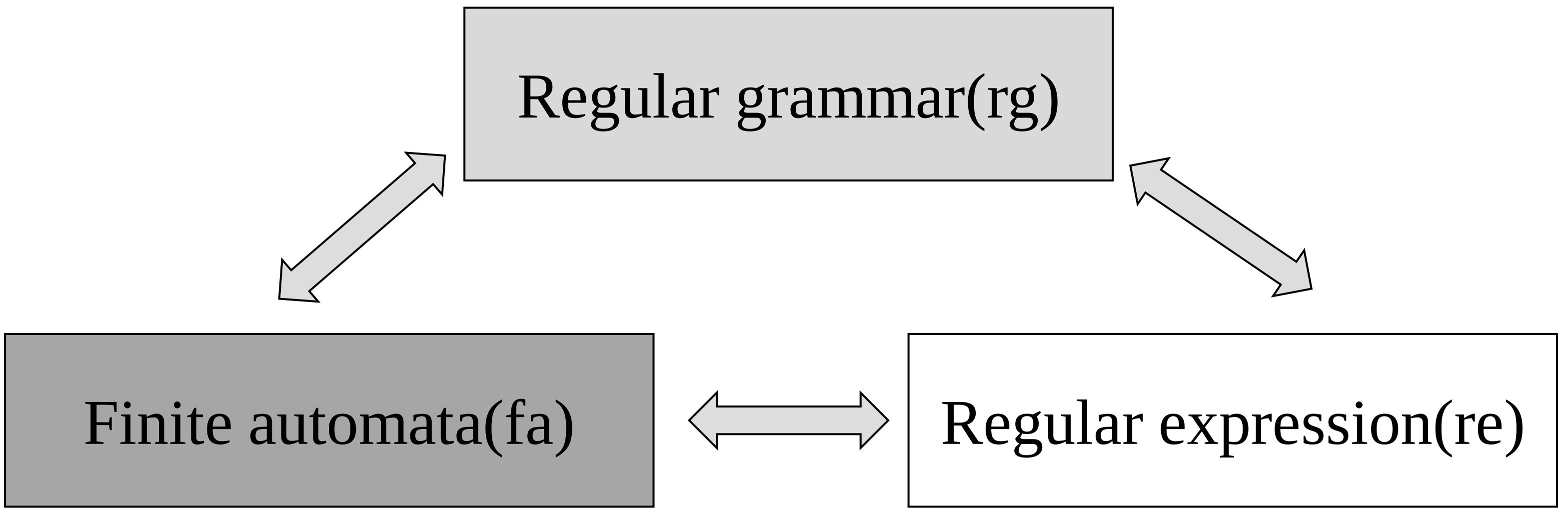
– δ가 주어졌을 때

δ	0	1
A	B	C
B	E	F
C	A	A
D	F	E
E	D	F
F	D	E

– ≡ : {A, B, C, D}, {E, F}   ≡ : {A, C}, {B, D}, {E, F}

δ	0	1
[A,C]=p	q	p
[B,D]=q	r	r
[E,F]=q	q	r

정규 언어의 속성



※ re ===> fa : scanner generator

참고 문헌

- [1] Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, "Compilers – Principles, Techniques, and Tools," Bell Telephone Laboratories, Incorporated, 1986.
- [2] 오세만, "컴파일러 입문", 정익사, 2004.