2003 Fall

CS322 오토마타 중간고사 (155점)

2003년 10월 21일 PM 4:00

1. 임의의 스트링 ω 의 순서를 뒤집는 연산 ω^R 은 다음과 같이 재귀적으로 정의된다.

$$(basis : \omega = \varepsilon) \Rightarrow \varepsilon^R = \varepsilon$$

$$(recursion : \omega = xa) \Rightarrow (xa)^R = a(x^R) \quad where\ x \in \Sigma^*, a \in \Sigma$$

이때 다음이 성립함을 증명하시오. (10점)

(1) $(ax)^R = x^R a$

(2) $(x^R)^R = x$

➔

(1) x 의 길이에 대한 mathematical induction

$(basis : \lvert x \rvert = 0)$	$(a\varepsilon)^R = (\varepsilon a)^R = a\varepsilon^R = a = \varepsilon a = \varepsilon^R a$
$(induction\ hypothesis)$	$\lvert x \rvert \le k \Rightarrow (ax)^R = x^R a$
$(induction\ step : \lvert x \rvert = k+1)$	$(axb)^R = b(ax)^R$ *by definition*
	$= bx^R a$ *by IH*
	$= (xb)^R a$ *by definition*

(2) x 의 길이에 대한 mathematical induction

$(basis : \lvert x \rvert = 0)$	$(\varepsilon^R)^R = \varepsilon^R = \varepsilon$
$(induction\ hypothesis)$	$\lvert x \rvert \le k \Rightarrow (x^R)^R = x$
$(induction\ step : \lvert x \rvert = k+1)$	$((xa)^R)^R = (ax^R)^R$ *by definition*
	$= (x^R)^R a$ *by* (1)
	$= xa$ *by induction hypothesis*

2. 다음 각 물음에 답하시오. (10점)

(1) 다음과 같은 language L을 생각해 보자.

$$L = \{x \in (a+b)^* \mid x\text{는 } b\text{로 끝나지 않고, } bb\text{를 포함하지 않는다}\}$$

$L = S^*$가 되는 유한한 language S를 찾으시오

(2) 다음과 같은 language L을 생각해 보자.

$$L = \{x \in (a+b)^* \mid bb\text{를 포함하지 않는다}\}$$

$L = S^*$가 되는 S가 존재하지 않음을 보이시오

➔

(1) $S = \{a, ba\}$

(2) 주어진 조건을 만족하는 S가 존재한다고 가정하자.

그리고 b 하나로 이루어진 스트링은 bb를 포함하지 않으므로 $b \in L = S^*$이다.

그런데, $S^* = (S^*)^*$ 이므로, $bb \in (S^*)^* = S^*$가 되어 가정에 위배된다.

따라서, 이러한 S는 존재하지 않는다.

3. 다음 연산 table을 생각해 보자

*	a	b	c
a	c	a	b
b	b	c	a
c	c	b	c

$\{a,b,c\}^+$ 로 이루어진 임의의 스트링 ω 에 대해 $value(\omega)$ 라는 함수를 다음과 같이 정의한다.

$value(\omega)$: ω의 leftmost symbol 부터 rightmost symbol까지 순차적으로 연산한 결과

예를 들어 $value(abcb) = ((a*b)*c)*b = (a*c)*b = b*b = c$ 이다.

이때 다음 각 language에 해당되는 minimal DFA를 구하시오 (10점)

(1) $L_a = \{x \mid value(x) = a\}$ (2) $L_{a^R} = \{x \mid value(x^R) = a\}$ (x^R 은 x 의 reverse)

➔

(1) Start symbol : S

*	a	b	c
S	A	B	C
$\underline{A}$	C	A	B
B	B	C	A
C	C	B	C

(2) Start symbol : A

*	a	b	c
A	F	A	B
B	B	C,F	A
C	A,C	B	C,F
F	.	.	.

$\Rightarrow$

*	a	b	c
A	F	A	B
B	B	CF	A
$\underline{F}$	.	.	.
$\underline{CF}$	AC	B	CF
AC	ACF	AB	BCF
$\underline{ACF}$	ACF	AB	BCF
AB	BF	ACF	AB
$\underline{BCF}$	ABC	BCF	ACF
BF	B	CF	A
ABC	ABCF	ABCF	ABCF
$\underline{ABCF}$	ABCF	ABCF	ABCF

4. 다음의 언어를 받아들이는 DFA를 정의하시오. (15점) (1) $L = \{x \in (0+1)^* \mid x$의 오른쪽 끝에서 3번째 symbol이 $1\}$ (2) $L = \{x \in (0+1)^* \mid x$의 오른쪽 끝에서 n번째 symbol이 $1\}$

➔

(1)

δ	0	1
q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_4	q_5
q_3	q_6	q_7
q_4	q_0	q_1
q_5	q_2	q_3
q_6	q_4	q_5
q_7	q_6	q_7

(2)

$$dfa\ M = (Q, \{0,1\}, \delta, q_0, F)$$

$$\begin{pmatrix} Q = \{q_0, q_1, \ldots, q_{2^n-1}\} \\ \delta(q_i, a) = q_{(2i+a) \bmod 2^n},\ 0 \le i \le n-1,\ a \in \{0,1\} \\ F = \{q_{2^{n-1}}, q_{2^{n-1}+1}, \ldots, q_{2^n-1}\} \end{pmatrix}$$

5. $M=(Q,\Sigma,\delta,s,F)$ 가 L 을 받아들이는 $\varepsilon-NFA$ 라고 하자. s 로 들어가는 transition이 없고, $F=\{f\}$ 오직 하나로 이루어져 있고, f 에서 나오는 transition이 없을 때, 다음의 automata가 받아들이는 언어를 기술하시오 (10점)

(1) s 로 부터 도달 가능한 모든 state에 대해 ε-transition 추가
(2) f 에 도달할 수 있는 모든 state로부터 f 까지 ε-transition 추가
(3) (1), (2) 둘다 추가
(4) 임의의 state $p,q\in Q$ 에 대해 p 로부터 도달할 수 있는 모든 state q 에 대해 p 에서 q 로 ε-transition 추가

➔
(1) $\{x \mid y\in L(M), \alpha x = y\}$
(2) $\{x \mid y\in L(M), x\beta = y\}$
(3) $\{x \mid y\in L(M), \alpha x\beta = y\}$
(4) $\{x \mid y\in L(M), x \textit{ is } \text{subsequence } \textit{of } y\}$

$$\left(\begin{array}{l} y=a_1a_2...a_n \\ \Rightarrow \text{subsequence } \textit{of } y=a_{i_0}a_{i_1}...a_{i_m},\ i_{k-1}<i_k\ (1\le k\le m), i_0\ge 1, i_m\le n, m<n \end{array}\right)$$

6. 다음과 같은 집합 A 와 $0 \sim (n-1)$ 의 symbol로 이루어진 language L_n 이 있다. (15점)

$$A = \{x \mid x = 2^n, n \geq 1\}, \quad L_n = \{x \mid x \text{ is } n-ary \text{ expression of } y \in A\}$$

(1) L_2 가 regular language인지 아닌지를 판명하고, 이를 증명하시오

(2) L_3 가 regular language인지 아닌지를 판명하고, 이를 증명하시오

➔ Problem solving… P85~86

(1) yes

$L_2 = 100*$

(2) no

Let $i(x)$ be integer represented by ternary expansion x.

$\forall n \geq 0,$

$\forall w \in L_3$ such that $|x| \geq n$,

$\forall x, y, z$ such that $w = xyz$, $y \neq \varepsilon$, $|xy| \leq n$,

Let $w^2 = xy^2z$,

$i(w) = i(xyz) = i(x)3^{|y|+|z|} + i(y)3^{|z|} + i(z) = 2^m$.

$i(w_2) = i(xy^2z) = i(x)3^{2|y|+|z|} + i(y)(3^{|y|+|z|} + 3^{|z|}) + i(z) = 2^{m_2}$.

$i(w_3) = i(xy^3z) = i(x)3^{3|y|+|z|} + i(y)(3^{2|y|+|z|} + 3^{|y|+|z|} + 3^{|z|}) + i(z) = 2^{m_3}$.

$i(w_3) - i(w_2) = 3^{2|y|+|z|}(i(x)(3^{|y|} - 1) + i(y))$.

그런데 $i(w_3) - i(w_2) = 2^{m_3} - 2^{m_2}$, $m_3 > m_2$ 이므로

2^{m_2} 가 $i(x)(3^{|y|} - 1) + i(y)$ 를 나눌 수 있어야 한다.

하지만, $i(x)(3^{|y|} - 1) + i(y)) \leq i(x)3^{|y|+|z|} + i(y)3^{|z|} + i(z) = i(w) = 2^m < 2^{m_2}$.

따라서 $w_3 \notin L_3$ 이므로

L_3 는 regular language가 아니다.

7. 다음 각각에 대해 참/거짓을 판명하고, 참인 경우는 좌변과 우변의 minimal DFA가 서로 같음을 보이고, 거짓인 경우는 counter example을 보여서 증명하시오 (20점)

(1) $(a^*b)^*aa^*bb = (a+b)^*abb$
(2) $a(ab+b)^*b = aa^*b(aa^*b)^*$
(3) $a(ba+aa)^*b = (ab+aa)^*ab$

➔

(1) 참

	a	b
S	A	S
A	A	B
B	A	F
F	A	S

(2) 거짓 : 우변에서는 abab가 생성되지만 좌변에서는 생성되지 않음

(3) 참

	a	b
S	A	
A	S	F
F	A	

8. 다음과 같은 두 regular expression이 있다. (15점)

$$A = 01(0+1)^* \quad B = (0+1)^*10$$

(1) bitwise-OR 연산인 $\vee$에 대해 다음과 같은 regular language를 표현하는 regular expression을 구하고 우리말로 표현하시오.

$$L(A) \vee L(B) = \{x \vee y \mid x \in L(A), y \in L(B), |x| = |y|\}$$

(2) $\Sigma = \{0, 1\}$인 임의의 두 regular language L_1, L_2에 대해 (1)의 $L_1 \vee L_2$가 regular language 임을 보이시오

➔

(1) $11 + (0+1)1(0+1) + (0+1)1(0+1)^*1(0+1)$

➔ 앞에서2번째, 뒤에서2번째가 각각 1인 string

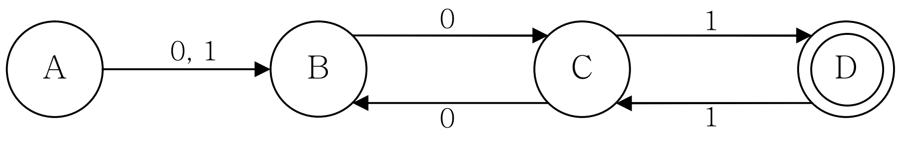
(2) $M_A = (Q_A, \{0,1\}, \delta_A, s_A, F_A), \quad M_B = (Q_B, \{0,1\}, \delta_B, s_B, F_B)$

$$M_{A\vee B} = (Q_A \times Q_B, \{0,1\}, \delta_{A\vee B}, [s_A, s_B], F_A \times F_B)$$

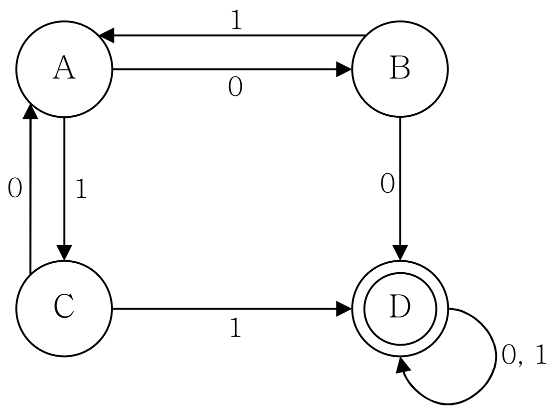
➔ $\begin{cases} \delta_{A\vee B}([p,q],0) = \{[\delta_A(p,0), \delta_B(q,0)]\} \\ \delta_{A\vee B}([p,q],1) = \{[\delta_A(p,1), \delta_B(q,1)], [\delta_A(p,1), \delta_B(q,0)], [\delta_A(p,0), \delta_B(q,1)]\} \end{cases}$

9. [finite automata ➔ regular expression] 다음 finite automata에 해당되는 regular expression을 구하시오 (start symbol은 A) (15점)

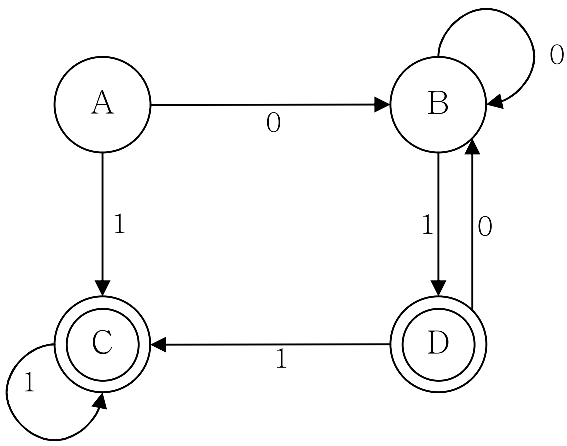
(1)



(2)



(3)



➔

(1)

$$A = 0B + 1B = (0+1)B$$
$$B = 0C$$
$$C = 0B + 1D$$
$$D = 1C + \varepsilon = (0+1)^*$$

$$C = 0B + 1D = 00C + 11C + 1 = (00+11)^*1$$
$$B = 0C = 0(00+11)^*1$$
$$A = (0+1)0(00+11)^*1$$

(2)

$$A = 0B + 1C + \varepsilon$$
$$B = 0D + 1A = 1A + 0(0+1)^*$$
$$C = 0A + 1D = 0A + 1(0+1)^*$$
$$D = 0D + 1D + \varepsilon = (0+1)^*$$

$$\begin{aligned} A &= 0B + 1C + \varepsilon \\ &= 0(1A + 0(0+1)^*) + 1(0A + 1(0+1)^*) + \varepsilon \\ &= (01+10)A + (00+11)(0+1)^* + \varepsilon \\ &= (01+10)^*(00+11)(0+1)^* \end{aligned}$$

(3)

$$A = 0B + 1C$$
$$B = 0B + 1D$$
$$C = 1C + \varepsilon = 1^*$$
$$D = 0B + 1C + \varepsilon = 0B + 11^* + \varepsilon$$

$$\begin{aligned} B &= 0B + 1D \\ &= 0B + 10B + 111^* + 1 \\ &= (0+10)^*(111^*+1) \\ A &= 0B + 1C \\ &= 0(0+10)^*(111^*+1) + 11^* \end{aligned}$$

10. 다음 각 물음에 답하시오 (15점)
(1) $L = \{ww^R \mid w \in (0+1)^*\}$ 이 정규언어가 아님을 증명하고, $L(G) = L$ 인 $cfg\ G$ 를 구하시오.

(2) 어떠한 $cfg\ G = \{V, \Sigma, P, S\}$ 의 production rule이 모두 $A \to xB$ 이거나 $A \to Bx$ 혹은 $A \to x$ ($A, B \in V,\ x \in \Sigma^*$)의 형태라 하자. $L(G)$ 가 반드시 정규언어(regular language)일 필요는 없다는 것을 (1)번의 결과를 이용해서 보이시오.

➔
(1)

$\forall n \geq 0,$
$\exists w = 0^n 110^n\ .\ni.\ |w| = 2n+2$
$\forall x, y, z\ .\ni.\ w = xyz,\ y \neq \varepsilon,\ |xy| \leq n,$
$\Leftrightarrow \forall i, j\ .\ni.\ i \geq 0,\ j > 0,\ i + j \leq n,\ x = 0^i,\ y = 0^j,\ z = 0^{n-i-j}110^n$
$\exists k = 0,\ xy^k z = xz = 0^i 0^{n-i-j} 110^n = 0^{n-j} 110^n \notin L$ since $j > 0$

$\therefore\ L$ is not regular

CFG : $S \to 0S0 \mid 1S1 \mid \varepsilon$

(2)
$S \to 0S0 \mid 1S1 \mid \varepsilon$ 를 다음의 동등한 CFG로 변환할 수 있다.

$S \to 0A \mid 1B \mid \varepsilon$
$A \to S0$
$B \to S1$

이는 주어진 G 의 조건을 모두 만족한다. 하지만 (1)에서 이것이 regular language가 아님을 보였으므로 G 가 regular language 라는 것의 counter example이 된다.

11. 다음 각 물음에 답하시오 (20점)

(1) 다음의 CFG는 b 의 개수가 a 의 개수의 2배인 string들을 만들어 내는 CFG이다. 이 때 A_1, A_2, B 가 어떤 string들을 만들어내는지 말로 설명하라.

$$
\begin{aligned}
S &\to \varepsilon \mid aBB \mid bA_1 \\
A_1 &\to aB \mid bA_2 \\
A_2 &\to aS \mid bA_1A_2 \mid bA_2A_1 \\
B &\to bS \mid aBBB
\end{aligned}
$$

(2) 다음 언어의 CFG를 구하라.

$L_2 = \{x \in (a+b)^* \mid x$의 b의 개수는 a의 개수의 2배이다.$\}$

(3) $n \geq 1$ 일 때 다음 언어의 CFG를 구하라.

$L_n = \{x \in (a+b)^* \mid x$의 b의 개수는 a의 개수의 n 배이다.$\}$

➔

(1) A_1: b 의 개수가 a 의 개수의 2배 보다 하나 적은 string

A_2: b 의 개수가 a 의 개수의 2배 보다 2개 적은 string

B: b 의 개수가 a 의 개수의 2배 보다 하나 많은 string

(2)

$$
\begin{aligned}
S &\to \varepsilon \mid aBBB \mid bA_1 \\
A_1 &\to aBB \mid bA_2 \\
A_2 &\to aB \mid bA_1 \\
A_3 &\to aS \mid bA_3A_1 \mid bA_2A_2 \mid bA_1A_3 \\
B &\to bS \mid aBBBB
\end{aligned}
$$

(3) $G = (V, \{a,b\}, P, S)$

$V = \{S, A_1, A_2, \ldots, A_n, B\}$

$$
P = \left\{
\begin{aligned}
S &\to \varepsilon \mid aB^n \mid bA \\
A_1 &\to aB^{n-1} \mid bA_2 \\
&\vdots \\
A_i &\to aB^{n-i} \mid bA_{i+1} \\
&\vdots \\
A_{n-1} &\to aB^{n-(n-1)} \mid bA_n \\
A_n &\to aS \mid bA_nA_1 \mid bA_{n-1}A_2 \mid \cdots \mid bA_2A_{n-1} \mid bA_1A_n \\
B &\to bS \mid aB^{n+1}
\end{aligned}
\right\}
\qquad (B^n = \underbrace{BB\cdots BB}_{n})
$$

2004 Fall

CS322 오토마타 중간고사 (180점)

2004년 10월 22일 PM 4:10-6:30

1. 다음의 집합이 finite, countably infinite, uncountable 중에 어디에 속하는지 결정하고 이를 설명하시오. (20점) (1) Java로 쓰여진 모든 프로그램의 집합 (2) 집합 B를 $\{F,T\}$라 할 때 $B \times B \times \cdots$ (infinite product) (3) Countably infinite 집합의 finite product Ex.) A_i가 각각 countably infinite 일 때, $A_1 \times \cdots \times A_n$ (4) 모든 원소 s가 다음의 두 조건을 만족하는 집합 a. s는 1과 2로 이루어진 스트링이다. b. s의 n번째 문자가 2라면 $n+1$번째 문자는 반드시 1이어야 한다.

➔

(1) countably infinite.

Σ^*의 부분 집합

(2) uncountable

만약 $f : N \to A$로 가는 bijection이 존재한다고 하자.

$f(i) = b_{i0}b_{i1} \cdots b_{ij} \cdots$ 이라고 할 때,

$$C_i = F \text{ if } b_{ii} = T$$
$$\quad\; T \text{ if } b_{ii} = F$$

로 정의하자.

이 경우, $C = C_0C_1 \cdots C_i \cdots$ 는 모든 i에 대해서 $f(i)$와 같을 수 없다. 그러나 C는 A에 속해야 하므로 contradiction에 도달한다. 따라서 uncountable.

(3) countably infinite.

두 countably infinite 집합의 Cartesian product가 countably infinite (숙제1-2 참고)하다는 것으로부터 다음과 같이 mathematical induction으로 증명하면 된다.

“0보다 큰 모든 자연수 n에 대해 임의의 countably infinite 집합, $A_1, \cdots, A_n$이 있을 때, $A_1 \times \cdots \times A_n$은 countably infinite 하다.”

$(basis : n = 1)$ By assumption, A is countably infinite.

$(induction\ hypothesis)$ $A_1 \times \cdots \times A_k$ is countably infinite.

$(induction\ step : n = k + 1)$ From IH and assumption that A_{k+1} is countably infinite, cartesian product of two countably infinite set, $(A_1 \times \cdots \times A_k) \times A_{k+1}$ is countably infinite.

(4) uncountable

0 과 1 의 sequence를 원소로 갖는 집합 S' 은 uncountable임이 알려져 있다. 여기서는 S 와 S' 이 1:1 onto(bijection)관계에 있다는 것을 보임으로서 S 가 uncountable임을 증명하겠다.

- $S \to S'$: s 에서 나오는 모든 21 pair를 0 으로 대체함으로써 S 의 모든 원소를 S' 의 원소로 표현할 수 있다.
- $S' \to S$: S' 의 원소 s' 의 모든 0 을 21 pair로 대체함으로써 S' 의 모든 원소를 S 의 원소로 표현할 수 있다.

2. 다음의 각각의 등식에서, 모든 language A, B 에 대해 다음이 성립하는 지의 여부를 판단하여라. 성립한다면 증명을 보이고, 그렇지 않다면 반례를 드시오. (10점)
(단 $A^R = \{x^R \mid x \in A)\}$, x^R 은 string의 reverse연산을 의미한다.)

(1) $(A^R)^* = (A^*)^R$
(2) $(A^+)^* = A^*$

➔

(1) True

- $(A^R)^*$ 에 속한 general string을 x 라 하자. x 는 $x_1 \cdots x_n$, $x_i \in A^R$ 로 분해된다. x_i 를 reverse 시켰을 때, x_i'이 된다고 하면, $x_i = x_i'^R$ 이 된다. 여기서, $x_i' \in A$ 임을 알 수 있다. 따라서, $x = x_1 \cdots x_n = x_1'^R \cdots x_n'^R$ 이고, $x_1'^R \cdots x_n'^R = (x_n' \cdots x_1')^R$ 이다.
$x_n' \cdots x_1' \in A$ 이므로 $(A^R)^* \subseteq (A^*)^R$ 이다.
- $(A^*)^R$ 에 속한 general string을 $(x_1 \cdots x_n)^R$, $x_i \in A$ 이라 하자. 이는 $x_n^R \cdots x_1^R$ 임을 알 수 있고, $x_i \in A$ 이므로, $x_n^R \cdots x_1^R \in (A^R)^*$ 가 된다. 따라서, $(x_1 \cdots x_n)^R = x_n^R \cdots x_1^R \in (A^R)^*$ 이므로, $(A^R)^* \supseteq (A^*)^R$ 이다.

그러므로, 주어진 등식은 성립한다.

(2) True

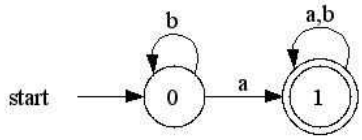
- $A \subseteq A^+$ 이므로, $(A^+)^* \supseteq A^*$ 이다.
- $A^+ \subseteq A^*$ 이므로, $(A^+)^* \subseteq (A^*)^*$ 이다. 그런데, $(A^*)^* = A^*$ 이므로 $(A^+)^* \subseteq A^*$

그러므로, 주어진 등식은 성립한다.

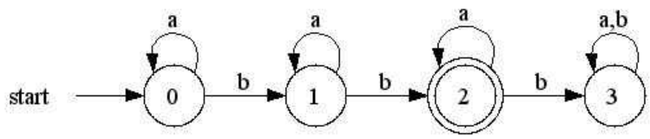
3. $\Sigma = \{a, b\}$에 대해서 다음의 집합을 받아들이는 minimal state DFA 작성하시오. (20점)
(필요한 경우 dead state를 포함해 transition function을 total function으로 만드시오.)
(1) 적어도 한 개의 a를 포함하는 모든 string
(2) 정확히 두 개의 b를 포함하는 모든 string
(3) 최대 세 개의 a를 포함하는 모든 string
(4) (1), (2)의 조건을 둘 다 만족하는 모든 string

➔

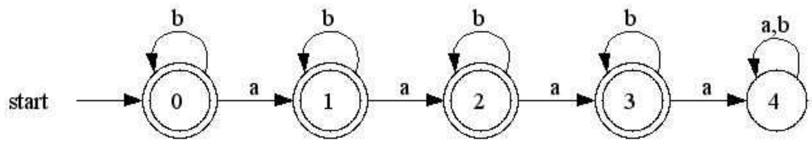
(1)



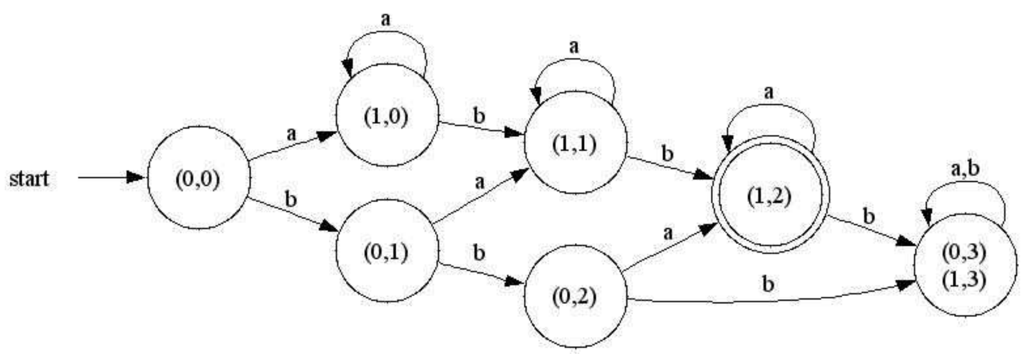
(2)



(3)



(4)
(1),(2) 의 product automata
3은 dead state이므로 (0,3), (1,3)은 equivalent 함.



4. DFA인 $A=\{Q,\Sigma,\delta,q_0,F\}$ 에서 $\hat{\delta}(q,\varepsilon)=q$ 이고 $\hat{\delta}(q,wa)=\delta(\hat{\delta}(q,w),a)$ 인 $\hat{\delta}$ 이 있다고 할 때, $q\in Q$ 인 모든 state q 와, $x,y\in\Sigma^*$ 인 모든 string x,y 에 대해서 $\hat{\delta}(q,xy)=\hat{\delta}(\hat{\delta}(q,x),y)$ 임을 증명하여라. (10점)

➔

Mathematical Induction으로 다음과 같이 증명한다.

$(basis: |y|=0)$ $\quad \hat{\delta}(q,x\varepsilon)=\hat{\delta}(q,x)=\hat{\delta}(\hat{\delta}(q,x),\varepsilon)$

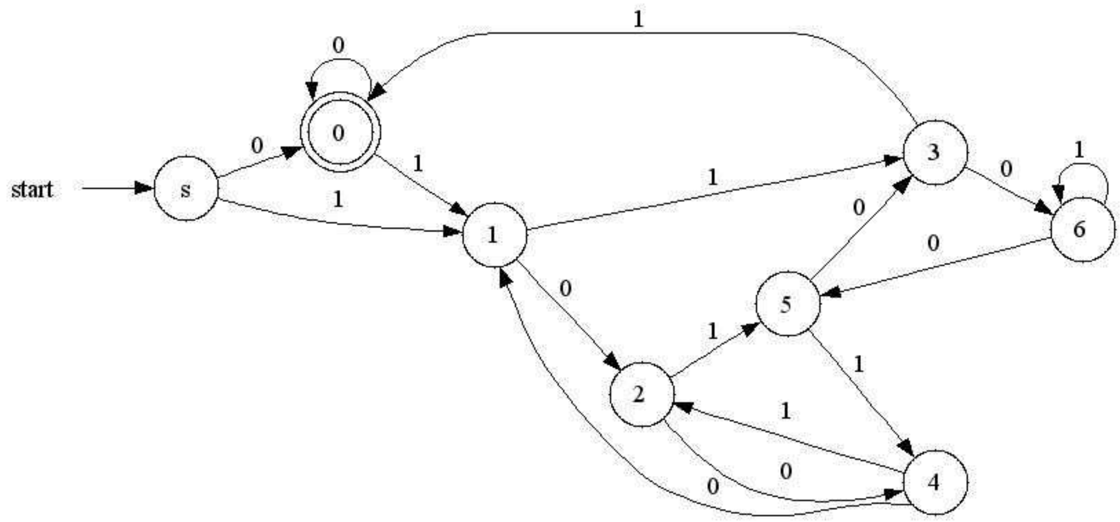
$(induction\ hypothesis)$ $\quad |y|\le k \Rightarrow \hat{\delta}(q,xy)=\hat{\delta}(\hat{\delta}(q,x),y)$

$(induction\ step: |y|=k+1)$

$$\begin{aligned}\hat{\delta}(q,x(ya)) &= \hat{\delta}(q,(xy)a)\\ &= \delta(\hat{\delta}(q,xy),a)\ \ by\ definition\\ &= \delta(\hat{\delta}(\hat{\delta}(q,x),y),a)\ \ by\ IH\\ &= \hat{\delta}(\hat{\delta}(q,x),ya)\ \ by\ definition\end{aligned}$$

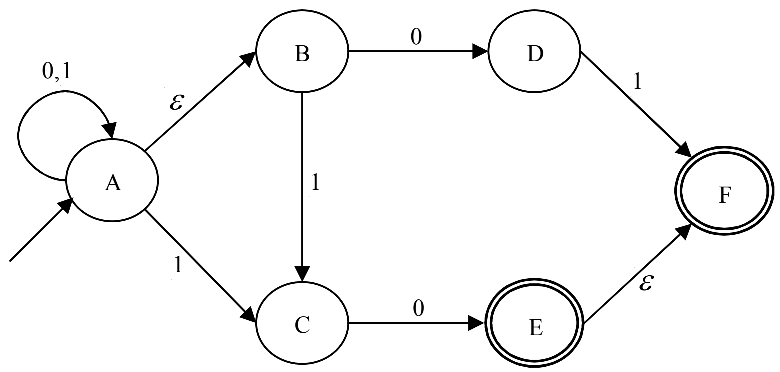
5. 7로 나누어지는 이진수를 MSB(Most Significant Bit)부터 읽었을 때 받아들이는 DFA를 작성하시오. (10점)

➔



$$\delta(s,0) = \delta(0,0)$$
$$\delta(s,1) = \delta(0,1)$$
$$\left.\begin{aligned}\delta(i,0) &= (2*i+0) \bmod 7 \\ \delta(i,1) &= (2*i+1) \bmod 7\end{aligned}\right\} for\ 0 \le i \le 6$$

6. 다음의 NFA에 대해 질문에 답하시오. (20점)



(1) 이 NFA를 수업 시간에 다룬 simultaneous equation을 세우고 substitution을 이용해 푸는 방법으로 regular expression으로 변환하시오. (식을 세우고 푸는 과정을 기술할 것)
(2) 이 NFA의 각 state의 e-closure를 구하시오.
(3) Subset construction을 사용해 이 NFA를 DFA로 변환하시오. (DFA의 state는 subset construction에서 나타나는 NFA의 state의 set으로 기술할 것)
(4) (3)에서 구한 DFA를 (1)에서와 같은 방법으로 regular expression으로 변환해서 (1)과 똑같은 regular expression으로 변환됨을 보이시오. (식을 세우고 푸는 과정을 기술할 것)

➔

(1)

$A = (0+1)A + B + 1C = (0+1)A + 01 + 10 + 10 = (0+1)^*(01+10)$

$B = 0D + 1C = 01 + 10$

$C = 0E = 0$

$D = 1F = 1$

$E = F + \varepsilon = \varepsilon + \varepsilon = \varepsilon$

$F = \varepsilon$

(2)

$\mathrm{ECLOSE}(A) = \{A, B\}$

$\mathrm{ECLOSE}(B) = \{B\}$

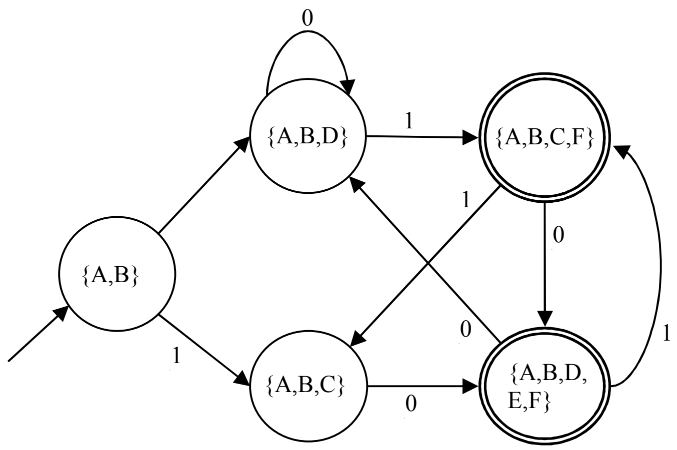
$\mathrm{ECLOSE}(C) = \{C\}$

$\mathrm{ECLOSE}(D) = \{D\}$

$\mathrm{ECLOSE}(E) = \{E, F\}$

$\mathrm{ECLOSE}(F) = \{F\}$

(3)



(4)

Let

$V = \{A, B\}$

$W = \{A, B, D\}$

$X = \{A, B, C\}$

$Y = \{A, B, C, F\}$

$Z = \{A, B, D, E, F\}$

Then,

$V = 0W + 1X = 0V + 01 + 1V + 10 = (0+1)^*(01+10)$

$W = 0W + 1Y = 0W + 1X + 1 = V + 1$

$X = 0Z + 1X = 0W + 0 + 1X = V + 0$

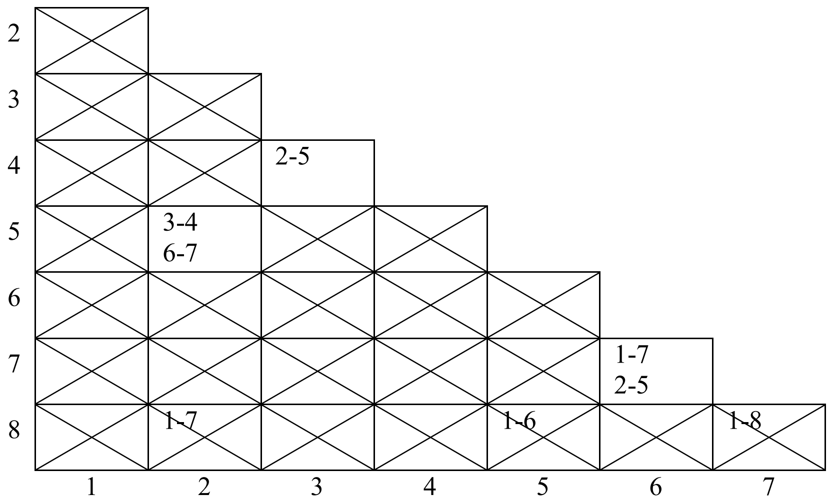
$Y = 0Z + 1X + \varepsilon = X + \varepsilon$

$Z = 0W + 1Y + \varepsilon = W + \varepsilon$

7. 다음의 DFA를 equivalent한 minimal state DFA로 변환하시오. (10점)
(변환 과정을 기술할 것)

	a	*b*
→1	7	8
2	3	6
3F	2	4
4F	5	3
5	4	7
6	7	5
7	6	2
8	8	1

➔



Minimized DFA

	a	*b*
→1	67	8
25	34	67
34*F*	25	34
67	67	25
8	8	1

8. $\Sigma = \{a,b\}$ 일 때 다음의 문장들이 참인지 거짓인지 답을 하고 참인 경우 증명을, 거짓인 경우 반례를 제시하시오. (regular language들이 union, concatenation, closure, complementation, intersection에 대해서 닫혀있는 것은 가정해도 됨) (25점)

(1) 임의의 $N, L \subseteq \Sigma^*$ 에 대해 N 이 nonregular 이고 $N \subseteq L$ 이면 L 도 nonregular 이다.
(2) 임의의 $L, N \subseteq \Sigma^*$ 에 대해 $L \subseteq N$ 이고 N 이 nonregular 이면 L 도 nonregular 이다.
(3) 임의의 $R, L \subseteq \Sigma^*$ 에 대해 R 이 regular 이면, 모든 L 에 대해 $R \cup L$ 도 regular 이다.
(4) 임의의 $N \subseteq \Sigma^*$ 에 대해 N 이 nonregular 이면, N 의 complement도 nonregular 이다.
(5) 임의의 $N_1, N_2 \subseteq \Sigma^*$ 에 대해, N_1, N_2 가 nonregular 이면 $N_1 \cap N_2$ 도 nonregular 이다.

➔

(1) False
$N = \{a^n b^n \mid n \geq 0\}$ 일 때,
$L = \Sigma^*$ 는 $N \subseteq L$ 이지만, regular

(2) False
$N = \{a^n b^n \mid n \geq 0\}$ 일 때,
$L = \{a,b\}$ 인 경우, $L \subseteq N$ 이지만, L 은 regular

(3) False
$R = \varnothing$ 일 때
$L = \{a^n b^n \mid n \geq 0\}$ 에 대해
$R \cup L = L$ 가 regular 이다는 것은 거짓

(4) True
Proof by contradiction
N 의 complement ($\overline{N}$) 가 regular 라고 하자.
그러면, $\overline{\overline{N}}$ 는 regular language가 complement에 대해 닫혀있으므로 역시 regular 이다.
하지만, $\overline{\overline{N}} = N$ 이고 이는 N 이 nonregular 라는 가정에 모순이 된다.
따라서, $\overline{N}$ 는 nonregular 이다.

(5) False
$N_1 = \{a^n b a^n \mid n \geq 0\}$
$N_2 = \{b^n a b^n \mid n \geq 0\}$
일 때, $N_1 \cap N_2 = \varnothing$ 이고 regular

9. L 이 language이고, a 는 symbol일 때, $L_a = \{w \mid aw \text{ is in } L\}$ 인 집합을 생각해 보자.
이 때 "L 이 regular이면, L_a 또한 regular이다"를 다음과 같이 증명하고자 한다.
L 이 regular이므로 $L = L(A)$ 인 DFA $A = (Q_A, \sum, \delta_A, q_A, F_A)$ 이 존재한다. A 로부터 $L(L_a) = L(B)$ 인 NFA-ε $\mathrm{B} = (Q_B, \sum, \delta_B, q_B, F_B)$ 를 구성하면 L_a 이 regular임을 보일 수 있다. 우선 Q_B, q_B 는 $Q_B = Q_A \cup \{q_B\}$, $q_B \notin Q_A$ 로 정의하자. (15점)

(1) δ_B, F_B 를 정의하시오.
(2) $L(L_a) = L(B)$ 임을 증명하시오.

➔
(1)
$\delta_B(q_B, \varepsilon) = \{r \in Q_B \mid r = \delta_A(q_A, a)\}$
$or\ \ \delta_B(q_B, \varepsilon) = \delta_A(q_A, a)$
$For\ all\ q \in Q_A, b \in \Sigma, \delta_B(q, b) = \{r \in Q_B \mid r = \delta_A(q, b)\}$
$or\ \ \delta_B(q, b) = \delta_A(q, b)$
$F_B = F_A$

(2)
- $L(L_a) \subseteq L(B)$
$b_1 \cdots b_n \in L(L_a)$ 이라 하면, L_a 의 정의에 의해 $ab_1 \cdots b_n \in L = L(A)$.

$$\begin{array}{lcl} \delta_A(q_A, a) = q_1 & & \varepsilon^*(q_B) = \{q_1\} \\ \delta_A(q_1, b_1) = q_2 & ➔ & \delta_B(q_1, b_1) = \{q_2\} \\ \vdots & & \vdots \\ \delta_A(q_n, b_n) = q_f \in F_A & & \delta_B(q_n, b_n) = \{q_f\} \in F_B \end{array}$$

인 $q_1, \cdots, q_n, q_f$ 가 존재한다.
그리고, $\varepsilon(q) = \varnothing,\ for\ q \in Q_A$ 이므로,

$$\begin{aligned} \hat{\delta}_B(q_B, b_1 \cdots b_n) &= \varepsilon^*(\delta_B(\cdots \varepsilon^*(\delta_B(\varepsilon^*(q_B), b_1)), \cdots, b_n)) \\ &= \varepsilon^*(\delta_B(\cdots \varepsilon^*(\delta_B(\{q_1\}, b_1)), \cdots, b_n)) \\ &= \varepsilon^*(\delta_B(\cdots \varepsilon^*(\{q_2\}), \cdots, b_n)) \\ &= \varepsilon^*(\delta_B(\cdots \{q_2\}, \cdots, b_n)) \\ &\vdots \\ &= \varepsilon^*(\delta_B(\{q_n\}, b_n)) \\ &= \varepsilon^*(\{q_f\}) \\ &= \{q_f\} \subseteq F_A = F_B \end{aligned}$$

이고, $b_1 \cdots b_n \in L(B)$

- $L(B) \subseteq L(L_a)$

$b_1 \cdots b_n \in L(B)$ 이라 하자. $L(B)$ 의 정의에 따르면,

$$
\begin{array}{ccc}
\delta_B(q_B, \varepsilon) = \{q_1\} & & \delta_A(q_A, a) = q_1 \\
\delta_B(q_1, b_1) = \{q_2\} & \Rightarrow & \delta_A(q_1, b_1) = q_2 \\
\vdots & & \vdots \\
\delta_B(q_n, b_n) = \{q_f\} \subseteq F_B & & \delta_A(q_n, b_n) = q_f \in F_A
\end{array}
$$

인 $q_1, \cdots, q_n, q_f$ 가 존재한다.

따라서,

$$
\begin{aligned}
\hat{\delta}_A(q_A, ab_1 \cdots b_n) &= \delta_A(\cdots(\delta_A(\delta_A(\hat{\delta}_A(q_A, \varepsilon), a), b_1), \cdots), b_n) \\
&= \delta_A(\cdots(\delta_A(\delta_A(q_A, a), b_1), \cdots), b_n) \\
&= \delta_A(\cdots(\delta_A(q_1, b_1), \cdots), b_n) \\
&\vdots \\
&= \delta_A(q_n, b_n) \\
&= q_f \in F_A
\end{aligned}
$$

이고, $ab_1 \cdots b_n \in L(A) = L$ 이므로, L_a 의 정의에 의해 $b_1 \cdots b_n \in L(L_a)$

10. 다음 language L이 regular language 가 아님을 pumping lemmar를 사용해 증명하시오. (10점)

$L = \{a^i b^j \mid 2i \le j\}$

➔

$\forall n \ge 0,$

$\exists w \in L$ such that $|w| \ge n$,

$w = a^n b^{2n}$ 이라 하자.

$\forall x, y, z$ such that $w = xyz$, $y \ne \varepsilon$, $|xy| \le n$,

$y = a^i$, $1 \le i \le n$ $(\because y \ne \varepsilon)$

그런데,

$xy^2 z = a^{n+i} b^{2n} \notin L$ $(\because 1 \le i)$

따라서 pumping lemma에 의해 L은 regular language가 아니다.

11. 다음의 language에 대한 CFG를 구하시오. (15점)

(1) $\{x^R 2y \mid x, y \in \{0,1\}^*, x$는 y의 substring$\}$ (x^R 은 string x 의 reverse)
(2) $\{x^R 2y \mid x, y \in \{0,1\}^*, x$는 y의 subsequence$\}$ (a_i, b_i 이 symbol 일 때, $x = a_1 \cdots a_k$ 가 $y = b_1 \cdots b_n$ 의 subsequence 이려면, 자연수 $j_1, \cdots, j_k$ 가 존재해서 $1 \le j_1 < j_2 < \cdots < j_k \le n$ 이고, $a_1 = b_{j_1}, a_2 = b_{j_2}, \cdots, a_k = b_{j_k}$ 이어야 한다.)

➔
(1)
$S \to A \mid S0 \mid S1$
$A \to 0A0 \mid 1A1 \mid B$
$B \to B0 \mid B1 \mid 2$

(2)
$S \to A \mid S0 \mid S1$
$A \to 0A0 \mid 1A1 \mid A0 \mid A1 \mid B$
$B \to B0 \mid B1 \mid 2$

12. Language를 기술하는데 있어서 language에 속하는 하나의 문장에 대해 이를 표현하는 방법이 두 가지 이상이면 그러한 기술을 ambiguous하다고 말한다. Regular expression의 경우 예를 들면 다음과 같은 경우 ambiguous 하다.

- $0 \cup 0$: 0 을 표현하는 방법이 두 가지임
- $(0 \cup 01)(\varepsilon \cup 1)$: 01 을 표현하는 방법이 두 가지
- $(0 \cup 000)^*$: $0^{n \geq 2}$ 을 표현하는 방법이 최소한 두 가지 이상
- ε^* : ε 을 표현하는 방법이 countably infinite 함

Regular expression 이 unambiguous 하다는 것의 정의를 regular expression에 대한 structural induction으로 정의하면 다음과 같이 된다.

a) $\varnothing$, ε , a (any symbol) 는 unambiguous 하다.
b) (E) 는 E 가 unambiguous 하면 unambiguous 하다.
c) $E_1 \cup E_2$
d) $E_1 E_2$
e) E^*
f) a)-e)를 만족하지 않는 다른 어떤 것도 unambiguous 하지 않다.

c),d),e)에 대해 위의 정의를 완성하시오. (15점)

➔
c) $E_1 \cup E_2$ 는 E_1, E_2 가 unambiguous 하고 $L(E_1) \cap L(E_2) = \varnothing$ 이면 unambiguous 하다.

d) $E_1 E_2$ 는 E_1, E_2 가 unambiguous 하고 모든 $x \in L(E_1 E_2)$ 에 대해 $x_1 x_2 = x$, $x_1 \in L(E_1)$, $x_2 \in L(E_2)$ 인 x_1, x_2 가 정확히 하나 존재하면 unambiguous 하다.

e) E^* 는 E 가 unambiguous 하고 모든 $x \in L(E^*)$ 에 대해 정확히 하나의 $n \geq 0$ 이 존재해서 $x_1 \cdots x_n = x$ 인 $x_i \in L(E)$ 의 sequence $(x_1, \cdots, x_n)$ 이 정확히 하나 존재하면 unambiguous 하다.

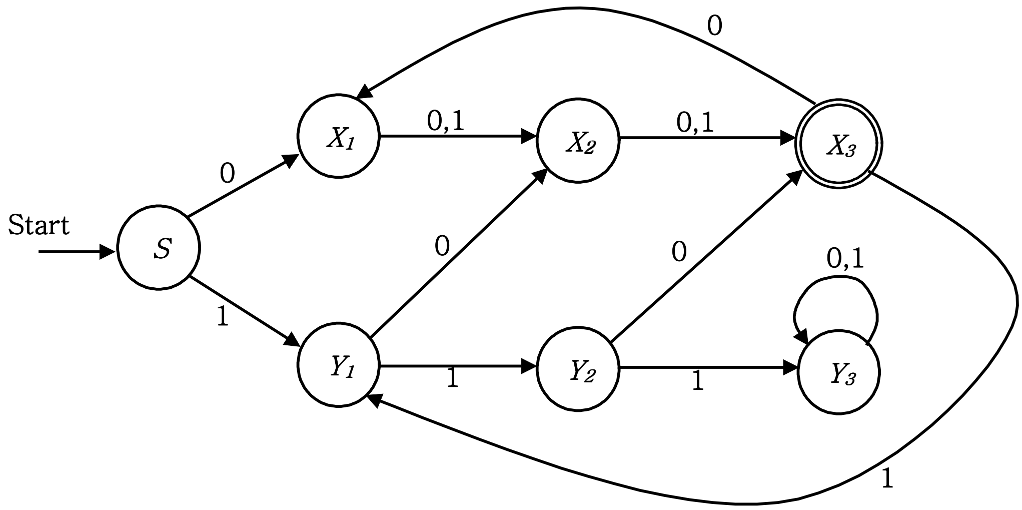
▲ Section A : Finite Automata

1. 다음 각각의 language를 받아들이는 finite automata를 제시하시오.

(1) L_1 = { $x\in\{0,1\}^*$ | $x=(abc)^n$ for $n\geq 1$, a=0 or b=0 or c=0 } <15점>
(즉, 길이 3n의 문자열로서, 처음부터 3개씩 자른 n개의 뭉치가 0을 적어도 한 개씩은 가지고 있다.)
(단, DFA를 구성하여라.)

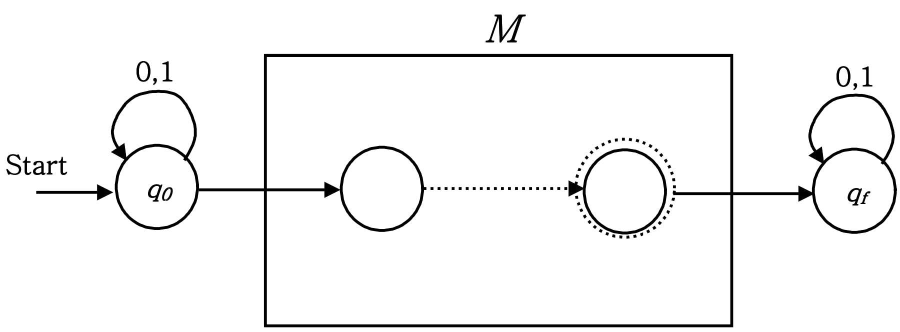
<Solution>

이 language는 start state S와 X_n, Y_n (n=1,2,3)의 모두 7개의 state로 이루어진 DFA로 표현된다.
여기서 X는 0을 하나라도 본 state를, Y는 하나도 보지 못한 state를 의미하며, n은 각각의 뭉치에서 처리한 문자의 개수를 나타낸다. 또한, final state는 X_3가 된다.
이와 같은 사실을 바탕으로, 이를 다음과 같은 transition diagram으로 나타낼 수 있다.



(2) 주어진 임의의 fa M에 대해, L_2 = { $w\in\{0,1\}^*$ | w의 substring이 M에 의해 accept } <10점>
(단, ε-NFA를 구성하여라.)

<Solution>



▲ Section B : Regular Expressions

2. 다음 각 문제가 요구하는 바에 따라, 적절한 Regular Expression을 제시하시오.

(1) 다음의 grammar가 만들어 내는(generate) language를 나타내는(denote) regular expression을 제시하시오. (단, 대문자는 nonterminal symbol을, 소문자와 세미콜론(;)은 terminal symbol을 의미한다.) <10점>

P → b D S e
D → d ; D | ε
S → S ; s | s

<solution>
D = d ; D + ε = (d ;)*
S = S ; s + s = s (; s)*
⇒ P = b (d ;)* s (; s)* e

(2) 세 번째 *b*가 나타난 후에는 *a*가 나타나지 않는, *a*와 *b*로 구성된 string들의 집합을 원소로 갖는 language에 대한 regular expression을 제시하시오. <10점>

<Solution>
Note : *b*가 반드시 세 개 이상 나올 필요는 없다.
$a^* ba^* ba^* b^* + a^* ba^* b^* + a^* b^*$
또는 $a^*(b+\varepsilon)a^*(b+\varepsilon)a^* b^*$

3. 다음 물음에 답하시오

(1) {0,1}을 alphabet으로 가지는 language L에 대한 regular expression e가 주어졌을 때, regular expression $Rev(e)$을 정의하시오. (단, $L(Rev(e)) = L(e)^R$) <10점>

<solution>

Base Cases :

$Rev(\phi) = \phi$

$Rev(a) = a$ for all $a \in \Sigma$

Recursive Cases :

$Rev(\alpha+\beta) = Rev(\alpha)+Rev(\beta)$ (Union)

$Rev(\alpha\beta) = Rev(\beta)Rev(\alpha)$ (Concatenation)

$Rev(\alpha^*) = (Rev(\alpha))^*$ (Star)

No more.

(2) 위 (1)에서 정의한 $Rev(e)$를 regular expression $(0(01+11)^*)^*$에 적용해 보시오. <5점>

$Rev((0(01+11)^*)^*)$

$=$

<solution>

$Rev((0(01+11)^*)^*)$

$= (Rev(0(01+11)^*))^*$

$= (Rev((01+11)^*)Rev(0))^*$

$= ((Rev(01+11))^*Rev(0))^*$

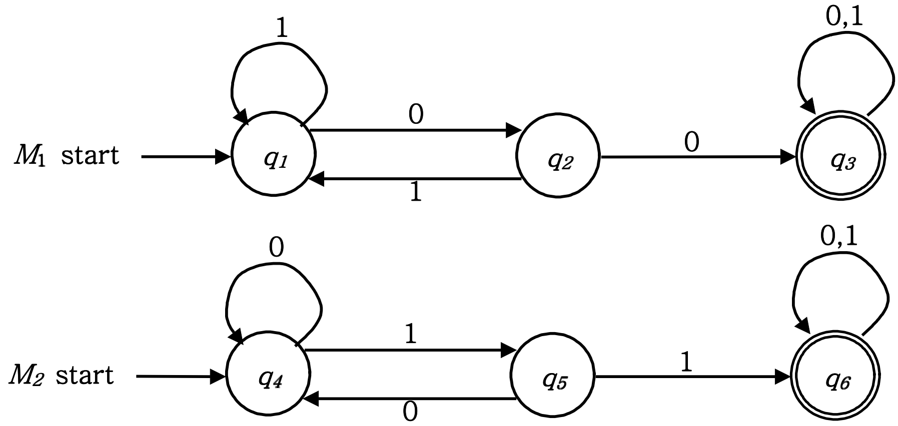
$= ((Rev(01)+Rev(11))^*Rev(0))^*$

$= ((Rev(1)Rev(0)+Rev(1)Rev(1))^*Rev(0))^*$

$= ((10+11)^*0)^*$

▲ Section C : Finite Automata & Regular Expressions

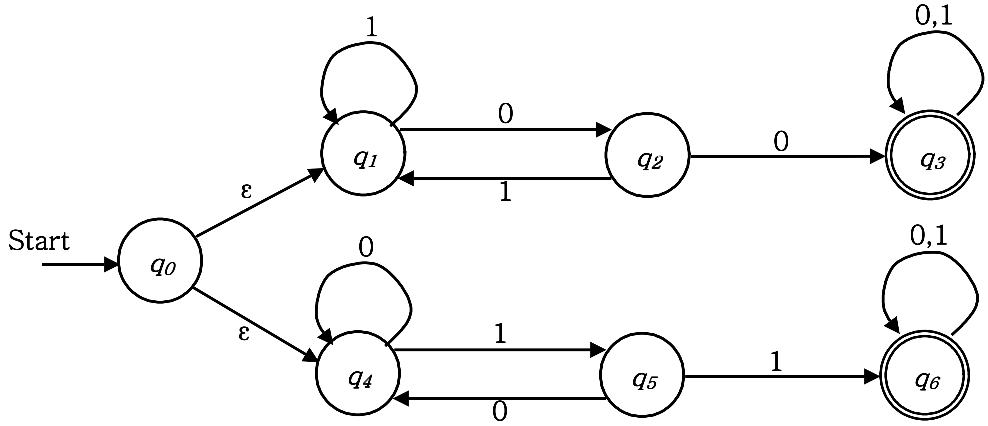
4. 다음과 같은 2개의 Finite Automata가 주어졌을 때, 물음에 답하시오.



$L=L(M_1)\cup L(M_2)$이라 하자.

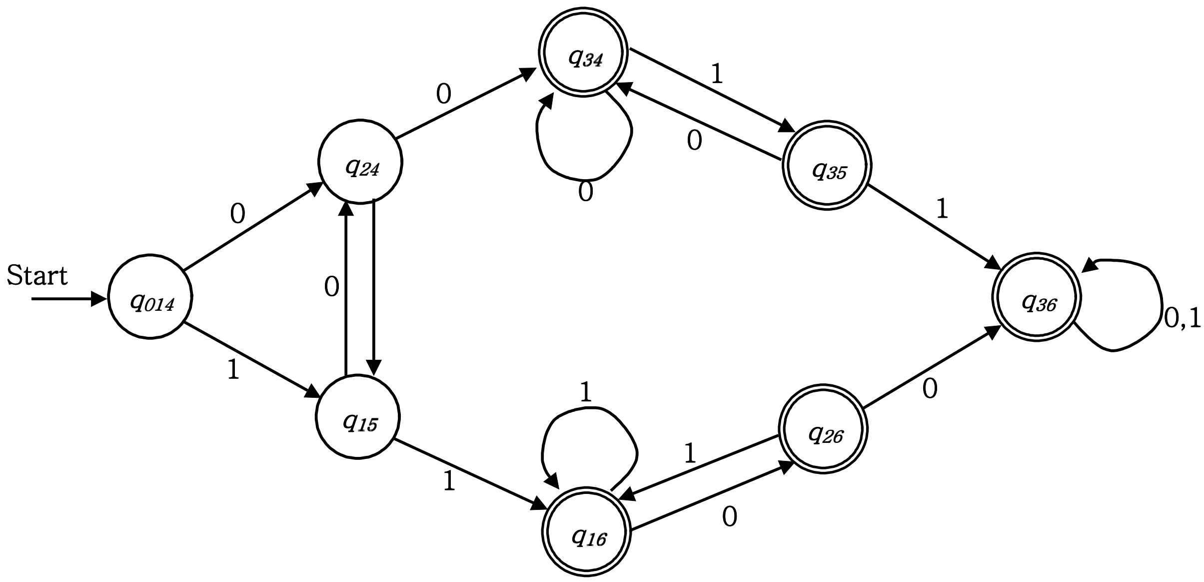
(1) $L=L(M)$인 ε-NFA를 구성하시오. <5점>

<solution>



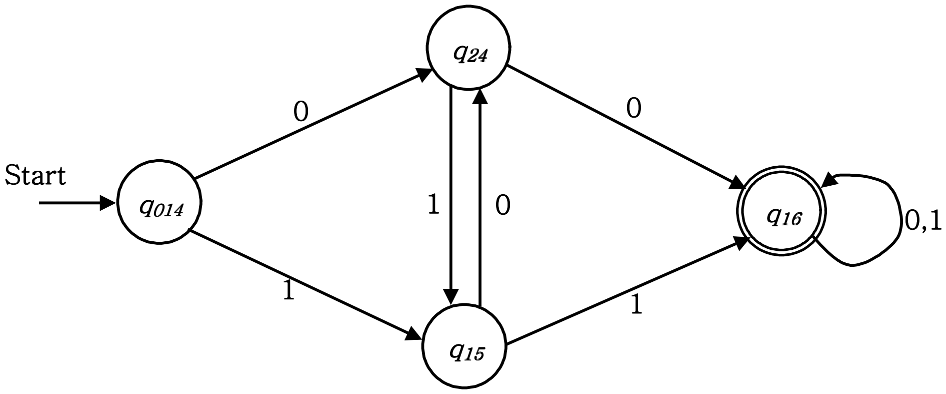
(2) (1)에서 구성한 ε-NFA를 DFA로 바꾸시오.(hint : subset construction을 이용하시오.) <10점>

<solution>



(3) (2)에서 구성한 DFA를 minimize 하시오. (hint : table-filling algorithm을 이용하시오.) <10점>

<solution>



▲ Section D : Properties of Regular Languages

5. 다음 물음에 답하시오.

(1) L = { $w \in \{0,1\}^*$ | w의 뒤에서 k번째 문자가 0이다.} 일 때, L을 받아들이는 DFA의 state 개수는 2^k개 이상임을 증명하시오. (단, 비둘기집 원리와 귀류법을 이용하시오) <10점>

<solution>

State 개수가 2^k보다 작은 DFA인 M이 $L_{(1)}$를 받아들인다고 하자.
시작 state를 q_0라고 할때,
길이가 k인 2^k개의 string w_1 ... w_{2^k} 에 대해, 비둘기집 원리에 의해,

$$\delta(q_0, w_a) = \delta(q_0, w_b) \quad \text{-----} \ (*)$$

를 만족하는 서로 다른 w_a, w_b가 w_1 ... w_{2^k} 중에 존재한다.

w_a와 w_b를 처음부터 읽어서 처음으로 서로 다른 symbol이 l번째에 나온다고 하자.
그러면 $w_{a}1^{(l-1)}$과 $w_{b}1^{(l-1)}$은 뒤에서 k번째에 서로 다른 문자를 가지게 된다.
그래서 $w_{a}1^{(l-1)}$과 $w_{b}1^{(l-1)}$은 하나는 final, 다른 하나는 non-final로 가야되나,
(*)에 의해서,
$\delta(q_0, w_{a}1^{(l-1)}) = \delta(q_0, w_{b}1^{(l-1)})$ 이 만족하게 되어 모순.

고로 $L_{(1)}$을 받아들이려면 적어도 2^k개의 state가 필요하다.

(2) *Pumping lemma*의 역을 증명 또는 반증하시오

<*solution*>

반례:

L = { $0^k.n \in \{0,1\}^*$ | $k \geq 1$, $w \in 1^*$, $|w|$ 는 소수 }

n을 1로 잡으면, 모든 L안의 w에 대해, L안의 w는 $|w| \geq 1$을 만족.

$w = xyz$, $x=\varepsilon$, $y=0$, z를 나머지로 잡으면,
$x(y^k)z$는 모두 L안의 문자열이 된다.

고로 *Pumping lemma*의 결론부를 만족한다.

Regular language에서 특정 문자를 ε으로 치환해도 여전히 regular language가 되어야 함을 알고 있다.
그러나 L에서 0을 모두 지우면 L은 regular language가 아니다.
고로 원래의 L도 regular language가 아니다.

그러므로, *Pumping lemma*의 역은 일반적으로 참이 아니다.

6. 다음 각각의 statement에서, 모든 language L, L_1, L_2에 대해 다음이 성립하는지의 여부를 판단하여라. 성립한다면 증명을 보이고, 그렇지 않다면 반증하시오.

(1) $L_1 \cup L_2$가 regular language이면, L_1, L_2 중 적어도 하나는 regular language이다. <10점>

<solution>

False.

L_1 = { $w \in \{a,b\}^*$ | w에서 a의 개수가 b의 개수보다 많거나 같다. }

L_2 = { $w \in \{a,b\}^*$ | w에서 b의 개수가 a의 개수보다 많거나 같다. }

합치면 전체, 고로 regular지만, 각자는 모두 non-regular

(2) $L_1 \cup L_2$가 regular language이고 L_1, L_2 중 하나가 유한하면, 나머지 하나는 regular language이다. <10점>

<solution>

True.

It can be obtained by starting from the set identity.

L_2 = ((L_1 ∪ L_2) ∩ complement(L_1)) ∪ (L_1 ∩ L_2)

The key observation is that since L_1 is finite, $L_1 \cap L_2$ is finite and therefore regular for all L_2. The rest then follows easily from the known closures under ∪ and complementation.

7. 다음에 제시된 각각의 language에 대해 regular language 여부를 판단하고, 이를 증명 또는 반증하시오.

(1) $L_1 = \{ xwx^R \mid x, w \in \{0,1\}^+ \}$ (hint : L_1은 regular language이다.)<10점>

<solution>

Regular.

$L_1 = 0(0+1)^+0 + 1(0+1)^+1.$

x and w cannot be null. x must begin with the symbol with which x^R ends.

All and only strings of length 3 or more that begin and end with the same symbol are in L_1.

The set of such strings can be represented by the above regular expression.

(2) $L_2 = \{ w \in \{0,1\}^* \mid w = x00y,\ |x| = |y| \text{ for } x, y \in \Sigma^* \}$ <10점>

<solution>

Let us first simplify L_2 by intersecting it with 1^*001^*.

$L_2' = L_2 \cap 1^*001^* = \{ 1^n001^n \mid n \geq 0 \}$

Assume that L_2' is regular. Assume that the pumping length is p.

We pick $1^p001^p \in L_2'$ and $|1^p001^p| \geq p$ as the string $w = xyz$ that we will pump.

By the requirement that $|xy| \leq p$, we know that y must consist only of (a non-zero number of) 1's. However, this string cannot be pumped - adding or removing a non-zero number of 1's from the initial p 1's leads to a string that is not in L_2.

1^p001^p is a string in L_2' which is not pumpable. Hence, L_2' cannot be regular. But if L_2' is not regular, then since the regular languages are closed under intersection, neither is L_2

(3) L_3 = { $w \in \{0,1\}^*$ | w에서 01과 10의 개수가 같다. } <10점>

<solution>

Regular.

DFA를 그릴수 있다.

시작은 accept.

start에서 0, 1 각기 다른 state로. 이것도 accept.

위 두 state에서 0,1로 돌다가 1,0이 나오면 각기 01이 하나 많음을 나타내는 state

10이 하나 많음을 나타내는 state로 이동.

거기서 1,0으로 돌다가 각기 0을 보고 start가 0으로 간곳으로.

1을 보고 start가 1을 통해 간곳으로.

(4) L_4 = { xx^Rw | $x, w \in \{0,1\}^+$ } <15점>

<solution>

Not Regular.

To see why L_4 cannot be regular, let us try to list some of the strings that L_4 contains :

$00(0+1)^+$, $11(0+1)^+$

what about strings in $01(0+1)^+$, and in $10(0+1)^+$?

Clearly $0110(0+1)^+$ and $1001(0+1)^+$ are in L_4.

But then again what about $0100(0+1)^+$, $0101(0+1)^+$, and $0111(0+1)^+$? For example, suppose you have seen $(01)^n$ so far. What information do you need to check if what is to come is in L_4? You need to remember that you have seen n 01's - since n is unbounded, what you need is unbounded memory. This is not something that a FA can do.

A more formal pumping lemma proof follows.

L_4 is not regular. Assume L_4 is regular. Trying to contradict this by pumping L_4 directly does not work. Therefore, we first intersect L_4 with the regular language $L((10)^*(01)^*)$.

This gives us the language

$L_4' = \{ (10)^m(01)^{m+k} \mid m,k>0 \}$.

Let the pumping length be p.

Pick the string to be pumped as $w = (10)^p(01)^{p+n}$.

Now y, which has to lie in the first p symbols of w, can be (i) a string that begins and ends with 0, (ii) a string that begins and ends with 1, (iii) a string that begins with 0 and ends with 1 or (iv) a string that begins with 1 and ends with 0.

If y is of form (i -iv), pumping w even once produces a string not in L_4'. If y is of form (iv), then pumping w $k+1$ times produces a string with more 10's than 01's i.e. a string not in L_4'. Hence, w is not pumpable and consequently, L_4' cannot be regular. But if L_4' is not regular, then neither is L4 due to the closure of the regular languages under intersection.

▲ Section E : Context-Free Grammars and Languages

8. Context-free grammar 대한 다음 물음에 답하시오.

(1) Grammar G = ({S},{a,b},S,P) 는 아래와 같은 production rule을 가진다.

$$S \rightarrow aSa$$
$$S \rightarrow bSb$$
$$S \rightarrow \varepsilon$$

이 language의 complement를 생성(generate)하는 context-free grammar를 고안하시오. <15점>

<solution>

It is normally not possible to use a grammar for L directly to get a grammar for complement of L, so we need another, hopefully recursive description for complement of L.

This is a little hard to see here. One obvious subset of complement of L contains the strings of odd length, but this is not all. Suppose we have an even length string that is not of the form ww^R. Working from the center to the left and to the right simultaneously, compare corresponding symbols.

While some part around the center can be of the form ww^R, at some point we get an a on the left and a b in the corresponding place on the right or vice versa. The string must therefore be of the form $uaww^Rbv$ or $ubww^Rav$ with $|u| = |v|$.

Once we see this, we can construct grammars for these types of strings. One solution is

$$S \rightarrow ASA \mid B$$
$$A \rightarrow a \mid b$$
$$B \rightarrow bCa \mid aCb$$
$$C \rightarrow aCa \mid bCb \mid \lambda$$

The first two productions generate the u and v, the third the two disagreeing symbols, and the last the leftmost palindrome.

(2) 다음 CFG G에 대해 L(G)안의 문자열들은 *ba*를 부분 문자열로 가지고 있지 않음을 induction을 이용하여 보이시오. <10점>

$$S \rightarrow aS \mid bT \mid a$$
$$T \rightarrow bT \mid b$$

<solution>

T는 b로만 이뤄진 길이 1이상의 문자열이다.
S에 대해 S가 *ba*를 부분 문자열을 가지고 있지 않음을 structural induction으로 보이자.

case *a*S)
induction가정에 의해 S가 부분 문자열로 *ba*를 가지고 있지 않고
S는 *a*또는 *b*로 시작하는데 각기 부분 문자열로 *aa*와 *ab*만을 만들어 낸다.

case *b*T)
*b*로만 이뤄져 있다. 당연히 *ba*를 가지고 있지 않다.

case *a*)
*ab*를 부분 문자열로 가질 수 없다.

∴ S로 만들어지는 문자열들은 *ba*를 부분 문자열로 가지고 있지 않다.

9. Grammar G가 어떤 언어 L을 생성한다는 것을 증명하기 위해서는 반드시 L(G)=L임을 보여야 한다. 다시 말해서, 먼저 G로 부터 생성되는 모든 sentence가 L에 속한다는 것을 보인 다음, 역으로 L에 속하는 모든 string이 G에 의해서 생성될 수 있다는 것을 증명해야 한다는 것이다.

이와 같은 사실을 바탕으로, 다음의 문법 G가 L을 생성한다는 것을 induction을 이용하여 증명하시오. <15점>

G=({S}, {(,)}, {S→(S)S | ε}, S) ⇔ L={w | w는 균형을 이룬 괄호로 구성되어 있다.}

(즉, L={ε, (), ()(), (()(())()), ...})

<solution>

(⇒) G가 생성하는 언어가 L임을 보이기 위하여, S로 부터 유도(derivation)되는 모든 sentence는 균형을 이루는 괄호로 되어있다는 것을 증명해야 한다.

S로 부터의 유도(derivation) 횟수 k에 관하여,

ⅰ) k=1일 때, S⇒ε. 여기서 ε은 확실히 균형을 이루기 때문에 성립한다.

ⅱ) k<n인 경우, S로부터 유도(derivation)되는 모든 문장이 균형을 이룬다고 가정하자.

k=n인 경우의 유도(derivation) 과정을 살펴보면,

$$S \Rightarrow (S)S \Rightarrow^{m} (x)y,\ x,y \in \{(,)\}^*,\ m=n-1$$

의 유도(derivation) 과정의 형태를 이룬다. 여기서, terminal string x와 y는 S로부터 n단계 이내에 유도(derivation)되기 때문에 가정에 의해 균형을 이루는 괄호로 구성된다. 그러므로, $(x)y$는 균형을 이루는 괄호로 구성되어 있다. 따라서, S로부터 유도(derivation)되는 모든 sentence는 균형을 이루는 괄호로 구성되어 있다.

(⇐) L을 생성하는 문법이 G임을 보이기 위하여, 균형을 이루는 괄호로 구성된 모든 string이 S로부터 유도(derivation)될 수 있다는 것을 증명해야 한다.

L 안의 모든 sentence는 길이가 짝수다.

ⅰ) $|w|$=0일 때, S ⇒ ε이기 때문에 성립한다.

ⅱ) $|w|$<2n인 경우, 모든 균형을 이룬 괄호로 구성된 string이 S로부터 유도될 수 있다고 가정하자.

$|w|$=2n인 경우, w의 prefix에서 균형을 이룬 가장 짧은 스트링을 (x)라 하자. 그러면 w=(x)y 형태로 표현된다. 여기서 w가 균형을 이루기 때문에 역시 x와 y도 균형을 이룬 괄호로 구성되어 있다. 그런데 x와 y의 길이가 2n보다 작으므로 귀납적 가설에 의해 S로부터 유도될 수 있다. 그러므로, 다음과 같은 유도(derivation) 과정이 존재한다.

$$S \Rightarrow (S)S \Rightarrow^{*} (x)S \Rightarrow^{*} (x)y = w$$

따라서, w도 S로부터 유도(derivation)될 수 있다.

CS322 2006 중간고사
일시: 10/24 오후 4:30
장소: 1501 공동강의실

2006년 10월 24일

전공	
학번	
이름	

문제번호	점수
1	
2	
3	
4	
5	
6	
7	
8	
9	
총점	

Problem 1 (15 pt) 집합론에서 함수는, 특정 조건을 만족하는 "relation의 일종"으로 정의된다. 전산학에서는 이러한 함수를 입력 따라 정해진 출력으로 종료하는 "유한하게 표현할수 있는 알고리즘"으로 구현해서 이용한다.

여기서 질문: 집합론에서의 어떤 함수 $f : \mathbb{N} \to \mathbb{N}$이 주어졌을때 이를 구현한 알고리즘이 언제나 있을까? 증명 또는 반증하시오.

Solution 언제나 존재하는 것은 아니다. 왜냐하면 집합론에서의 함수의 갯수가 알고리즘의 갯수보다 많을수 있기 때문이다.

$$f : \mathbb{N} \to \{0, 1\}$$

를 생각해보자. 전통 집합론에서의 함수는 $2^{\mathbb{N}}$개가 있으나, 알고리즘은 그 표현이 유한해야한다는 조건때문에 그 가짓수가 countably infinite을 넘지 못한다. 고로 대응되는 알고리즘을 가지지 못하는 함수가 존재한다.

Problem 2 (15 pt) (0+1)*01을 받아들이는 minimal DFA를 구성하고, 그 DFA의 simultaneous equation을 풀어서 그 DFA가 받아들이는 언어가 (0+1)*01임을 보이시오. (즉, simultaneous equation을 풀어서 regular expression $(0+1)^*01$을 만들어내는 과정을 보이시오.)

Solution 우선 minimized DFA는 아래의 식을 만들어 내는 DFA다. final state가 아닌 두개의 state A, B가 string 1로 구분 가능하므로, 이 DFA는 minimal이다.

$$A = 0B + 1A$$
$$B = 1C + 0B$$
$$C = 0B + 1A + \epsilon$$

$C = A + \epsilon$, 그러면 $B = 1A + 0B + 1 = A + 1$. 이걸 A의 우변에 대입하면 $A = 0A + 01 + 1A = (0+1)A + 01$ 이므로, $A = (0+1)^*01$이 된다.

Problem 3 (7+8 pt) 다음이 regular 인지 아닌지 판별하시오. (regular이면 R.E.나 automaton제시, 아니면 pumping lemma 등으로 증명)

1. $\{xy \in \{0,1\}^* \mid$ x와 y는 길이가 3인 common substring을 가지고 있다.$\}$

   Solution regular. $(0{+}1)^*(000)(0{+}1)^*(000)(0{+}1)^* + \cdots + (0{+}1)^*(111)(0+1)^*(111)(0+1)^*$

2. $\{0^{(3n+1)^2} \mid n \geq 0\}$

   Solution non regular. 주어진 n에 대해 그 길이보다 큰 언어안의 어떤 string을 x라 하자. 위 언어는 길이가 k번째로 긴 string과 $k+1$번째로 긴 string의 간격이 k가 커짐에 따라 증가한다. 고로, x의 substring이 만들어 내는 균일한 간격의 string은 언젠가 언어를 벗어난다.

Problem 4 (5+5 pt) 다음 language에 대한 regular expression을 구하시오

1. $L = \{a^n b^m \mid n \geq 1, m \geq 1, mn \geq 3\}$

   Solution n과 m은 1이상, 두 수의 곱은 3이상이므로, $aaaa^*bb^*+aaa^*bbb^*+aa^*bbbb^*$

2. $L = \{a^n b^m \mid m + n \text{ is even.}\}$

   Solution n과 m이 모두 짝수인 경우와 홀수인 경우로 나누어서 생각하여 답을 구할수 있다. $(aa)^*(bb)^* + a(aa)^*b(bb)^*$

Problem 5 (20 pt) L을 regular language라고 하자. 이때 L'을

$$L' = \{x \mid xy \in L \text{ and } |y| = |x|\}$$

와 같이 정의하자. L'이 regular language임을 보이시오. (automaton or R.E.를 제시하시오.)

Solution L의 DFA가 $(Q, \Sigma, \delta, q_s, F)$라 하자. L'을 받아들이는 ϵ-NFA는 $(Q \times Q \cup \{S\}, \Sigma, \delta', S, \{(q,q) \mid q \in Q\})$인데 여기서 δ'는,

$\delta'(S, \epsilon) = \{(q_s, q_f) \mid q_f \in F\}$
$\delta'((q_1, q_2), a) = \{(q_1', q_2') \mid q_1' = \delta(q_1, a) \text{ and } q_2' \in \delta^{-1}(q_2, c) \text{ for some } c \in \Sigma\}$

여기서 δ^{-1}는, $\delta^{-1}(q, a) \in q'$ iff $\delta(q', a) \in q$를 만족하는 함수이다.

Problem 6 (10+10 pt) 다음 언어를 받아들이는 CFG를 만드시오.

1. $L = \{a^m b^n c^p d^q \mid m + n = p + q \text{ where } m, n, p, q \in \mathbb{N}\}$

Solution

$$
\begin{aligned}
S &\rightarrow aSd \mid A \mid B \\
A &\rightarrow aAc \mid C \\
B &\rightarrow bBd \mid C \\
C &\rightarrow bCc \mid \epsilon
\end{aligned}
$$

2. $L = \{w \in \{a, b\}^* \mid w$의 길이가 홀수이고 w의 처음, 중간, 마지막 symbol이 같다.$\}$

Solution

$$
\begin{aligned}
S &\rightarrow aAa \mid bBb \mid a \mid b \\
A &\rightarrow aAa \mid aAb \mid bAa \mid bAb \mid a \\
B &\rightarrow aBa \mid aBb \mid bBa \mid bBb \mid b
\end{aligned}
$$

Problem 7 (15 pt) R_1과 R_2 두개의 regular expression이 주어졌을때, $L(R_1) \subseteq L(R_2)$인지를 알아보는 방법을 제시하시오.

Solution $L(R_1) \cap L(R_2) = L(R_1) \Leftrightarrow L(R_1) \subseteq L(R_2)$이므로 $R' = R_1 \cap R_2$에 해당하는 DFA를 교과서 Theorem 4.8을 이용해 만들고, R_1에 해당하는 DFA를 만들어 minimize한 두 DFA가 state로 rename을 해서 동일해질수 있으면 그리고 꼭 그경우에만 $L(R_1) \subseteq L(R_2)$가 성립한다.

Problem 8 (5+10+5 pt) n은 양의 정수이다.

$$L_n = \{xx^R \mid x \in \{0,1\}^n\}$$

1. L_n을 받아들이는 DFA를 만드시오.

   Solution state는 S, $S0$, $S1$, $S00$, $S01$, $S10$, $S11$, …, $S1^n$와, F, $F0$, $F1$, $F00$, $F01$, $F10$, $F11$, …, $F1^{n-1}$. δ는,

   $$\begin{aligned}&\delta(Sx, a) = Sxa, \text{ for } |x| \le n-1\\ &\delta(Sxa, a) = Fx, \text{ for } |xa| = n\\ &\delta(Fxa, a) = Fx, \text{ for } |xa| \le n-1\end{aligned}$$

   시작은 S, final은 F 하나.

2. minimized DFA를 만들고, 그 DFA가 정말 minimized DFA임을 증명하시오.

   Solution 앞문제의 답으로 제시한 DFA가 minimized DFA이다. 위 DFA는 구조상 각 state의 S로부터의 거리가 유일하게 정해진다. 그리고 그에 따라 F로부터의 거리가 $2n-$(S로 부터의 거리)로 정해진다. 또한 F에 더 가깝지 않은 경우 S에서 오는 string이 유일하며, S에 더 가깝지 않을 경우 F까지가는 string이 유일하다. 이제 이 DFA에서 구분 불가능한 state 두개를 고르자. 이들이 S로부터 거리가 다르면 하나가 F로 이르는 string을 가지고 다른 하나는 F에 절대 이르지 못한다. 고로 S로 부터의 거리는 같아야 한다.

   - F에 더 가까운경우, 이들이 F에 이르는 길은 각기 유일한 경로 하나뿐이다. 두개가 indistinguishable하므로 final에 이르게 하는 string이 같아야되고 결국 둘은 같은 state일수밖에 없다.
   - F에 더 가깝지 않은 경우, S에서 이들에 이르게 하는 두개의 string을 x_a, x_b라 하자. 길이는 l. 첫번째 state는 $0^{2(n-l)}x_a^R$에 의해 F에 도달한다. indistinguishable 성질때문에 두번째 state도 $0^{2(n-l)}x_a^R$에 의해 F에 도달해야한다. 여기서 $x_b0^{2(n-l)}x_a^R$는 palindrom이어야하므로, 결국 $x_a = x_b$, 즉 두 state는 동일하다.

   즉, 위 DFA에서 어떤 state와 indistinguishable state는 자기 자신뿐이므로 위 DFA는 minimal하다.

3. minimized DFA의 state 갯수를 n에 대한 함수꼴로 나타내시오. (dead state 포함할것)

   Solution state 갯수는 $3 \cdot 2^n - 1$ or $(2^{n+1} - 1) + (2^n - 1) + 1$.

Problem 9 (7+8+15 pt) 열어볼수 없는 블랙박스 속에 어떤 DFA가 들어있다.

- 블랙박스 외부에는 DFA symbol 입력 버튼과 enter키가 있다.
- 블랙박스의 외부에 Yes/No 표시창이 있어 string을 입력하고 enter를 치면 그 DFA가 입력된 string을 accept하는지를 표시창을 통해 알수있다.
- enter를 치고 나면 입력란은 clear된다.
- 그 DFA의 state 갯수 n은 누군가 알려주어서 알수 있었다.

1. 이 DFA가 나타내는 언어가 $\emptyset$인지 아닌지 위 기계를 유한번만 사용하고 알아낼수 있는가? 증명 또는 반증하시오.

   Solution 이 DFA가 받아들이는 string이 적어도 하나 있고, 길이가 모두 n이상이라고 가정해보자. 이 string은 DFA가 받아들이는 string 중 가장 짧은 길이를 가진 string중 하나를 x라고 하자. DFA가 처리하는 과정에서 적어도 한번은 같은 state를 방문하게 된다($|x| \geq n$이므로). 그러면 x에서 loop를 만드는 부분을 삭제한 string도 DFA의 string이다. x가 최소 길이라는 것에 모순. 고로 DFA가 받아들이는 string이 하나라도 있을때 가장 짧은 string은 길이가 n보다 작다.

   그러므로 우리는 길이가 n보다 작은 string을 모두 입력해 봄으로서 DFA의 언어가 $\emptyset$인지 아닌지 판단할수 있다.

2. 이 DFA가 나타내는 언어가 무한한 크기의 집합인지 위 기계를 유한번만 사용하고 알아낼수 있는가? 증명 또는 반증하시오.

   Solution 비슷하게 DFA가 받아들이는 string이 무한히 많다고 해보자. 받아들이는 string중 DFA에서 cycle을 만들어 내는 것이 적어도 하나 있다(안그러면 유한). 이런 cycle을 가지고 있는 string중 가장 짧은 것을 x라 하자. x내에는 cycle이 딱 한개, 그리고 그 cycle은 딱 한번 반복되고 중복 방문하는 state는 하나뿐이다. 고로 x에서 cycle을 제거하면 길이가 n보다 작게 되며, cycle을 한번 추가할때마다 그 길이가 $k < n$만큼 증가한다. 고로 DFA가 받아들이는 길이가 n이상이고 $2n$보다 작은 string이 반드시 있다.

   그러므로 우리는 길이가 n보다 크고 $2n$이하인 string을 모두 입력해 봄으로써 DFA의 언어가 무한인지 아닌지 알수 있다.

3. 블랙박스 내의 DFA가 어떤 언어를 나타내는지 위 기계를 유한번만 사용하고 알아낼수 있는가? 증명 또는 반증하시오.

Solution 가능하다.

Claim: DFA에서 두개의 구분가능한 state를 구분시키는 string중 가장 짧은 것은 길이가 $n-2$이하다. 즉, 길이 $n-2$ 이하의 문자열에 대해 모두 같은 결과면 두 state는 equivalent. (숙제에서 증명)

우리는 길이 $2n-2$이하의 모든 string을 블랙박스에 입력하여 결과를 얻어내자. 그러면 우선 그 길이 이하의 모든 string을 accept하는지 안하는지 알수 있다. 그리고, 이 결과를 통해 우리는 $2n-1$이상 길이의 어떤 string을 accept하는지 안하는지 알아내는 방법은 아래와 같다.

길이가 $2n-2$보다 큰 어떤 string x가 주어졌다. 이 string의 앞 n개의 string을 처리하면 중복 방문하는 state가 생긴다. 고로 $\delta(S,y) \equiv \delta(S,z)$를 만족하는 길이 n이하인 서로다른 x의 prefix y,z가 반드시 존재한다. w.l.o.g. $|y| < |z|$. 이를 만족하는 y,z는 길이 $n-2$ 길이 이하의 string에 대해 모두 같은 결과를 주는 것을 찾으면 되는데, 이는 이미 알고있는 길이 $2n-2$ 이하의 string에 대한 결과에서 알아낼수 있다. $y(z^{-1}x)$는 x와 정확히 같은 상태를 주며 길이가 x보다 strict하게 작다. 이와 같은 작업을 길이가 $2n-2$보다 큰 경우에는 언제나 적용할수 있으므로 결국 처음에 주어진 string의 acceptance를 길이 $2n-2$ 이하인 어떤 string의 acceptance로 바꿀수 있다.

위와 같은 방법과 길이 $2n-2$이하의 string에 대한 결과를 통해 DFA가 나타내는 언어를 알아낼수 있다.

Korea Advanced Institute of Science and Technology
Department of Electrical Engineering and Computer Science
Division of Computer Science

CS322 : Formal Language and Automata

Midterm Fall 2007

Instructor : Prof. Choe, Kwang-Moo

▲ Instructions

- **Open Book, Open Note**
  노트북, 휴대폰 등의 통신 장비를 제외한 paper material을 참고하는 것이 허용됩니다.
  하지만, 시험 중 주위 응시자와 material을 share하는 것은 부정행위로 간주되니 주의하시기 바랍니다.
- **Please write legibly.**
  가능한 한 깔끔하게 답안을 작성해 주시기 바랍니다. 채점자가 알아볼 수 없는 답안에 대해서는 점수를 줄 수 없습니다.
- **11 problems**
  문제는 총 11 문제로 구성되어 있습니다. 문제 풀이에 들어가기 전에, 파본을 확인하면서 각 장의 상단에 자신의 학번과 성명을 기입하여 주시기 바랍니다.
- **120 minutes**
  시험 시간은 2시간이 주어집니다. (11:00 ~ 13:00)

▲ **Student ID** : ____________________

▲ **Name** : ____________________

▲ **Signature** : ____________________

▲ **Grading Result**

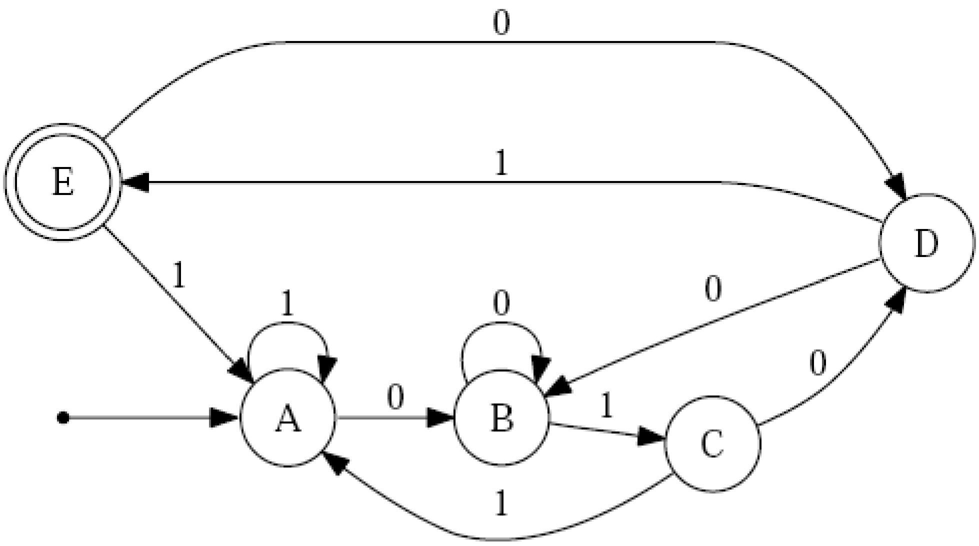
1. 문자열의 집합은 countable 하다. 상태집합이 Σ^*의 부분 집합이고, alphabet이 Σ인 모든 DFA에 대한 집합을 Ψ라 부르자. Ψ가 countable함을 보여라.(10점)

(SOLUTION)

$|\Sigma|=k$이라 하자. Σ를 $\{x_1, x_2, \cdots, x_k\}$로 표기하자.
$|Q|=n$이고 $|F|=m$인 임의의 DFA $(Q, \Sigma, \delta, q_s, F)$를 다음과 같이 encoding하자.
$q_1\$q_2\$q_3\cdots q_n \mid q'_{1,1}\$q'_{1,2}\$\cdots q'_{1,k}\$q'_{2,1}\$q'_{2,2}\$\cdots q'_{2,k}\$ \cdots q'_{n,1}\$q'_{n,2}\$\cdots q'_{n,k} \mid q_s \mid f_1\$f_2\$\cdots f_m$
여기에서 $(q_1,q_2,q_3\cdots q_n)$은 Q의 원소들을 정렬한 순열이다. $q'_{i,j}=\delta(q_i,x_k)$이다. $(f_1, f_2 \cdots f_m)$은 F의 원소들을 정렬한 순열이다. 이는 $T=(\Sigma\cup\{\$,|\})*$의 부분 집합이다. T는 문자열의 집합이므로 countable.
$f\subseteq T\times N$인 one-to-one 함수 f가 존재. $g=\{(x, y)\in f|\ x\in$DFA의 encoding$\}$이면, g도 one-to-one인 함수가 되므로 Ψ도 countable.

2. (0 + 1)*0101 을 받아들이는 minimal DFA를 구성하고, 그 DFA의 Simultaneous equation 을 풀어서 그 DFA가 받아들이는 언어가 정확히 (0 + 1)*0101 임을 보이시오. (10점)

<u>(SOLUTION)</u>



A = 1A + 0B

B = 0B + 1C

C = 1A + 0D

D = 1E + 0B

E = 1A + 0D + ε

E = 1A + 0D + ε → E = C + ε (C = 1A + 0D)

D = 1E + 0B → D = 0B + 1C + 1

D = 0B + 1C + 1 → D = B + 1 (B = 0B + 1C)

C = 1A + 0D → C = 1A + 0B + 01 (D = B + 1)

C = 1A + 0B + 01 → C = A + 01 (A = 1A + 0B)

B = 0B + 1C → B = 0B + 1A + 11 → B = A + 101

A = 1A + 0B = 1A + 0A + 0101 = (1+0)*0101

3. 다음의 언어 L을 받아들이는(accept) state의 개수를 구하라: $L=\{w_1w_2w_3w_4..w_n \in \{0,1\}^* \mid n \geq 0, 1 \leq i \leq n, |w_i|=32$ and w_i내의 0의 개수는 짝수$\}$. (단, dead state 포함) (10점)

<u>(SOLUTION)</u>

DFA X=(Q, Σ, δ, (0,e), F) where

Q={0,1,⋯,31}x{o,e}, Σ={0,1}, F={(0,e)} and

δ={(((q,p),1), (q+1,p))|q∈{1,⋯,30}, p∈{o,e}}∪{(((31,p),1), (0,p))|p∈{o,e}}∪{ (((0,e),1), (1,e)) }

∪{(((q,o),0), (q+1,e))|q∈{1,⋯,30}}∪{(((31,o),0), (0,e)) }

∪{(((q,e),0), (q+1,o))|q∈{0,⋯,30}}∪{(((31,e),0), (0,o)) }

∪{ (((0,o),x), (0,o)) | x∈Σ}.

L((Q,Σ,δ,(0,o), F))은 공집합이지만, 다른 상태에서 시작한 언어는 공집합이 아니다.

q<q'이고 p, p'∈{o,e}일 때 (q,p)에서 시작한 언어와 (q', p') 시작한 언어는 길이가 같을 수 없다.

q∈{0,1,⋯,31}일 때 (q, e)에서 시작한 언어와 (q, o)에서 시작한 언어는 32-q길이의 접두사에서 0의 개수가 같을 수 없다.

결국 모두 distinguish하다. 결국 상태 개수는 |Q|=32x2=64이다.

4. Σ = {0, 1} 에서 정의된 다음의 언어들에 대해, 각 언어를 표현하는 적절한 Regular expression을 제시하시오. (10점)

(1) 01 로 끝나지 않는 모든 string들
(2) 짝수개의 0을 포함하는 모든 string들
(3) 0 과 1 을 각각 적어도 하나 이상은 포함하는 모든 string들

<u>(SOLUTION)</u>
(1) $(\varepsilon + 0 + 1) + (0 + 1)^*(00 + 10 + 11)$

$(\varepsilon + 0 + 1)$ 은 String의 길이가 1보다 작을 때를 나타내며, $(0 + 1)^*(00 + 10 + 11)$ 은 string의 길이가 2와 같거나 클 때의 String을 표현한다.

(2) $1^* + (1^*01^*01^*)^*$

(3) $(0 + 1)^*0(0 + 1)^*1(0 + 1)^* + (0 + 1)^*1(0 + 1)^*0(0 + 1)^*$

5. Language를 기술하는데 있어서 language에 속하는 하나의 문장에 대해 이를 표현하는 방법이 두 가지 이상이면 그러한 기술을 ambiguous하다고 말한다. Regular expression의 경우 예를 들면 다음과 같은 경우 ambiguous 하다. (10점)

- $0 \cup 0$: 0 을 표현하는 방법이 두 가지임
- $(0 \cup 01)(\varepsilon \cup 1)$: 01 을 표현하는 방법이 두 가지
- $(0 \cup 000)^*$: $0^{n \ge 2}$ 을 표현하는 방법이 최소한 두 가지 이상
- ε^* : ε 을 표현하는 방법이 countably infinite 함

Regular expression 이 unambiguous 하다는 것의 정의를 regular expression에 대한 structural induction으로 정의하면 다음과 같이 된다.

a) $\varnothing$, ε , a (any symbol) 는 unambiguous 하다.
b) (E) 는 E 가 unambiguous 하면 unambiguous 하다.
c) $E_1 \cup E_2$
d) $E_1 E_2$
e) E^*
f) a)-e)를 만족하지 않는 다른 어떤 것도 unambiguous 하지 않다.

c),d),e)에 대해 위의 정의를 완성하시오. (15점)

<u>(SOLUTION)</u>

c) $E_1 \cup E_2$ 는 E_1, E_2 가 unambiguous 하고 $L(E_1) \cap L(E_2) = \varnothing$ 이면 unambiguous 하다.

d) $E_1 E_2$ 는 E_1 , E_2 가 unambiguous 하고 모든 $x \in L(E_1 E_2)$ 에 대해 $x_1 x_2 = x$, $x_1 \in L(E_1)$, $x_2 \in L(E_2)$ 인 x_1, x_2 가 정확히 하나 존재하면 unambiguous 하다.

e) E^* 는 E 가 unambiguous 하고 모든 $x \in L(E^*)$ 에 대해 정확히 하나의 $n \ge 0$ 이 존재해서 $x_1 \cdots x_n = x$ 인 $x_i \in L(E)$ 의 sequence $(x_1, \cdots, x_n)$ 이 정확히 하나 존재하면 unambiguous 하다.

6. $L = \{a^n b^n | n > 0\}$이 regular language 가 아님을 pigeonhole principle 을 이용하여 증명하라. (10 점)

Show that $L = \{a^n b^n \mid n > 0\}$ is not regular using pigeonhole principle.

Solution Proof by contradiction. Suppose L is regular. There are an infinite number of values of n but M(L) has only a finite number of state. by the pigeonhole principle, there must be distinct values of i and j such that a^i and a^j end in the same state. From this state,

- b^i must end in a final state, because $a^i b^i$ is in L; and
- b^i must end in a nonfinal state, because $a^j b^i$ is not in L.

Since the state reached cannot be both final and nonfinal, we have a contradiction. Thus our assumption, that L is regular, must be incorrenct.

7. 아래 language들이 regular하지 않음을 증명하시오. (10점)

(a) $L_n = \{a^n b^n c^n | n \in N\}$

(b) $L = \{a^i b^j c^k | i \neq j \text{ or } j \neq k\}$

Solve the following problems.

(a) Show that $L_n = \{a^n b^n c^n \mid n \in N\}$ is not regular.

Solution Suppose that L_i is regular. Therefore there must be a DFA with size n, which is the minimal accepting automata.

Pick $z = a^n b^n c^n$

obviously (given $z = uvw$ and $|uv| \leq n$ and $|v| \geq 1$)

$v = a^j$

But, if we pick i to 0, then $uv^0w = a^{n-j}b^n c^n$ which is not in L_i, so L_i is not regular.

(b) Show that $L = \{a^i b^j c^k \mid i \neq j \text{ or } j \neq k\}$ is not regular.

Solution It's obvious that $L^* = a^* b^* c^*$ is regular because we just wrote the regular expression for it. If L is regular $L^* - L$ will produce a regular language by closure property of difference. But since $L^* - L = L_i$, and we know that L_i is not regular, then L cannot be regular.

8. 아래 statement들을 증명하거나 반증하시오. (10점)

(a) $L_1 \cup L_2$가 regular이면, L_1과 L_2 둘 가운데 적어도 하나는 regular이다.

(b) $L_1 \cup L_2$가 regular이고 L_1과 L_2 둘 가운데 하나가 finite하면, 다른 쪽 하나 또한 regular이다.

Prove or disprove the following statements.

(a) If $L_1 \cup L_2$ is regular, then at least one of L_1 and L_2 is regular.

Solution False.

$L_1 = \{w \in \{a, b\}^* \mid$ the number of a's is more than or equal to the number of b's$\}$

$L_2 = \{w \in \{a, b\}^* \mid$ the number of b's is more than or equal to the number of a's$\}$

$L_1 \cup L_2$ is regular, but each of them is non-regular.

(b) If $L_1 \cup L_2$ is regular and one of L_1, L_2 is finite, then the other is also regular.

Solution True.

It can be obtained by starting from the set identity. $L_2 = ((L_1 \cup L_2) \cap L_1^c) \cup (L_1 \cap L_2)$ The key observation is that since L_1 is finite, $L_1 \cap L_2$ is finite and therefore regular for all L_2. The rest then follows easily from the known closures under $\cup$ and complementation.

9. 어떤 equivalent relation R (Σ^* 상에서 정의된)이 Σ^*를 n개의 class로 분할한다고 하자. R은 다음의 조건을 만족한다고 하자: $\forall x, y \in \Sigma^*, \forall a \in \Sigma, [x]_R=[y]_R \rightarrow [xa]_R=[ya]_R$. equivalence class들의 임의의 합집합을 L이라 부르자. 단, equivalence relation의 정의는 다음과 같다. $R \subseteq \Sigma^* x \Sigma^*$에 대하여 $[x]_R=\{y \in \Sigma^* | x R y\}$이다. (10점)

(a) L을 받아들이는 DFA M_L을 만들어라.
(b) 문자열 길이에 대한 수학적 귀납법을 이용하여 $L=L(M_L)$임을 증명하라.

<u>(SOLUTION)</u>

(a) $M_L=(\{[w]_R | w \in \Sigma^*\}, \Sigma, \delta, [\varepsilon]_R, \{[w]_R | w \in L\})$ where $\delta([w]_R,a)=[wa]_R$.

M_L이 DFA임을 보이자. δ가 함수이면 M_L는 DFA이다.

δ가 함수가 아니면, $y \neq z$, $[x]_R=[x']_R$, $\delta([x]_R,a)=y$, $\delta([x']_R,a)=z$인 x, x', a가 있다.

$\delta([x]_R,a)=[xa]_R=y$, $\delta([x']_R,a)=[x'a]_R=z$가 된다.

R의 정의에 의하여 y=z이 되어 모순이다.

즉 M_L이 DFA이다.

(b) $\delta*([\varepsilon]_R,w)=[w]_R$ 임을 증명하자.

Basis: $\delta*(q,\varepsilon)=q$이므로 $\delta*([\varepsilon]_R,\varepsilon)=[\varepsilon]_R$

IH: $|w|=k-1$의 경우 $\delta*([\varepsilon]_R,w)=[w]_R$이 만족

Induction: $|w|=k$이면, $w=w'a$ ($w' \in \Sigma^{k-1}$, $a \in \Sigma$)로 표현가능.

$\delta(\delta*([\varepsilon]_R, w'), a)$

$=\delta([w']_R, a)$ (IH에 의하여)

$=[w'a]_R$ (δ의 정의에 의하여)

10. 다음 Language L의 Context Free Grammar를 구하라. (10점)

1) $L = \{w \in \Sigma* \mid n \geq 0. (w = a^n b^{2n}) \vee (w = a^{3n} b^n)\}$.

2) $L=\{ a^m b^n c^p d^q \in \{a,b,c,d\}* \mid m + n = p + q\}$

<u>(SOLUTION)</u>

1)

$S \rightarrow X \mid Y$ S is the start symbol

$X \rightarrow \varepsilon \mid aXbb$ X generates $a^n b^{2n}$

$Y \rightarrow \varepsilon \mid aaaYb$ Y generates $a^{3n} b^n$

2)

S -> aSd | A | B

A -> aAc | C

B -> bBd | C

C -> bCc | ε

11. 다음 Language L의 complement language $\bar{L}$이 Context Free인지 아닌지 답하고, 그 답을 증명하여라. (10점)

L={ ww| w ∈ Σ* } where Σ = {a, b}

<u>(SOLUTION)</u>

Proof)

$\bar{L} = L' \cup L''$

L' is the set of words of odd length

$L'' = \{uv \mid |u| = |v| \text{ and } u \neq v\}$

Accepted by

S -> AB | BA

A -> xAx | a

B -> xBx | b

x -> a | b

CS 322 2008 중간고사

일시: 10/22 오후 1:00 ~ 3:00

장소: 1501 공동강의실

전공	
학번	
이름	

문제번호	점수
1	
2	
3	
4	
5	
6	
7	
8	
9	
총점	

Problem 1. (10pt, 5pt each)

Let A be a set, and · is a product of two relations on A. (Refer to TP, 01 [보조] Reviews on Discrete Mathmatics)

(1) Show that ($2^{A \times A}$, ·, id_A) is a monoid.

(2) If a monoid has an inverse, it is called as group**.** What is the inverse in the above monoid?

Problem 2. (15pt, 5pt each)

Provide DFAs or NFAs for the following languages over the alphabet ∑ = {0, 1}

(1) (DFA) All string that contain at least two instances of the substring 0110.

(2) (DFA) All strings that do not end with 111.

(3) (NFA) All strings that two 0's separated by a number of positions that is a non-zero multiple of 3. Each position between the two 0's contains an arbitrary symbol from the alphabet. (Example: 1$\underline{0}$111$\underline{0}$, $\underline{0}$101$\underline{0}$1 are accepted, 1010110 is not accepted.)

Problem 3. (15pt)

Let

$$\sum_2 = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$$

Here, $\sum_2$ contains all columns of 0s and 1s of height two. A string of symbols in $\sum_2$ gives two rows of 0s and 1s. Consider each row to be a binary number and let

$$C = \{ w \in \sum_2 \mid \text{the bottom row of } w \text{ is three times the top row} \}$$

For example, $\begin{bmatrix} 0 \\ 0 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \end{bmatrix} \in C$, but $\begin{bmatrix} 0 \\ 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \notin C$.

Provide an NFA that recognizes C^R

Problem 4. (10pt)

Given a language L = {aa, aaa}*, prove that L = { a^n | $n \geq 2$ }

Problem 5. (10pt, 5pt each)

Prove or disprove whether the following equalities between regular expressions are true or not:

(1) $(r_1 r_2 + r_1)^* r_1 r_2 = (r_1 r_1^* r_2)^*$

(2) $(r_1 r_2 + r_1)^* r_1 = r_1 (r_2 r_1 + r_1)^*$

Problem 6. (20pt, 10pt each)

Show that for any regular language L, the languages below are regular

(1) $\mathit{cycle}(L) = \{w \mid w = xy \text{ such that } yx \in L \text{ for some strings } x, y\}$

(2) $\max(L) = \{w \in L \mid \text{ for any } x \in \Sigma^{+},\ wx \notin L\}$

Problem 7. (20pt, 10pt each)

Prove that the languages below are not regular.

(1) $\{1^k s \in \{0,1\}^* \mid s \text{ has at most } k \text{ 1's for } k \geqq 1\}$

(2) $\{1^{k^3} \mid \text{where } k \text{ is positive integer}\}$

Problem 8. (15pt, 5pt each)

For the following grammar: S → S and S | not S | true | false

(1) List all left-most derivations for the string "not true and true".

(2) Draw the parse trees for all of the left-most derivations above.

(3) Rewrite the grammar so that "and" is left associative and has lower precedence than "not".

Problem 9. (20pt, 10pt each)

Give context-free grammars generating the following languages over the alphabet {a,b}

(1) $\{ a^i b^j a^{i+j+k} b^k \mid i, j, k \geqq 0 \}$

(2) $\{ a^{i+j} b^j \mid i \geqq j \}$

CS322 2009 Fall

Midterm Exam.

10/21(Wed), 1:00~3:00 PM

Instructions

1. **No question about a problem during this Exam.** If you think a problem is wrong, then explain why you think so. If you think a problem has something ambiguous or unclear, then make an assumption that you think is necessary and proceed under the assumption. (In these cases, of course, your explanation or assumption must be reasonable.)
2. **Do not skip too much in your answer**. While evaluating your answer, TAs don't try to guess and fill any omission of important steps in your proofs. Especially, any argument such as "This is trivial." is not taken as a proof.
3. **Please write legibly** so that TAs can read your answer and give you appropriate points.
4. **Open book, open notes.** You are allowed to see any paper material during this Exam.
5. **Please turn off any mobile communication devices** such as mobile phones or laptop computers. This may interrupt other students or may be used for cheating.

Student ID	
Name	

Skip this page. TAs' use only

	(1)	(2)	(3)	(4)	Row Sum
1	/10	/10			/20
2	/5	/10			/15
3	/10	/15			/25
4	/10	/15	/10		/35
5	/10	/10	/15	/10	/45
6	/10	/10	/10	/10	/40
7	/10	/10	/10		/30
8	/5	/15	/15		/35

Total	/245

Problem 1. (10 + 10 points)

Consider following languages over the alphabet $\Sigma=\{0,1\}$. Construct an accepting DFA for each.

a) Every 00 is immediately followed by a 1 (for example, the strings ε, 101, 0010, 0010011001 are in the language, but 0001 and 00100 are not).
b) $\{vwv \mid v, w \in \Sigma^* \text{ and } |v|=2\}$

Problem 2. (5 + 10 points)

Consider a DFA $M = (Q, \Sigma, \delta, q_0, \{q_f\})$ which satisfies

$$\forall a \in \Sigma,\ \delta(q_f, a) = \delta(q_0, a)$$

(a) Show that for every non-empty string $w \in \Sigma^*$, it must be the case that $\hat{\delta}(q_f, w) = \hat{\delta}(q_0, w)$.

(b) Let w be a string in the language of M. Prove by induction that for all $k > 0$, $w^k \in L(M)$.

Problem 3. (10 + 15 points)

(a) Let $L \subset \{0,1\}^*$ be the language such that, in every string in L, there exists a pair of two 0's separated by a substring whose length is a non-zero multiple of 5 (for example, 1001110 is not in L, but 10111010 is in L). Construct an NFA for this language.

(b) Prove that any DFA accepting this language must have <u>at least</u> 2^5 states.

Problem 4. (10 + 15 + 10 points)

Decide whether each of the following languages is regular or not. Justify your answer. ($\Sigma = \{0,1\}$, x^R is the reverse of a string x.)

(a) $\{ xx^Rw \mid x, w \in \Sigma^+ \}$
(b) $\{ xwx^R \mid x, w \in \Sigma^+ \}$
(c) $\{ wxx^R \mid x, w \in \Sigma^+, |w|>|x| \}$

Problem 5. (10 + 10 + 15 + 10 points)

Prove or disprove following statements about languages over some alphabet.

(a) If $L_1 \cap L_2$ is regular and is not empty, then at least one of L_1 and L_2 is regular.

(b) If $\bigcup_{i \in N} L_i$ (i.e. countably infinite union of L_i's) is regular, then each of L_i is regular, where N is the set of all natural numbers.

(c) If L^* is regular, then L is regular.

(d) If $L_1 \cup L_2$ is regular and L_1 is finite, then L_2 is regular.

Problem 6. (10 + 10 + 10 + 10 points)

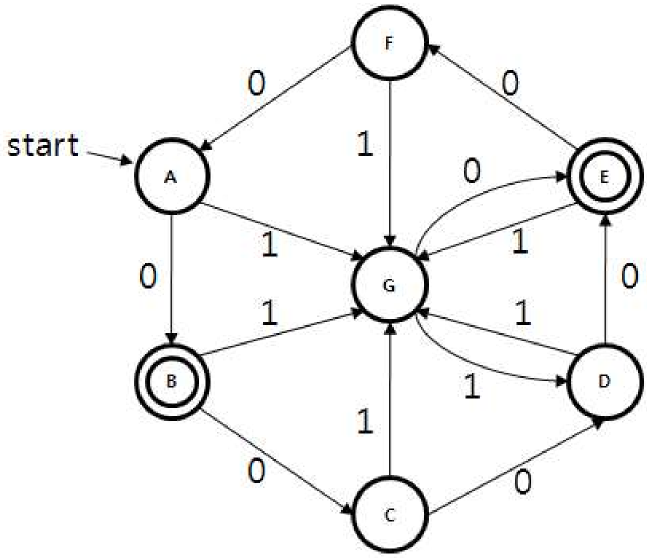
Let L be a language (which is not necessarily regular) over an alphabet Σ and let R_L be the relation on Σ^* induced by L defined as follows

$$\text{(Definition of } x\ R_L\ y) : \forall z \in \Sigma^*, (xz \in L \text{ iff } yz \in L)$$

(a) Prove that R_L is an equivalence relation.
(b) Prove that R_L is right invariant (i.e. for all x, y and w in Σ^*, $x\ R_L\ y$ implies $xw\ R_L\ yw$).
(c) What are the equivalence classes of R_L when $\Sigma = \{0,1\}$ and L is the language generated by 0^*1^*. Verify that your classes partition Σ^*. Are the classes finite or infinite?
(d) What are the equivalence classes of R_L when $\Sigma = \{0,1\}$ and $L = \{0^n1^n \mid n=0,1,2,3, \cdots\}$. Verify that your classes partition Σ^*. Are the classes finite or infinite?

Problem 7. (10 + 10 + 10 points)

Answer the following questions about the DFA below.



(1) Minimize above DFA.
(2) Describe the language accepted by above DFA in plain English.
(3) Derive a regular expression for the language.

Problem 8. (5 + 15 + 15 points)

Consider following language L on the alphabet $\Sigma=\{0,1\}$.

$$L = \{x \mid x \text{ has one more 0's than 1's}\}$$

(a) Prove that L is not regular.
(b) Write a context-free grammar G that generates L. Clarify that your production rules cover all the possible cases that can be derived from the start symbol. Note that this implies $L(G) \supset L$.
(c) Prove L(G) satisfies the condition "x has one more 0's than 1's" (hint: structural induction). Note that this implies $L(G) \subset L$ and, together with the result of (b), also implies $L(G)=L$.

CS322 : Formal Language and Automata
Midterm Exam 2011

Date: 10/20 14:30~16:30

Location: CS1501

Major	
Student ID	
Name	

Problem	Score
1 (15)	
2 (10)	
3 (20)	
4 (10)	
5 (15)	
6 (30)	
Sum (100)	

Problem 1 (15pt)

For a string x, let x^R be the reverse of the string and x^i be the string concatenated i times (e.g., $(abc)^R = cba$ and $(abc)^2 = abcabc$).

(1) Define x^i $(i \geq 0)$, recursively. (5pt)

(sol)

i) basis: $i=0$, $x^i = x^0 = \epsilon$

ii) recursion: $i > 0$, $x^i = x^{i-1} \cdot x = x \cdot x^{i-1}$

(2) Prove by induction that $(x^R)^i = (x^i)^R$. (10pt)

(sol)

i) basis: $i=0$, $(x^R)^0 = \epsilon$, $(x^0)^R = \epsilon^R = \epsilon$, the statement holds

ii) induction: Assume that the statement holds for $i=n$.

$(x^R)^{n+1} = (x^R)^n \cdot x^R$ (by definition of x^i)

$= (x^n)^R \cdot x^R$ (by inductive hypothesis)

$= (x \cdot x^n)^R$ ($a^R b^R = (ba)^R$)

$= (x^{n+1})^R$ (by definition of x^i)

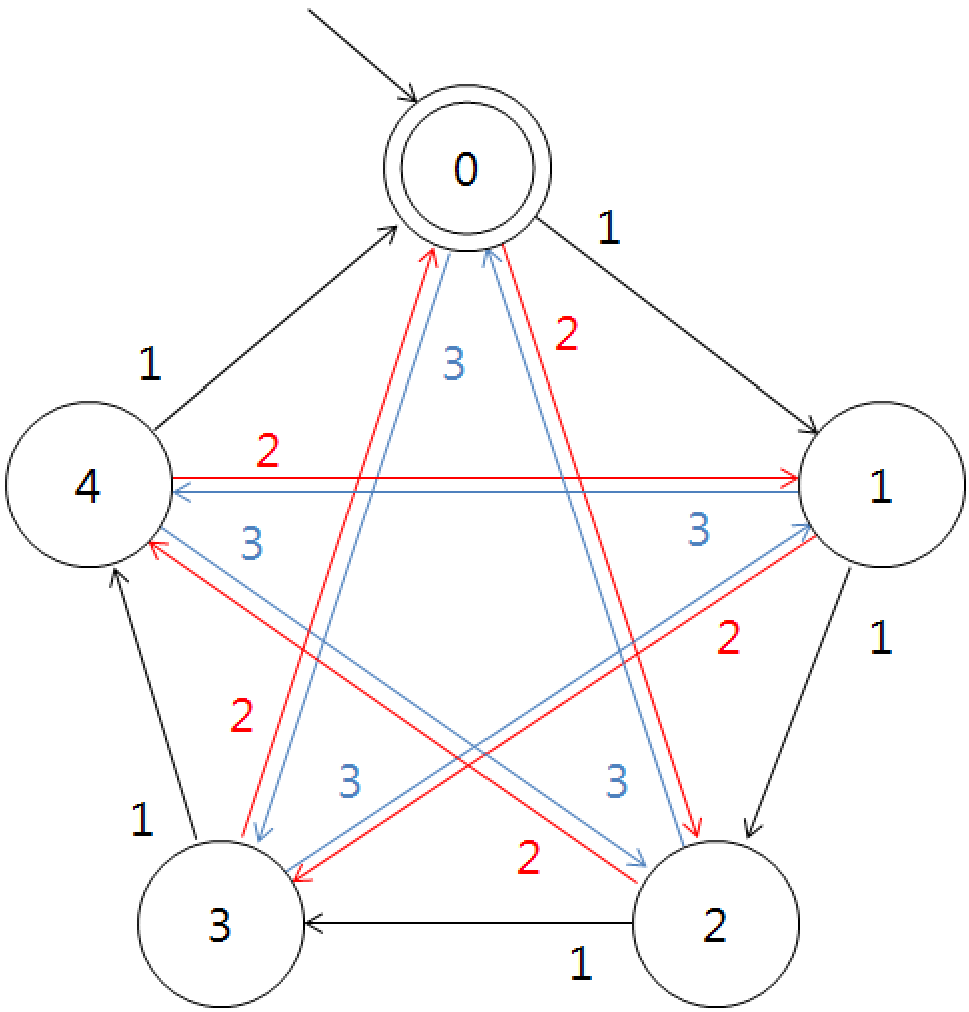
The statement also holds for $i = n+1$.

By mathematical induction, $(x^R)^i = (x^i)^R$.

Problem 2 (10pt)

Draw a DFA accepting language over the alphabet {1, 2, 3} in which the sum of all symbols is divisible by 5.

(sol)



Problem 3 (20pt, 5pt each)

For each of the following languages, prove whether the language is regular or non-regular.

(1) $\{a^i b^j a^j b^i \mid i,\ j \geq 0\}$

(2) $\{a^{i^2} \mid i \in N\}$

(3) $\left\{a^{j-i} \mid \frac{j}{i} = 4\right\}$

(4) $\{a^m b^n a^{m+n} \mid m,\ n \geq 0\}$

(sol)

(1) Non-regular. Call the given Language L_1. Suppose L_1 is regular. Then $L_1 \cap L(b^* a^*)$ is also regular. But this is simply $\{b^n a^n \mid n \geq 0\}$, which we know to be non-regular. Therefore, our assumption was false, and L_1 must be non-regular.

(2) Non-regular. We use the pumping theorem. Given any k > 0, we examine the string a^{k^2}, which has length greater than k. Now, in order for L to be regular, our w should be able to be written as $w = xyz$ with $y \neq \epsilon$, $|xy| \leq k$, and $xy^i z \in L$ for each $i \geq 0$. Since this holds for all $i \geq 0$, it must hold for i = 2. Because $y \neq \epsilon$, $|xy^2 z| > |xyz|$. Also, we know that $|y| \leq k$, so $|xy^2 z| - |xyz| \leq k$. In order to be in L, $xy^2 z$ must have a length that is a square number. But the next largest such string after a^{k^2} is $a^{(k+1)^2}$. This string has length $(k+1)^2 = k^2 + 2k + 1$. So there are no strings in L with lengths between k^2 and $k^2 + 2k + 1$.

But $0 < |xy^2 z| - |xyz| = |xy^2 z| - k^2 < k$, which means that $xy^2 z$ has length between k^2 and $k^2 + k$. Therefore, $xy^2 z$ is not in L, which means that $w = xyz$ could not be pumped. So our assumption that L is regular was false.

(3) Regular. $\frac{j}{i} = 4 \Rightarrow j = 4i$. So $j - i = 4i - i = 3i$. Keeping in mind that i cannot be 0, since $\frac{j}{i}$ would be undefined, the given language is just $\{a^{3i} \mid i \geq 1\}$, which is just $L(aaa(aaa)^*)$.

(4) Non-regular. Call the given language L_1. Suppose L_1 is regular. Then $L_1 \cap L(b^* a^*)$ is also regular. But this is simply $\{b^n a^n \mid n \geq 0\}$, which we know to be non-regular. Therefore, our assumption was false, and L_1 must be non-regular.

Problem 4 (10pt, 5pt each)

Below table is the transition table of deterministic automaton.
(Symbol → represents start state and symbol * represents final state)

	0	1
→ A *	B	D
B *	C	D
C	C	F
D	E	G
E *	C	G
F	C	F
G	H	D
H *	F	G

Answer the following questions.
(1) List the computed equivalence classes.

(sol) The equivalence classes are {A}, {B, E, H}, {C, F}, {D, G}

(2) Construct the minimum-state equivalent DFA.

(sol)

	0	1
{A}	{B, E, H}	{D, G}
{B, E, H}	{C, F}	{D, G}
{C, F}	{C, F}	{C, F}
{D, G}	{B, E, H}	{D, G}

Problem 5 (15pt)

Show that if A and B are regular languages over {0, 1}, then $A \vee B = \{x \vee y \mid x \in A,\ y \in B,\ |x| = |y|\}$ is also regular.

(sol)
Define DFA for proving given set is regular language.
Assume that DFA $M_A = (Q_A,\ \{0, 1\},\ \delta_A,\ S_A,\ F_A)$ and
$M_B = (Q_B,\ \{0, 1\},\ \delta_B,\ S_B,\ F_B)$ accept A and B respectively.
To accept language $A \vee B$, we build a product DFA M'.
$M' = (Q_A \times Q_B,\ \{0, 1\},\ \delta',\ [S_A,\ S_B],\ F_A \times F_B)$, where
$\delta'([p,\ q],\ 0) = \{(\delta_A(p,\ 0),\ \delta_B(q,\ 0))\}$,
$\delta'([p,\ q],\ 1) = \{(\delta_A(p,\ 0),\ \delta_B(q,\ 1)),\ (\delta_A(p,\ 1),\ \delta_B(q,\ 0)),\ (\delta_A(p,\ 1),\ \delta_B(q,\ 1))\}$

We can show that extension δ'' is well-defined from tansitive function δ' by induction.

Basis : $|x| = |y| = 0\ (x = \epsilon,\ y = \epsilon)$
$\delta_A{}'(p,\ \epsilon) = p,\ \delta_B{}'(q,\ \epsilon) = q$
$\delta''([p,\ q],\ x \vee y = \epsilon) = (p,\ q)$

Induction : Assumption, $|x| = |y| = n \rightarrow \delta''$ is well-defined.
Let $x' = xa,\ y' = yb$. Then $x' \vee y' = (x \vee y)(a \vee b)$.
Consider $a \vee b$.
1) $a \vee b = 1$ i.e. a = b = 1 or a = 1, b = 0 or a = 0, b = 1
Then $\delta''([p,\ q],\ (x \vee y)(a \vee b)) = \delta'(\delta''([p,\ q],\ (x \vee y)),\ 1)$
By assumption, $\delta''([p,\ q],\ (x \vee y))$ is well-defined.
So δ'' is also well-defined for $|x'| = |y'| = n+1$.
2) $a \vee b = 0$ i.e. a = b = 0.
$\delta''([p,\ q],\ (x \vee y)(a \vee b)) = \delta'(\delta''([p,\ q],\ (x \vee y)),\ 0)$
It can be developed by definition and assumption.

Hence we can construct transitive function δ' and build DFA M' accepting $A \vee B$.

Problem 6 (30pt)

(The Myhilll-Nerode theorem) The following three statements are equivalent:

(1) The set $L \subseteq \Sigma^*$ is accepted by some finite automaton.
(2) L is the union of some of the equivalence classes of a right invariant equivalence relation of finite index.
(3) Let equivalence relation R_L be defined by : xR_Ly if and only if for all z in Σ^*, xz is in L exactly when yz is in L. Then R_L is of finite index.

(sol)
(1) → (2) : Assume that L is accepted by some DFA $M=(Q, \Sigma, \delta, q_0, F)$. Let R_M be the equivalence relation $x R_M y$ if and only if $\delta(q_0, x) = \delta(q_0, y)$. R_M is right invariant since, for any z, if $\delta(q_0, x) = \delta(q_0, y)$, then $\delta(q_0, xz) = \delta(q_0, yz)$. The index of R_M is finite, since the index is at most the number of states in Q. Furthermore, L is the union of those equivalence classes that include a string x such that $\delta(q_0, x)$ is in F, that is , the equivalence classes corresponding to final states.

(2) → (3) : We show that any equivalence relation E satisfying (2) is a *refinement* of R_L; that is, every equivalence class of E is entirely contained in some equivalence class of R_L. Thus the index of R_L cannot be greater than the index of E and so is finite. Assume that xEy. Then since E is right invariant, for each z in Σ^*, $xzEyz$, and thus yz is in L if and only if xz is in L. Thus xR_Ly, and hence the equivalence class of x in E is contained in the equivalence class of x in R_L. We conclude that each equivalence class of E is contained within some equivalence class of R_L.

(3) → (1) : We must first show that R_L is right invariant. Suppose xR_Ly, and let w be in Σ^*. We must prove that xwR_Lyw; that is, for any z, xwz is in L exactly when ywz is in L. But since xR_Ly, we know by definition of R_L that for any v, xv is in L exactly when yv is in L. Let $v=wz$ to prove that R_L is right invariant.

CS322 형식 언어 및 오토마타 이론 중간고사

(2012년 10월 23일 PM 1:00 ~ 3:00)

학번 ________________

이름 ________________

점수 ________________

	1)	2)	3)	4)	Total
1.	/4	/4	/4	/4	/16
2.	/5	/5	/5	/5	/20
3.	/10				/10
4.	/4	/5	/5	/5	/19
5.	/5	/5	/5	/5	/20
6.	/5	/5	/5		/15

(아무것도 쓰지 마세요)

1. 다음의 집합이 finite, countably infinite, uncountable 중에 어디에 속하는지 결정하고 이를 설명하시오. (필요하다면, 자연수 집합의 모든 부분 집합은 finite 또는 countably infinite 하다는 것과 숙제에서 했던 증명 결과를 다시 증명하는 것 없이 이용하시오.) (16점)

1) Java로 쓰여진 모든 프로그램의 집합 (4점)

countably infinite. 기본 문자 집합 $\sum$의 $\sum^*$ 는 countably infinite하고 (숙제 #2) countably infinite한 집합의 모든 부분 집합 역시 countably infinite함. Java로 쓰여진 모든 프로그램의 집합은 $\sum^*$ 의 부분집합이므로 countably infinite함.

2) 집합 B를 $\{F,T\}$라 할 때 $B \times B \times \cdots$ (infinite product) (4점)

uncountable. $B \times B \times \cdots$ 집합을 A라 하고, A가 countable 하여 $f: N \to A$ 로 가는 bijection이 존재한다고 하자.

$f(i) = b_{i0} b_{i1} \cdots b_{ij} \cdots$ 이라고 할 때,

$$C_i = F \ if \ b_{ii} = T$$
$$\quad T \ if \ b_{ii} = F$$

로 정의하자. 이 경우, $C = C_0 C_1 \cdots C_i \cdots$ 는 모든 i에 대해서 $f(i)$와 같을 수 없다.

그러나 C는 A에 속해야 하므로 contradiction. 따라서 uncountable.

3) 모든 양의 유리수의 집합 (4점)

countably infinite. 양의 유리수 집합 Q에 속한 모든 원소들은 다음과 같이 2차원 테이블을 이용해 배치를 할 수 있다.

	1	2	3	...
1	1/1	2/1	3/1	...
2	1/2	x	3/2	...
3	1/3	2/3	x	...
..				

위의 테이블을 이용 1/1 1/2 2/1 3/1 3/1 ... 의 순서로 번호를 매길 수 있으며, 이 mapping 함수는 명백히 bijection이다. 따라서 countable infinite.

또는 양의 유리수 집합 Q는 분모 분자의 순서쌍인 자연수 집합의 Cartesian product N X N의 부분집합으로 표현 가능하다는 점을 이용할 수 있다. N X N가 countably infinite (숙제#1에 2번)하고, countably infinite한 집합의 부분 집합은 countably infinite하므로 Q는 countably infinite.

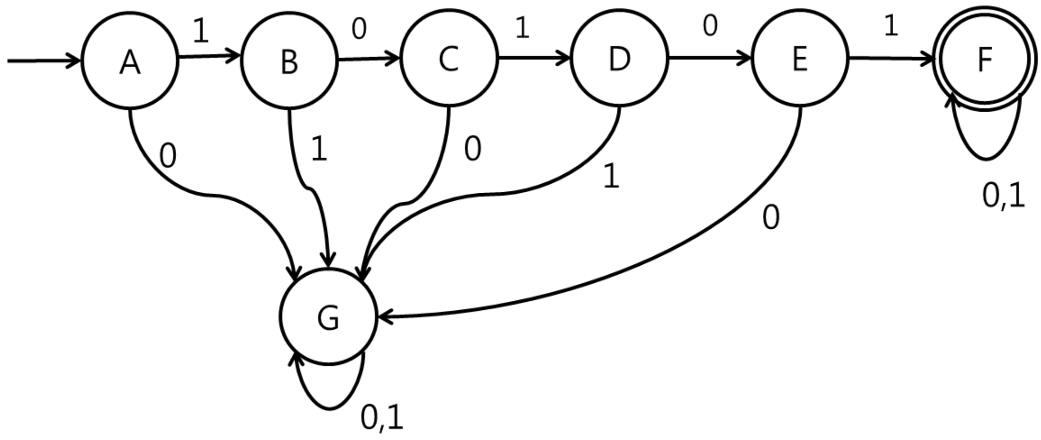
4) Countably infinite 집합의 finite product (4점)
Ex.) A_i가 각각 countably infinite 일 때, A_1 x … x A_n

두 countably infinite 집합의 Cartesian product가 countably infinite (숙제#1에 2번 참고)하다는 것으로부터 다음과 같이 mathematical induction으로 증명하면 된다.
"0 보다 큰 모든 자연수 n에 대해 임의의 countably infinite 집합, $A_1, \cdots, A_n$이 있을 때, $A_1 \times \cdots \times A_n$은 countably infinite 하다."

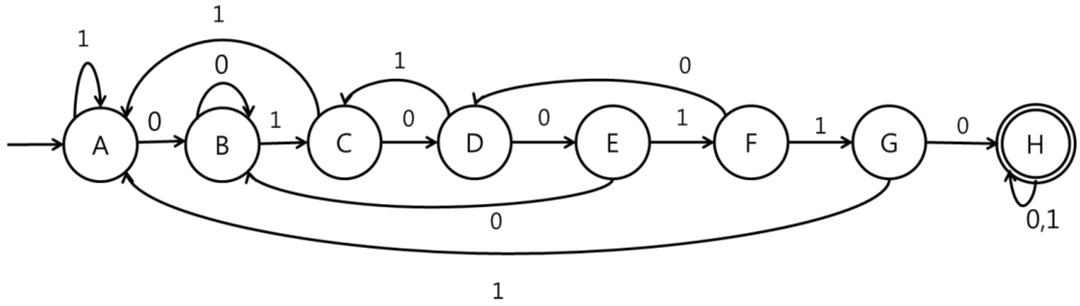
$(basis: n=1)$	By assumption, A is countably infinite.
$(induction\ hypothesis)$	$A_1 \times \cdots \times A_k$ is countably infinite.
$(induction\ step: n=k+1)$	From IH and assumption that A_{k+1} is countably infinite, cartesian product of two countably infinite set, $(A_1 \times \cdots \times A_k) \times A_{k+1}$ is countably infinite.

2. 다음의 언어를 인식하는 DFA를 각각 그리시오. (20점)

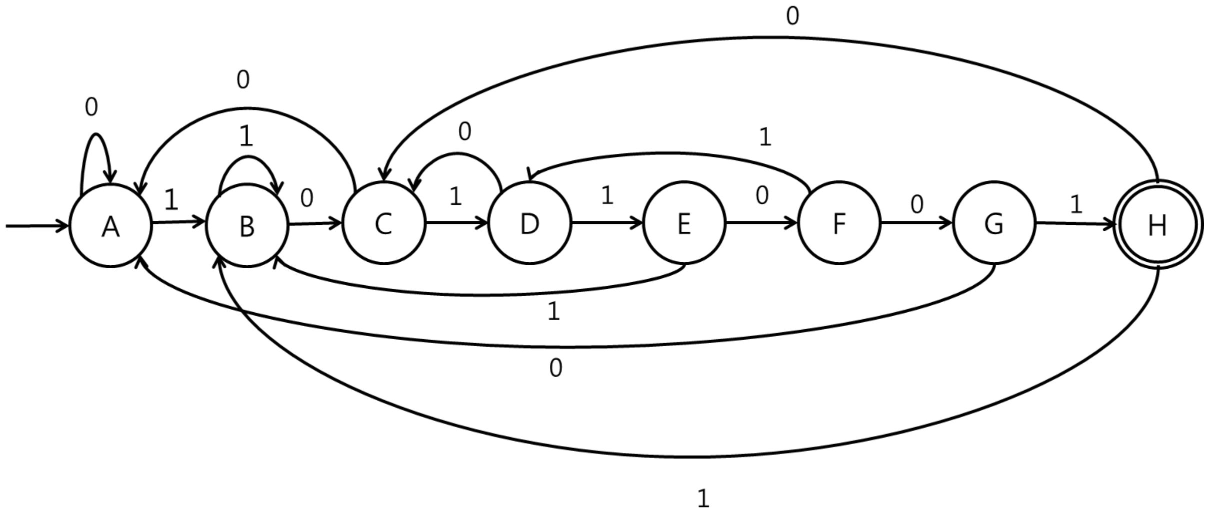
1) { w ∈ {1, 0}* | 문자열 w는 10101를 prefix로 가진다.} (5점)



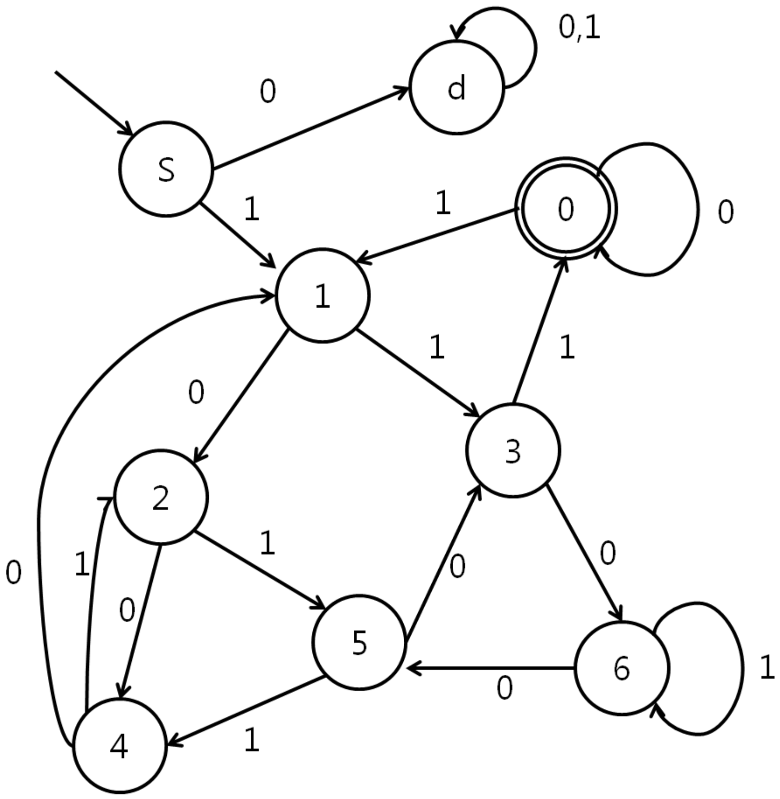
2) { w ∈ {1, 0}* | 문자열 w는 0100110을 substring으로 가진다.} (5점)



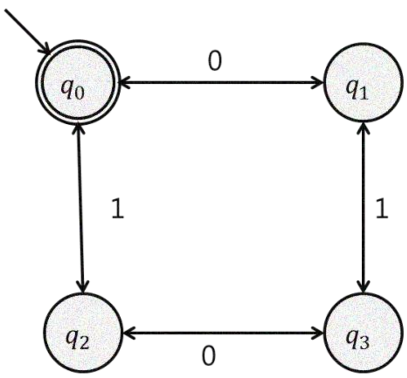
3) { w ∈ {1, 0}* | 문자열 w는 1011001을 postfix로 가진다.} (5점)



4) 양의 정수의 이진 표기 집합인 k가 있다. k mod 7의 결과 값이 '0'인 조건을 만족하는 k를 인식하는 DFA를 그리시오. (5점)



3. 다음의 DFA를 연립방정식을 이용하여 정규 표현식으로 고치시오. (풀이 과정을 쓰시오.) (10점)



$$\begin{pmatrix} q_0 = 0q_1 + 1q_2 + \varepsilon \\ q_1 = 0q_0 + 1q_3 \\ q_2 = 0q_3 + 1q_0 \\ q_3 = 0q_2 + 1q_1 \end{pmatrix} \Rightarrow \begin{pmatrix} q_0 = 0q_1 + 1q_2 + \varepsilon \\ \quad = 00q_0 + 01q_3 + 10q_3 + 11q_0 + \varepsilon \\ \quad = (00+11)q_0 + (01+10)q_3 + \varepsilon \\ q_3 = 0q_2 + 1q_1 \\ \quad = 10q_0 + 11q_3 + 00q_3 + 01q_0 \\ \quad = (10+01)q_0 + (00+11)q_3 \\ \quad = (00+11)^*(10+01)q_0 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} q_0 = (00+11)q_0 + (01+10)q_3 + \varepsilon \\ \quad = (00+11)q_0 + (01+10)(00+11)^*(10+01)q_0 + \varepsilon \\ \quad = (00+11+(01+10)(00+11)^*(10+01))^* \end{pmatrix}$$

4. 다음의 언어를 받아들이는 DFA가 존재하는가? 존재하지 않는다면 Pumping Lemma를 이용하여 이를 증명하라. (20점)

1) L = { a^nba^{2n} | n≥0 } (5점)

2.22(a) $L = \{a^nba^{2n} \mid n \geq 0\}$. Suppose L can be accepted by an FA, and let n be the integer in the statement of the pumping lemma. Let $x = a^nba^{2n}$. Then $x \in L$, and $|x| \geq n$. Therefore, by the statement of the pumping lemma, $x = uvw$ for some strings u, v, and w satisfying $|uv| \leq n$, $|v| > 0$, and $uv^mw \in L$ for every $m \geq 0$. The first of these three conditions implies that v is a string of a's from the first group of a's in x, and the second implies that $v \neq \Lambda$. Therefore, for some $j > 0$, $uv^2w = a^{n+j}ba^{2n}$. However, the third condition says that uv^2w must be in L. This contradiction proves that L cannot be

2) L = { a^ib^j | j≥2i } (5점)

$\forall n \geq 0,\ \exists w \in L$ such that $|w| \geq n$,

$w = a^nb^{2n}$ 이라 하자.

$\forall x, y, z$ such that $w = xyz$, $y \neq \varepsilon$, $|xy| \leq n$,

$y = a^i,\quad 1 \leq i \leq n \quad (\because y \neq \varepsilon)$

그런데,

$xy^2z = a^{n+i}b^{2n} \notin L \quad (\because 1 \leq i)$

따라서 pumping lemma에 의해 L은 regular language가 아니다.

3) L = { $a^{2i}b^{2j}$ | i≥0, j≥0 } (5점)

DFA가 존재합니다. (aa)*(bb)*

4) L = {w ∈ a* | |w|는 피보나치 수} (5점)

It is not regular. Given number n, find l that is the first fibonacci number larger than n. And find a fibonacci number m that is larger than 4l. Then, the difference between m and the next fibonacci number m′ is larger than 2n. Therefore, the intevral (m,m′) contains at least one x + y ∗ k + z for some k where x + y + z = l and x + y ≤ n and y != 0. Thus, L is not a regular language.

5. 다음 명제가 성립하는지 여부를 판단하시오. 성립하면 증명을 보이고, 그렇지 않다면 반증하시오. (15점)

1) $L_1 \cup L_2$ regular language이면, L_1, L_2 중 적어도 하나는 regular language이다. (5점)

2) $L_1 \cup L_2$가 regular language이고, L_1, L_2 중 하나가 유한하면, 나머지 하나는 regular language이다. (5점)

3) L_1이 regular language이고, L_2 = {uv : u ∈ L, v ∈ L^R}이면 L_2 역시 regular language 이다. (5점)

6. 다음의 Context-Free Language들에 대하여 물음에 답하여라. (19점)

- L_1 = {w ∈ (a+b)* | w는 첫 글자, 중간글자, 마지막 글자가 같고 길이가 홀수}
- L_2 = {x ∈ (a+b)* | $\#_b(x) = 3*\#_a(x)$}
- L_3 = {x ∈ (a+b)* | $\#_b(x) = k*\#_a(x)$} for natural numbers k>0

1) 언어 L_1을 조건 제시법으로 표현하여라. (4점)

2) 언어 L_1을 받아들이는 Context Free Grammar을 쓰라. (5점)

S -> aTa | bUb   T -> aTa | aTb | bTa | bTb | a   U -> aUa | aUb | bUa | bUb | b

3) 언어 L_2를 받아들이는 Context Free Grammar을 쓰라. (5점)

$$
\begin{aligned}
&S \to \varepsilon \mid aBBB \mid bA_1 \\
&A_1 \to aBB \mid bA_2 \\
&A_2 \to aB \mid bA_1 \\
&A_3 \to aS \mid bA_3A_1 \mid bA_2A_2 \mid bA_1A_3 \\
&B \to bS \mid aBBBB
\end{aligned}
$$

4) 언어 L_3를 받아들이는 Context Free Grammar을 쓰라. (5점)

$$G = (V, \{a, b\}, P, S)$$

$$V = \{S, A_1, A_2, \ldots, A_n, B\}$$

$$
P = \left\{
\begin{array}{r}
S \to \varepsilon \mid aB^n \mid bA \\
A_1 \to aB^{n-1} \mid bA_2 \\
\vdots \\
A_i \to aB^{n-i} \mid bA_{i+1} \\
\vdots \\
A_{n-1} \to aB^{n-(n-1)} \mid bA_n \\
A_n \to aS \mid bA_nA_1 \mid bA_{n-1}A_2 \mid \cdots \mid bA_2A_{n-1} \mid bA_1A_n \\
B \to bS \mid aB^{n+1}
\end{array}
\right\} \qquad (B^n = \underbrace{BB\cdots BB}_{n})
$$

Midterm Fall 2014

2014 년 10 월 10 월 21 일 화요일 오후 1:00 – 2:30
전산학동 E3-1 1501 호 (제 1 공동 강의실)

Prof. Choe, Kwang-Moo

Student ID:
Name:
Grading Result:

Problem				Total
1	/10			/10
2	/10	/10	/10	/30
3	/5	/10		/15
4	/10	/15	/15	/40
5	/10	/10		/20
6	/10	/5		/15
7	/10			/10
Grade				/140

페이지마다 학번과 이름을 적어주세요.

Student ID		**Name**	

1. 이산수학 복습 (Review of discrete mathematics) (10 점)

기본문자(vocabulary) $\sum$에서 정의된 문자열(string) 전체의 집합 $\sum^*$ 가 countably infinite 임을 증명하시오.

[정답]

Chapter 1-2 TP 18p 를 보고 bijective 함수 만들어냄

페이지마다 학번과 이름을 적어주세요.

Student ID		Name	

2. Regular Language 의 이해 (총 30 점)

아래 언어 L 이 regular language 인지 아닌지를 판별하고, regular 인 경우, DFA 나 regular expression 을 제시하고, regular language 가 아닌 경우, pumping lemma 를 이용하여 증명하시오.

2-1. $L = \{a^i b^j c^k \mid i, j, k \geq 1, i + k = j\}$ (10 점)

[정답]
DFA 로 표현할 수 없다. Pumping lemma 로 증명
Suppose L can be accepted by an FA, and let n be the integer in the statement of the pumping lemma. Let x = aibjck, i+k=j. Then x L

n >= 0, ₩exist w L such that |w| >= n
w=anbn+mcm, 라 하자.
x, y, z such that w =xyz, y , |xy| <=n,
y = ai, 1<=i<=n
그런데,
xy2z = an+ibn+mcm L
따라서, pumping lemma 에 의해 L 은 regular language 가 아니다.

페이지마다 학번과 이름을 적어주세요.

Student ID		**Name**	

2-2. $L = \{w \subseteq a^* \mid |w|\ is\ a\ Fibonacci\ number.\}$ **(10 점)**

[정답]

Regular 아님.

Given number n, find i that is the first Fibonacci number larger than n. And find a Fibonacci number m that is larger than 4i.

Then, the difference between m and the next fibonacci number m` is larger than 2n.

Therefore, the interval (m, m`) contains at least one x + y * k + z for some k where x + y + z = i and x + y ≤ n and y != 0.

Thus, L is not a regular language.

2-3. $L = \{a^{2i}b^{3j} \mid i \geq 0, j \geq 0\}$ **(10 점)**

[정답]

(aa)*(bbb)*

페이지마다 학번과 이름을 적어주세요.

Student ID		**Name**	

3. Regular Expression 의 이해 (총 15 점)

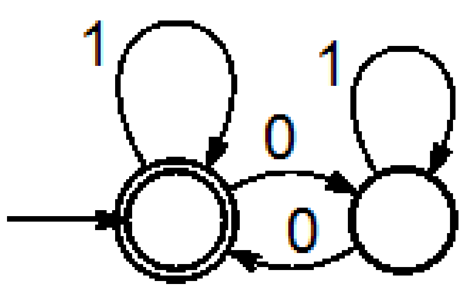
$\Sigma = \{0, 1\}$일 때, 두 개의 regular expression 이 같은 언어를 나타냄(denote)은 m-DFA 가 같음(isomorphism)을 보여 증명할 수 있다.
이를 이용하여, 아래의 두 개의 regular expression 이 같음을, m-DFA 를 그려 증명하시오.
(참고) RE 를 ε-NFA 로 고칠 때, 가능한 한 state 와 ε-move 의 수를 줄이시오.

3-1. $(\mathbf{0}^*\mathbf{1})^*\mathbf{0}^*$ 와 $(\mathbf{0}+\mathbf{1})^*$ (5 점)

[정답]
$(\mathbf{0}^*\mathbf{1})^*\mathbf{0}^* \rightarrow (\mathbf{0}+\mathbf{1})^*$
문자열을 right to left 방식으로 읽는다고 생각하자. ε 의 경우: 자명
길이가 1 이상인 문자열에 대해 postfix 가 0 이라면 1 이 나오기 전까지 0*로 표현가능하다.
그 다음 문자가 1 이 오면 다음 1 을 만날 때 까지 0*1 로 표현가능하다. 이 과정을 recursive 하게 하면 모든 문자열을 (0*1)*0*로 나타낼 수 있다.

3-2. 1*+(1*01*01*)* 와 (01*0+1)* (10 점)

[정답] 0=a, 1=b



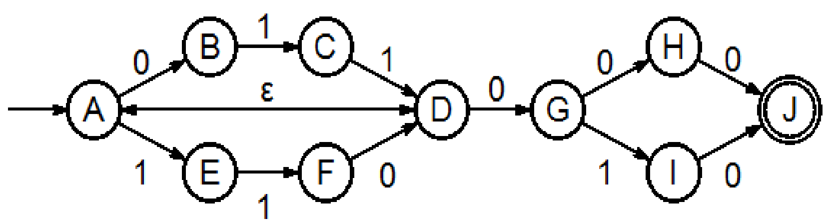
페이지마다 학번과 이름을 적어주세요.

Student ID		Name	

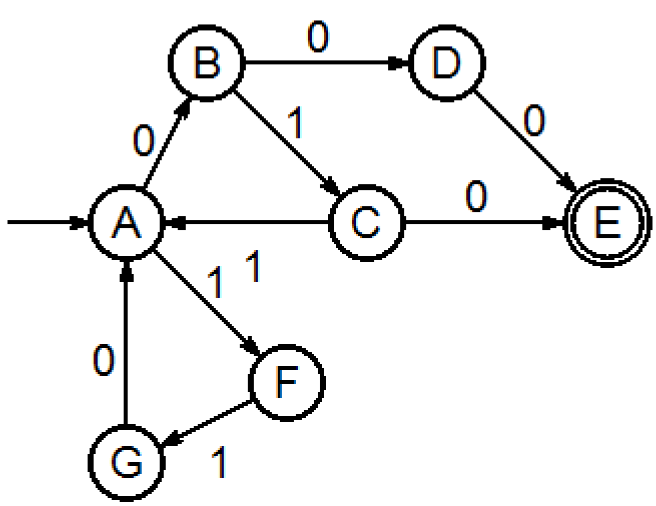
4.	Minimal state DFA 의 이해 (총 40 점)

4-1.	$\sum = \{0, 1\}$ 일 때, $(011 + 110)^*(000 + 010)$로 정의된 언어(language) L 에 대하여 minimal state DFA 를 그려라. (Hint: RE → ε-NFA → DFA → m-DFA).
(참고) RE 를 ε-NFA 로 고칠 때, 가능한 한 state 와 ε-move 의 수를 줄이시오. 또한 dead state 는 나타내지 마시오. (10 점)

[정답]
ε-NFA



DFA(minimal)



위의 DFA 는 minimal 이다. 왜냐하면 모든 state p, q 에 대해 p ≠ q 이기 때문이다. (표 참조)

State / input	0	1
q0:{A,D}	{B,G}	{E}
q1:{B,G}	{H}	{C,I}
q2:{E}	empty	{F}
q3:{H}	{J}	Empty
q4:{C,I}	{J}	{A,D}
q5:{F}	{A,D}	Empty
q6:{J}	empty	empty

페이지마다 학번과 이름을 적어주세요.

Student ID		**Name**	

4-2. DFA = (Q, Σ, δ, q_0, F)에 대하여 다음과 같은 관계(relation) ≡을 정의한다.

$$\equiv = \{(\mathbf{p},\mathbf{g}) \in \boldsymbol{Q} \times \boldsymbol{Q} \mid \forall \boldsymbol{w} \in \boldsymbol{\Sigma}^*, \boldsymbol{\delta}^*(\boldsymbol{p},\boldsymbol{w}) \in \boldsymbol{F} \Leftrightarrow \boldsymbol{\delta}^*(\boldsymbol{q},\boldsymbol{w}) \in \boldsymbol{F}\}$$

관계 ≡는 equivalent relation 이며, equivalent class $[q]_\equiv$는 정규식(regular expression)으로 표현 가능하다. 어떤 상태 q 에 대하여 $[q]_\equiv$를 정규식으로 나타내는 방법은 final state 를 q 라 두고, final state 를 바꾼 DFA 가 받아들이는 언어를 연립정규방정식을 푸는 방법(교과서 TP Chapter 4, 23-24p)으로 구하면 된다. (15 점)

(1) 4-1 의 DFA 에 대하여 각 equivalent class $[q]_\equiv$을 구하기 위한 연립정규방정식을 쓰시오. state 가 ∅인 경우, 나타내지 마시오. (5 점)

(2) 위 연립정규방정식을 사용하여 모든 equivalent class $[q]_\equiv$을 정규식으로 나타내어라. (10 점)

[정답]

(1) [A]:

A = 0B + 1F + e

B = 1C

C = 1A

F = 1G

G = 0A

[B]:

A = 0B + 1F

B = 1C + e

C = 1A

F = 1G

G = 0A

[C]:

A = 0B + 1F

B = 1C

C = 1A + e

F = 1G

G = 0A

[D]:

A = 0B + 1F

B = 0D + 1C

C = 1A

D = e

F = 1G

G = 0A

[F]:

A = 0B + 1F

B = 1C

C = 1A

F = 1G + e

G = 0A

[G]:

A = 0B + 1F

B = 1C

C = 1A

F = 1G

G = 0A + e

[E]:

A = 0B + 1F

B = 0D + 1C

C = 0E + 1A

D = 0E

E = e

F = 1G

G = 0A

(2)

[A] = (011+110)*

[B] = (011+110)*0

[C] = (011+110)*01

[D] = (011+110)*00

[E] = (011+110)*(000+010)

[F] = (011+110)*1

[G] = (011+110)*11

페이지마다 학번과 이름을 적어주세요.

Student ID		Name	

4-3. 다음은 어떤 DFA 를 연립정규방정식으로 나타낸 것이다.

A = 0B + 1A

B = 0B + 1C

C = 0B + 1D

D = 0B + 1A + ε

시작 상태(initial state)를 A 라고 할 때, $[A]_{\equiv} = \boldsymbol{\varepsilon + 1 + 11 + (0+1)^{*}111}$ 가 됨을 보여라. (15 점)

[정답]

$[\boldsymbol{A}]_{\equiv}$를 구하기 위해 상태 A 를 final state 라 둔다. 그러면 연립방정식이 다음과 같이 바뀐다.

A=0B+1A+ε

B=0B+1C

C=0B+1D

D=0B+1A

에서 A=D+ ε 이다. 이것을 D=0B+1A 에 대입하면 D=0B+1D+1=C+1 이 된다. 이를 C=0B+1D 에 대입하면 C=0B+1C+11 이고 따라서 C=B+11 이 된다. D=C+1 에 넣으면 D=B+11+1 이 되고 A=D+ ε 에 의해 A= B+11+1+ε 이다.

식 B=0B+1C 에서 B = 0*1C 이고, 이를 C=B+11 에 대입하면 C= 0*1C+11 이 된다. 이 식을 다시 B = 0*1C 에 넣게 되면

B = 0*1(0*1)*11 = (0*1)* 0*111.

따라서 A= B+11+1+ε = (0*1)* 0*111 + 11+1+ε 이다.

여기서 (0*1)* 0* = (0+1)* 이므로

A= (0*1)* 0*111 + 11+1+ε = (0+1)* 111 + 11+1+ε.

페이지마다 학번과 이름을 적어주세요.

Student ID		Name	

5. **Context Free Grammar(CFG)의 이해와 표현 (총 20 점)**

5-1. 다음 언어의 CFG 를 쓰고, 각 non-terminal symbol(비말단기호)들은 각각 어떤 문자열을 만들어내는지 설명하시오. (10 점)

$$L = \{ w \in (a,b)^* \mid \#_b(w) = 2 \times \#_a(w) \}$$

$\#_a$: The number of a words

$\#_b$: The number of b words

[답안]

$S \rightarrow \varepsilon \mid aBB \mid bA_1$

$A_1 \rightarrow aB \mid BA_2$ → A1: b 의 개수가 a 의 개수의 2 배보다 하나 적은 문자열

$A_2 \rightarrow aS \mid bA_1A_2 \mid bA_2A_1$ → A2: b 의 개수가 a 의 개수의 2 배보다 2 개 적은 문자열

$B \rightarrow bS \mid aBBB$ → B: b 의 개수가 a 의 개수의 2 배보다 하나 많은 문자열

페이지마다 학번과 이름을 적어주세요.

Student ID		Name	

5-2. 5-1 에서 쓴 CFG 를 이용하여, 다음 언어의 CFG 를 쓰시오. (10 점)

$$L = \{ w \in (a,b)^* \mid \#_a(w) = k \times \#_b(w) \}$$
$\#_a$: The number of a words
$\#_b$: The number of b words
k: k>0, natural numbers

[답안]

5-1 에서 a 와 b 개수 바뀜

G = (V, {a,b}, S, P)

V = {S, A, B_1, B_2, B_3, …, B_k}

P = $S \rightarrow \varepsilon \mid bA^k \mid aB_1$

$A \rightarrow aS \mid bA^{n+1}$

$B_1 \rightarrow bA^{k-1} \mid aB_2$

…

$B_i \rightarrow bA^{k-i} \mid aB_{i+1}$

…

$B_{k-1} \rightarrow bA^{k-(k-1)} \mid aB_k$

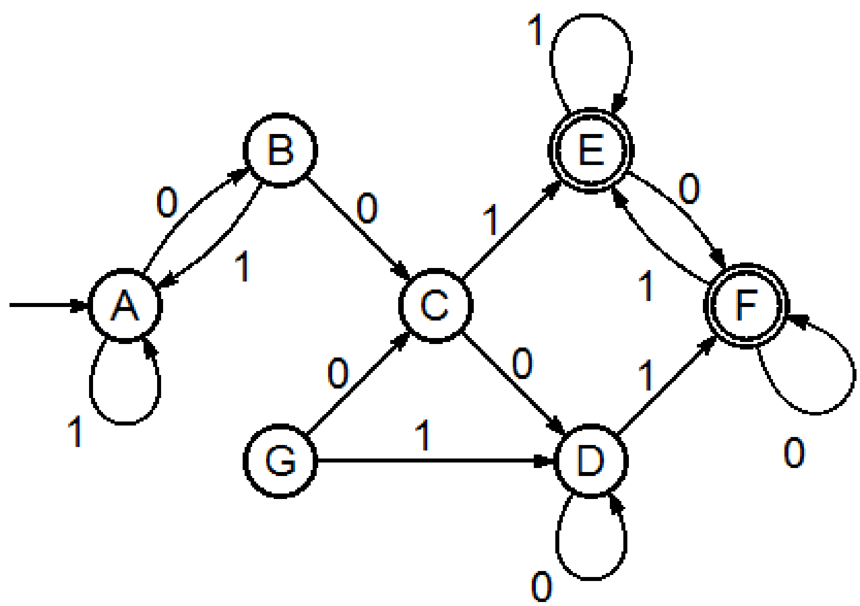
$B_k \rightarrow bA^{k-1} \mid aB_kB_1 \mid aB_{k-1}B_2 \mid \ldots \mid aB_2B_{k-1} \mid aB_1B_k$

페이지마다 학번과 이름을 적어주세요.

Student ID		**Name**	

6. DFA 와 table filling algorithm 의 이해

다음과 같이 DFA D 가 주어져 있다. 각 질문에 답하여라. (총 15 점)



6-1. DFA D 에 대해 table filling algorithm 을 실행할 때, 각 iteration 에 따른 table 의 변화를 나타낸 것이다. Iteration 마다 변한 table 을 완성하여라. Iteration 은 table 의 위에서부터 가로를 다 확인한 다음에 아래로 진행한다. (단, 두 state 가 distinguishable 이면 table 에 d 로 표현한다.) (10 점)
(참고) 교과서 TP Chapter 4, 20p

Basis:

B					
C					
D					
E					
F					
	A	**B**	**C**	**D**	**E**

페이지마다 학번과 이름을 적어주세요.

Student ID		Name	

change : 1

	A	B	C	D	E
B					
C					
D					
E					
F					

change : 2

	A	B	C	D	E
B					
C					
D					
E					
F					

change : 3

	A	B	C	D	E
B					
C					
D					
E					
F					

change : 4

	A	B	C	D	E
B					
C					
D					
E					
F					

change : 5

B					
C					
D					
E					
F					
	A	**B**	**C**	**D**	**E**

6-2. 6-1)의 table 을 참고하여 DFA D 의 minimal state DFA 를 그려라. (5 점)

[정답]

6-1)

Basis:

B					
C					
D					
E	d	d	d	d	
F	d	d	d	d	
	A	B	C	D	E

Iter : 1

B					
C	d				
D					
E	d	d	d	d	
F	d	d	d	d	
	A	B	C	D	E

Iter : 2

B					
C	d	d			
D					
E	d	d	d	d	
F	d	d	d	d	

	A	B	C	D	E

Iter : 3

	A	B	C	D	E
B					
C	d	d			
D	d				
E	d	d	d	d	
F	d	d	d	d	

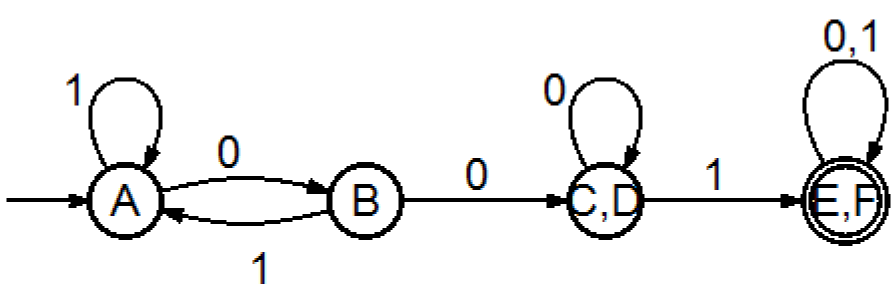
Iter : 4

	A	B	C	D	E
B					
C	d	d			
D	d	d			
E	d	d	d	d	
F	d	d	d	d	

Iter : 5

	A	B	C	D	E
B	d				
C	d	d			
D	d	d			
E	d	d	d	d	
F	d	d	d	d	

6-2)



페이지마다 학번과 이름을 적어주세요.

Student ID		**Name**	

7. Mealy Machine 과 일반 DFA 의 이해

다음 Mealy Machine 의 입력 문자열이 “1+2*2=”일 때, 그 출력 값을 구하시오. (5 점)

$M_e = (Q, \Sigma, \Pi, \delta, \lambda, q_0)$
$Q = \{s, n, o, f\}, \Sigma = \{1,2,3,4,+,*=\}$
$\Pi = \{push1, push2, push+, push *, calc, print\}$ ($p \in \Pi$는 프로그램 블록 또는 함수)
$\delta = \{\delta(s,1) = n, \delta(s,2) = n, \delta(n,+) = o, \delta(n,*) = o, \delta(o,1) = n, \delta(o,2) = n, \delta(n,=) = f\}$
$\lambda = \{\lambda(s,1) = push1, \lambda(s,2) = push2,$
$\quad \lambda(n,+) = push+, \lambda(n,*) = push *,$
$\quad \lambda(o,1) = calc, \lambda(o,2) = calc,$
$\quad \lambda(n,=) = print\}$
$q_0 = s$

프로그램 블록(함수) 설명 (괄호 안은 Stack 을 아는 분을 위한 설명입니다.):
전역변수S: 빈 list(array)(ordered set)로 초기화됨 (S = 빈 Stack)
$1+1=2, 1+2=3, 2+1=3, 2+2=4, 1+2=3, 1+4=5$
$1\times1=1, 1\times2=2, 2\times1=2, 2\times2=4, 3\times2=6$

$push1$: S의 위(top)에 1 를 추가, $S.push(1)$

$push2$: S의 위에 2 를 추가, $S.push(2)$

$push +$: S의 위에 +을 추가, $S.push(+)$

$push *$: S의 위에 *을 추가, $S.push(*)$

$calc$: $v_1 \in \{1,2,3,4\}, v_1 = S.pop()$
$op \in \{+,*\}, op = S.pop()$
$v_2 \in \{1,2,3,4\}, v_2 = S.pop()$
$if\ op = +\ then\ push(v_1 \times v_2)$ & $op = *\ then\ push(v_1 + v_2)$

$print$: S의 제일 끝 원소를 출력 ($print\ S.pop()$)

[정답]

4

Chapter 12 Boolean Algebras

12.1 Lattices and Algebraic Systems

Let $(A, \leq)$ be a poset. Then

$ub(a, b) = \{c \mid a \leq c, b \leq c\}$ ***upper*** *bound*

$lub(a, b) = \{c \mid c \in ub(a,b), \nexists d \text{ .э. } d \in ub(a,b), d \leq c, c \neq d\}$

least *upper bound* $(d < c)$

$lb(a, b) = \{c \mid c \leq a, c \leq b\}$ ***lower*** *bound*

$glb(a, b) = \{c \mid c \in lb(a,b), \nexists d \text{ .э. } d \in lb(a,b), c \leq d, c \neq d\}$

greatest *lower bound* $(c < d)$

A poset $(A, \leq)$ is called a ***lattice****,*

*if $\forall a, b \in A$, $\exists$**unique** $lub(a, b)$ and $glb(a, b)$.*

$(A, \vee)$ and $(A, \wedge)$ are algebraic systems.

Let $(A, \leq)$ be a ***lattice****, an* ***algebraic system*** *$(A, \vee, \wedge)$, is defined by the* ***lattice*** *$(A, \leq)$ where*

$\vee: A \times A \to A$ *least upper bound(**join**)*

$\wedge: A \times A \to A$ *greatest lower bound(**meet**)*

$(P(S), \cup, \cap)$ is an algebraic system defined by the lattice $(P(S), \subseteq)$.

(N, max, min): *$(N, \leq)$.*

(N^+, lcm, gcd): *$(N^+, |)$. $a|b$, if a divides b.*

Theorem 12.1(*ub, lb*)

$a \wedge b \leq a, b \qquad\qquad a, b \leq a \vee b$

Proof

$\wedge$(**glb**) *is lower bound, and* $\vee$(**lub**)*is upper bound.*

Theorem 12.2

If $a \leq b$, $c \leq d$,

$a \vee c \leq b \vee d$, *and* $\quad a \wedge c \leq b \wedge d$.

Proof

$b \leq b \vee d,\ a \leq b \Rightarrow a \leq b \vee d.$ *ub, tran.*

$d \leq b \vee d,\ c \leq d \Rightarrow c \leq b \vee d.$ *ub, tran.*

$\therefore\ a \vee c \leq b \vee d.$ *lub* $\leq$ *ub*

$a \wedge c \leq a,\ a \leq b \Rightarrow a \wedge c \leq b.$ *lb, tran.*

$a \wedge c \leq c,\ c \leq d \Rightarrow a \wedge c \leq d.$ *lb, tran.*

$\therefore\ a \wedge c \leq b \wedge d.$ *lb* $\leq$ *glb*

12.2 Principle of Duality

left vs. right
freedom vs. equality
Capitalism vs. Socialism
Flowers in the mirror
$(A, \leq)$ *vs.* $(A, \geq)$

$(A, \vee, \wedge)$ *vs* $(A, \wedge, \vee)$.

12.3 Basic Properties of Algebraic systems defined by Lattices

Let $(A, \vee, \wedge)$ be a algebraic system defined by the lattice $(A, \leq)$.

Theorem 12.3 *Both $\vee$ and $\wedge$ are **commutative**.*

$a \vee b = b \vee a. \qquad a \wedge b = b \wedge a.$

Proof

by the definition lub and glb.

Theorem 12.4 *Both $\vee$ and $\wedge$ are **associative**.*

$a \vee (b \vee c) = (a \vee b) \vee c.$

$a \wedge (b \wedge c) = (a \wedge b) \wedge c.$

Proof

Let $a \vee (b \vee c) = g$ and $(a \vee b) \vee c = h$.

$\therefore\ a \leq g,\ b \vee c \leq g,\ \therefore\ b \leq g,\ c \leq g.$ *g is ub.*

$\therefore\ a \vee b \leq g,\ c \leq g,\ (a \vee b) \vee c \leq g.$ *lub ≤ ub*

$\therefore\ h \leq g.$

$a \leq h,\ b \leq h,\ c \leq h.$ *h is ub.*

$\therefore\ a \leq h,\ b \vee c \leq h,\ a \vee (b \vee c) \leq h,$ *lub ≤ ub*

$\therefore\ g \leq h.$

Theorem 12.5 *Both $\vee$ and $\wedge$ are **idempotent**.*

$a \vee a = a \qquad a \wedge a = a$

Proof

$a \leq a \vee a$, *and since* $a \leq a$, $a \vee a \leq a$ *lub≤ub.*

$\therefore\ a \vee a = a$

Theorem 12.6 abortion property.

$a \vee (a \wedge b) = a \qquad a \wedge (a \vee b) = a.$

Proof

$a \le a \vee (a \wedge b)$. $\qquad$ $a \vee (a \wedge b)$ *is ub of* a

Since, $a \wedge b \le a$ $\qquad$ $a \wedge b$ *is lb of* a

$a \vee (a \wedge b) \le a \vee a = a$. $\qquad$ *T12.2, T12.5*

$\therefore\ a \vee (a \wedge b) = a$.

$(A, \vee)$ *and* $(A, \wedge)$ *are (**commutative) semigroups**.*

Example (N, max, min) defined by $(N, \le)$.

$\min(a, b) \le a, b \le \max(a, b)$

$a \le b$, *and* $c \le d \Rightarrow$

$\min(a,c) \le \min(b,d)$, *and* $\max(a,c) \le \max(b,d)$.

$\max(a, b) = \max(b, a)$,

$\min(a, b) = \min(b, a)$. $\qquad$ ***commutative***

$\min(a, \min(b, c)) = \min(\min(a, b), c)$,

$\max(a, \max(b, c)) = \max(\max(a, b), c)$. $\qquad$ ***assoc***.

$\max(a, a) = \min(a, a) = a$. $\qquad$ ***idempotent***

$\max(a, \min(a, b)) = a$,

$\min(a, \max(a, b)) = a$. $\qquad$ ***absortion***

(N, min) is a semigroup.

(N, max) is a momoid with identity 0.

Example (N^+, lcm, gcd) *defined by* $(N^+, |)$.

$gcd(a, b) \mid a, b \mid lcm(a, b)$.

$a \mid b$ *and* $c \mid d \Rightarrow$

$gcd(a, c) \mid gcd(b, d)$, *and* $lcm(a, c) \mid lcm(b, d)$.

$lcm(a, b) = lcm(b, a)$

$gcd(a, b) = gcd(b, a)$. ***commutative***

$lcm(a, lcm(b, c)) = lcm(lcm(a, b), c)$,

$lcm(a, lcm(b, c)) = lcm(lcm(a, b), c)$. ***assoc.***

$lcm(a, a) = gcd(a, a) = a$. ***idempotent***

$lcm(a, gcd(a, b)) = a$,

$gcd(a, lcm(a, b)) = a$. ***absortion***

(N^+, lcm) *is a monoid with identity with 1*

(N^+,gcd) *is a semigroup.*

Example $(P(S), \cup, \cap)$ *defined by* $(P(S), \subseteq)$.

$A \cap B \subseteq A, B \subseteq A \cup B$.

$A \subseteq B$ *and* $C \subseteq D$

$A \cup C \subseteq B \cup D$, *and* $A \cap C \subseteq B \cap D$.

$A \cup B = B \cup A$

$A \cap B = B \cap A$ ***commutative***

$(A \cup B) \cup C = A \cup (B \cup C)$

$(A \cap B) \cap C = A \cap (B \cap C)$ ***associative***

$A \cup A = A \cap A = A$ ***idempotent***

$A \cup (A \cap B) = A$

$A \cap (A \cup B) = A$ ***absortion***

$(P(S), \cup)$ *is a monoid with identity* $\emptyset$.

$(P(S), \cap)$ *is a monoid with identity* S.

12.4 Distributive and Complemented Lattices

A lattice is a **distributive** *lattice, if*
the meet($\wedge$) operation distributes over the join($\vee$) operation and
the join($\vee$) operation distributes over the meet($\wedge$) operation.

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$
$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

Since $\vee$ and $\wedge$ are comutative

$$(a \vee b) \wedge c = (a \wedge c) \vee (b \wedge c)$$
$$(a \wedge b) \vee c = (a \vee c) \wedge (b \vee c)$$

Theorem 12.7 *If the meet operation distributes over the join operation, the join operation distributes over the meet operation, and vice versa.*

Proof

Assume $\wedge$ distributes over $\vee$ and consider

$(a \vee b) \wedge (a \vee c)$	
$= ((a \vee b) \wedge a) \vee ((a \vee b) \wedge c)$	*$\wedge$ distributes over $\vee$*
$= a \vee ((a \vee b) \wedge c)$	*abortion*
$= a \vee ((a \wedge c) \vee (b \wedge c))$	*$\wedge$ distributes over $\vee$*
$= ((a \vee (a \wedge c)) \vee (b \wedge c)$	*$\vee$ is associative*
$= a \vee (b \wedge c)$	*abortion*

Let $(A, \leq)$ be a lattice.

*$0 \in A$ is called **universal lower bound(bottom)**, if $^{\forall}a \in A, 0 \leq a$.*

*$1 \in A$ is called **universal upper bound(top)**, if $^{\forall}a \in A, a \leq 1$.*

***Theorem 12. 7** If there exists a **universal lower(upper)bound**, it is **unique**.*

***Proof** Assume $^{\exists}$**two** universal lower bound a and b*

$a \leq b, b \leq a. \therefore a = b$.

Example

$(2^A, \subseteq)$ universal lower bound(0) is $\emptyset$,
universal upper bound(1) is A.

***Theorem 12.8** Let $(A, \leq)$ ba a lattice, and 0 and 1 be **universal lower** and **upper bounds**. Then $^{\forall}a \in A$,*

$0 \vee a = a$,	*0 is **identity** for* $\vee$
$0 \wedge a = 0$,	*0 is **zero** for* $\wedge$
$1 \vee a = 1$,	*1 is **zero** for* $\vee$
$1 \wedge a = a$.	*1 is **identity** for* $\wedge$

Proof

$1 \leq 1 \vee a$,(ub) and $1 \vee a \leq 1$(uub)

$\therefore 1 \vee a = 1$.

$1 \wedge a \leq a$(lb) and $1 \wedge a \geq a \wedge a$(uub, Thm12.2) $= a$

$\therefore 1 \wedge a = a$.

Let $(A, \leq)$ ba a lattice, and 0 and 1 be a ***universal lower*** *and* ***upper bounds****, respectively. Then*

a is said to be a ***complement*** *of b, if*

$a \vee b = 1$ *and* $a \wedge b = 0$.

(Example) $(P(A), \subseteq)$ $\quad A \cup \overline{A} = S, A \cap \overline{A} = \varnothing$.

If a is a ***complement*** *of b, b also is a* ***complement*** *of a.*
(commutative)

An element may have ***many*** *complements, if any.*

Example: Fig 12.1

But 0 and 1 are ***unique complements*** *of each other.*

A lattice is said to be a ***complemented lattice****, if every elements has* ***complement(s)****.*

Theorem 12.9 *In a* ***distributive*** *lattice, if an element has a* ***complement*** *then this* ***complement*** *is* ***unique****.*

Proof *Suppose a has two complements* a_1 *and* a_2

$a \vee a_1 = 1, a \wedge a_1 = 0.$

$a \vee a_2 = 1, a \wedge a_2 = 0.$

$a_1 = a_1 \wedge 1$ — *1 is the* ***identity*** *for* $\wedge$

$= a_1 \wedge (a \vee a_2)$ — a_2 *is the* ***complement*** *of a*

$= (a_1 \wedge a) \vee (a_1 \wedge a_2)$ — $\wedge$ ***distributes*** *over* $\vee$

$= \underline{0} \vee (a_1 \wedge a_2)$	*a_1 is the **complement** of a*
$= (\underline{a \wedge a_2}) \vee (a_1 \wedge a_2)$	*a_2 is the **complement** of a*
$= (a \vee a_1) \wedge a_2$	*$\wedge$ **distributes** over $\vee$*
$= 1 \wedge a_2$	*a_1 is the **complement** of a*
$= a_2$	*1 is an **identity** for $\wedge$.*

12.5 Boolean Lattice and Boolean algebra

*A **complemented** and **distributive** lattice is called a **boolean lattice**.*

*Since every element has a unique complement, **complement** is an **algebraic operator**.*

***complement**: **unary** operator on A.*

$\neg: A \rightarrow A$

*A boolean lattice $(A, \leq)$ defines an **boolean algebra** $(A, \vee, \wedge, \neg)$*

- $\vee$ *join*
- $\wedge$ *meet*
- $\neg$ *complement*

$a \vee \neg a = 1,$ $\quad$ $a \wedge \neg a = 0.$

$\neg(\neg a) = a.$

Example

$(2^A, \cup, \cap, \bar{\ })$ *is a **boolean algebra** defined by the **boolean lattice** $(2^A, \subseteq)$*

Theorem 12.10 DeMorgan's laws

$\neg(a \vee b) = \neg a \wedge \neg b,$
$\neg(a \wedge b) = \neg a \vee \neg b.$

Proof

$(a \vee b) \vee (\neg a \wedge \neg b)$	
$= ((a \vee b) \vee \neg a) \wedge ((a \vee b) \vee \neg b)$	$\vee$ dist. over $\wedge$
$= ((a \vee \neg a) \vee b) \wedge (a \vee (b \vee \neg b))$	$\vee$ comm, assoc.
$= (1 \vee b) \wedge (a \vee 1)$	complement
$= 1 \wedge 1 = 1$	uub
$(a \vee b) \wedge (\neg a \wedge \neg b)$	
$= (a \wedge(\neg a \wedge\neg b)) \vee (b \wedge(\neg a \wedge\neg b)$	$\wedge$ dist. over $\vee$
$= 0 \vee 0 = 0$	..., ulb

$\therefore$ *$a \vee b$ is a complement of $\neg a \wedge \neg b$*

$\neg(a \vee b) = \neg a \wedge \neg b,$
$\neg(a \wedge b) = \neg a \vee \neg b.$

*$(\{0, 1\}, \vee, \wedge, \neg)$ is a **boolean algebra** defined by*
a lattice $(\{0, 1\}, \leq)$ with $(0 \leq 1)$

$0 \vee 0 = 0,\ \ 0 \vee 1 = 1,\ \ 0 \vee 0 = 1,\ \ 1 \vee 1 = 1.$
$0 \wedge 0 = 0,\ \ 0 \wedge 1 = 0,\ \ 0 \wedge 1 = 0,\ \ 1 \wedge 1 = 1.$
$\neg 0 = 1, \neg 1 = 0.$

*$(\{0, 1\}, \wedge, \vee, \neg)$ is a **boolean algebra** defined by*
a lattice $(\{0, 1\}, \leq)$ with $(1 \leq 0)$

$0 \wedge 0 = 0,\ \ 0 \wedge 1 = 0,\ \ 0 \wedge 1 = 0,\ \ 1 \wedge 1 = 1.$
$0 \vee 0 = 0,\ \ 0 \vee 1 = 1,\ \ 0 \vee 0 = 1,\ \ 1 \vee 1 = 1.$

Principle of Duality

12.6 Uniqueness of Finite Boolean Algebra

Boolean algebra

freedom?

Every *finite* ***boolean algebra*** *has* 2^n *elements*

for ***some*** $n > 0$.

There is a ***unique boolean algebra*** *of* 2^n *element*

for ***every*** $n > 0$.

$a \in A$ *is called to* ***cover*** $b \in A$,

if $b \le a$, ${}^{\exists}c \in A$.϶. $b < c < a$.*(Hasse diagram)*

$b \le c \le a,\ b \ne c,\ c \ne a$

Let $(A, \le)$ *be a lattice with* ***bottom*** *0.*

An element is called ***atom****, if it covers 0.*

Lemma 12.0

Let $(A, \le)$ *be a* ***finite*** *lattice with* ***bottom*** *0.*

${}^{\forall}b \ne 0 \in A$, ${}^{\exists}$*an* ***atom*** $a \in A$.϶. $a \le b$.

Proof

If b is an atom, $b \le b$.

If b is not an atom, since $(A, \le)$ *is a* ***finite*** *lattice,*

${}^{\exists}$*a chain* $0 < b_i < \ldots < b_2 < b_1 < b$

b_i *is an atom*

Lemma 12.1 *In a* ***distributed*** *lattice,*

if $b \wedge \neg c = 0$, then $b \leq c$.

Proof

$(b \wedge \neg c) \vee c = c$	$b \wedge \neg c = 0$
$(b \vee c) \wedge (\neg c \vee c) = c$	distributive
$b \vee c = c$	1 is a identity for $\wedge$.
$\therefore\ b \leq c$	c is lub.

Lemma 12.2 *Let* $(A, \vee, \wedge, \neg)$ *be a* ***finite boolean*** *algebra. Then*

$^{\forall}b \in A\text{-}\{0\}, 1 \leq {}^{\exists}k \leq |A|, 1 \leq {}^{\forall}i \leq k$, a_i's are **atoms** of A,

$$b = a_1 \vee a_2 \vee \ldots \vee a_k.$$

Proof

Since $^{\forall}b \in A\text{-}\{0\}, 1 \leq {}^{\exists}k \leq |A|, 1 \leq {}^{\forall}i \leq k$,

a_i's are **atoms** of A and $a_i \leq b$. — L12.0

$c = a_1 \vee a_2 \vee \ldots \vee a_k \leq b$. — T12.2

Suppose $b \wedge \neg c \neq 0$.

$^{\exists}$an atom $a \in A$.∍. $a \leq b \wedge \neg c$. — L12.0

$\therefore\ a \leq b$ and $a \leq \neg c$. — T12.2

Since a is an atom, $a \leq c$

$\therefore\ a \leq c \wedge \neg c = 0$ — T12.2

But contradiction!

$\therefore\ b \wedge \neg c = 0$

$\therefore\ b \leq c$ — L12.1

$b = c = a_1 \vee a_2 \vee \ldots \vee a_k$.

Lemma 12.3 *Let* $(A, \vee, \wedge, \neg)$ *be a* ***finite*** *boolean algebra. Then*

$^{\forall}b \in A-\{0\}, 1 \le {}^{\exists}k \le |A|, 1 \le {}^{\forall}i \le k$, a_i*'s are* ***atoms*** *of* A,

$$b = a_1 \vee a_2 \vee \ldots \vee a_k.$$

is the ***unique*** *way to represent b as* ***join*** *of* ***atoms***.

Proof

Suppose an alternative representation

$b = c_1 \vee c_2 \vee \ldots \vee c_t$. *join of atoms*

$\therefore$ *For* $1 \le {}^{\forall}j \le t$,

$c_j \le b$. c_j*'s are atoms*

$\therefore c_j \wedge b = c_j$.

$c_j \wedge (a_1 \vee a_2 \vee \ldots \vee a_k) = c_j$. *repalce b*

$(c_j \wedge a_1) \vee \ldots \vee (c_j \wedge a_k) = c_j$. $\wedge$ *dist. over* $\vee$.

$\therefore 1 \le {}^{\exists}i \le k$.϶. $c_j \wedge a_i \ne 0$.

Since both c_j *and* a_i *are atoms,* $c_j = a_i$.

$\therefore 1 \le {}^{\forall}j \le t, 1 \le {}^{\exists}i \le k$.϶. $c_j = a_i$.

$\therefore t = k$.

Q.E.D.

Theorem 12.11 *Let $(A, \vee, \wedge, \neg)$ be a **finite** boolean algebra and S be a set of **atoms**. Then $(A, \vee, \wedge, \neg)$ is **isomorphic** to the boolean algebra $(2^S, \cup, \cap, ^-)$.*

Proof

*Let $(A, \leq)$ be a **boolean lattice** and S be set of **atoms**.*

$f: A \to 2^S$

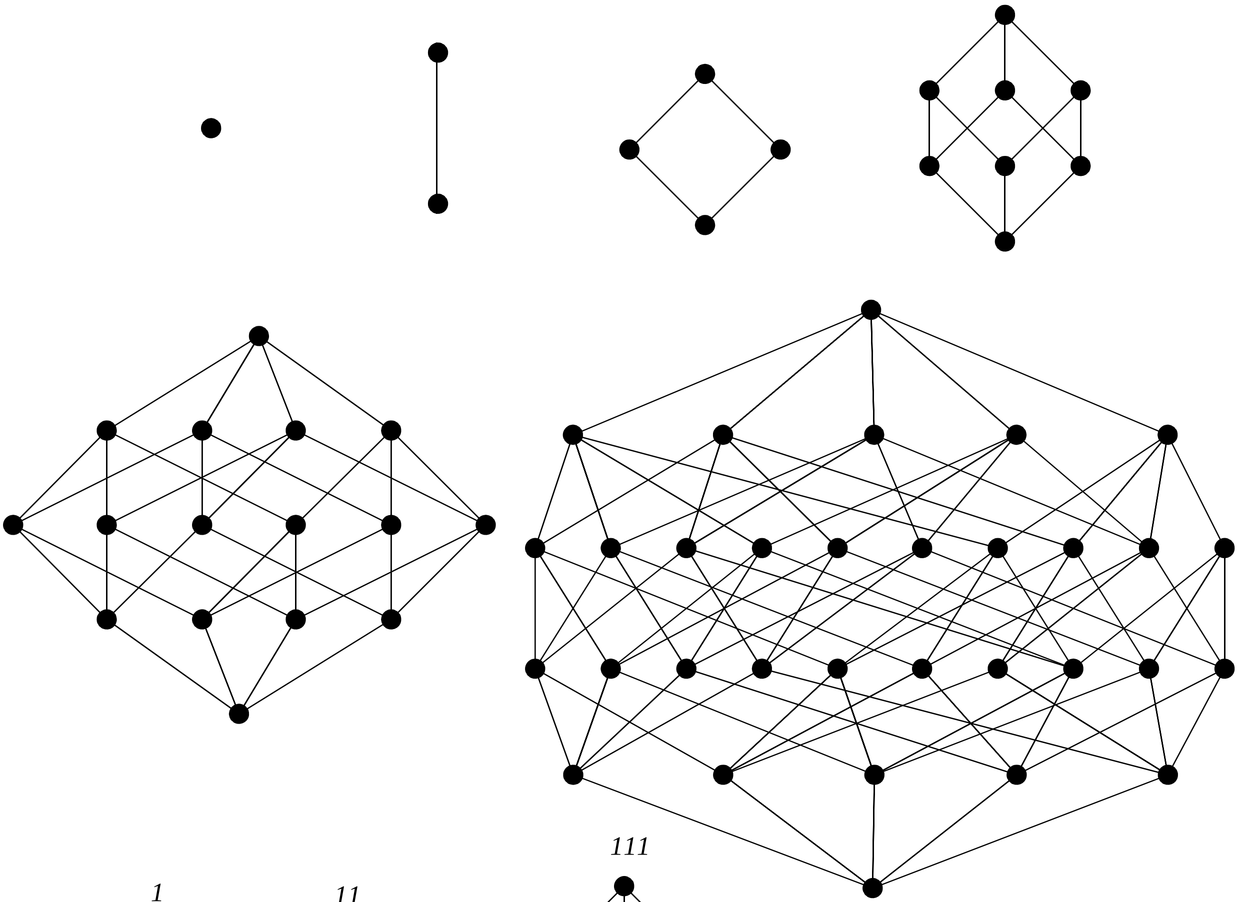
$f(b) = \{a \in S | a \leq b\}$

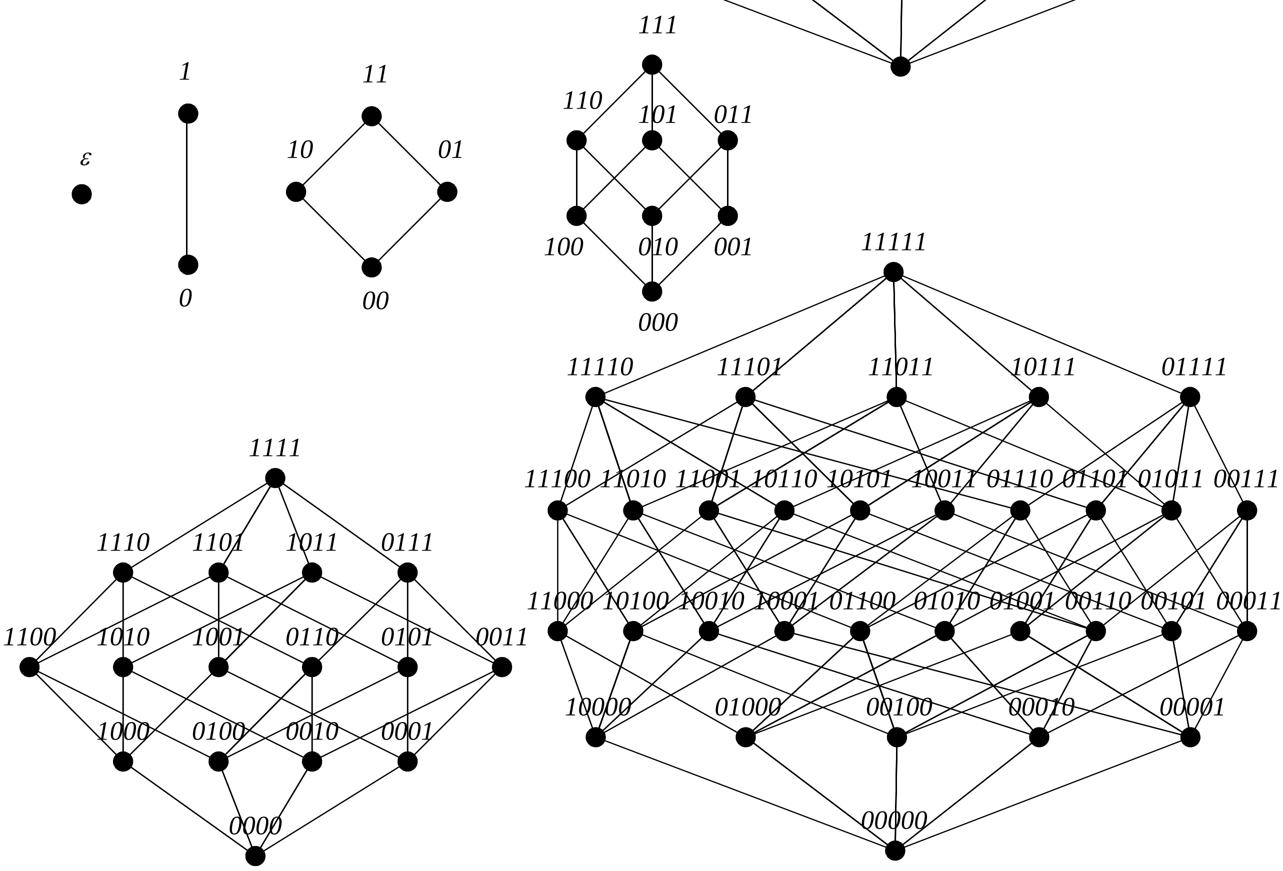
*f is **one-to-one onto**.(Lemma 12.2. 12.3)*

*Thers is a **isomorphism** f from*
*any **finite boolean lattice** $(A, \leq)$ to*
***finite power set latice** $(2^S, \subseteq)$*
*where S is a set of **atoms** of A.*

*Any **finite boolean algebra** $(A, \vee, \wedge, \neg)$*
*is **isomorphic** to the*
***finite power set boolean algebra** $(2^S, \cup, \cap, ^-)$.*
*where S is a set of **atoms** of A.*

*There exists a **unique finite boolean algebra***
of $\mathbf{2^n}$ elements for any $n > 0$, and
*there is **no** other finite boolean algebra.*




ε
1
0
11
10
01
00
111
110
101
011
100
010
001
000
1111
1110
1101
1011
0111
1100
1010
1001
0110
0101
0011
1000
0100
0010
0001
0000
11111
11110
11101
11011
10111
01111
11100
11010
11001
10110
10101
10011
01110
01101
01011
00111
11000
10100
10010
10001
01100
01010
01001
00110
00101
00011
10000
01000
00100
00010
00001
00000

12.7 Boolean Functions and Boolean Expressions

Let $(A, \vee, \wedge, \neg)$ be a boolean algebra.
Consider a function $f: A^n \to A$
    function
        $(|A|^n \times 1)$ table of $|A|$ values.
        ***closed form expression** with n variables.*

Let $(A, \vee, \wedge, \neg)$ be a boolean algebra.
*A **boolean expression** over $(A, \vee, \wedge, \neg)$ is defined*
    *1. $a \in A$ is a **boolean expression**,*
    *2. Any variable name is a **boolean expression**,*
    *3. If e_1 and e_2 are **boolean expressions**, then*
        *$e_1 \vee e_2$, $e_1 \wedge e_2$, $\neg e_1$ are **boolean expressions**.*

***Syntactic grammar** of boolean expressions*

$B \to a$	$a \in A.$	
$\mid\ v$	$v \in V$	*variable name.*
$\mid\ B \vee B \mid B \wedge B \mid \neg B$		

Boolean expression over $(A, \vee, \wedge, \neg)$ is
    *a **language** over $A \cup V \cup \{\vee, \wedge, \neg\}$*
        *where V is a set of **variable** names.*

*A boolean expression that contains n **distinct** variable, is called a boolean expression with n-variables.*

Let $E(x_1, \ldots, x_n)$ be a ***boolean expression*** *of n-variables over an boolean algebra $(A, \vee, \wedge, \neg)$.*

$E(a_1, \ldots, a_n) \in A$ where $a_1, \ldots, a_n \in A$

assignment of values $a_1, \ldots, a_n$

to the variable $x_1, \ldots, x_n$.

Example

$E(x_1, x_2, x_3) = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2) \wedge \neg(x_2 \vee x_3)$

$E(0, 1, 0) = (0 \vee 1) \wedge (1 \vee 0) \wedge \neg(1 \vee 0) = 0$

Two boolean expressions of n variables are ***equivalent****, if* ***every*** *assignment of values to n variables results same values.*

Let $E_1, E_2: A^n \to A$.

$E_1(x_1, \ldots, x_n) = E_2(x_1, \ldots, x_n)$, if and only if,

$\forall (a_1, \ldots, a_n) \in A^n$,

$E_1(a_1, \ldots, a_n) = E_2(a_1, \ldots, a_n)$.

Example

$(x_1 \wedge x_2) \vee (x_1 \wedge \neg x_3) = x_1 \wedge (x_2 \vee \neg x_3)$

Equivalence *of two boolean expressions*

1. Check every $|A|^n$ cases.
2. Properties of boolean algebra.

$f: A^n \to A$ vs $E(x_1, \ldots, x_n)$

*Not every function f has the **equivalent** boolean expression $E(x_1, \ldots, x_n)$*

*$f: A^n \to A$ is called **boolean function**, if it can be specified by a **boolean expression***

*But $\{0, 1\}^n \to \{0, 1\}$ is always a **boolean function***

***Proof** Let $f: \{0, 1\}^n \to \{0, 1\}$.*

Let $(x_1, \ldots, x_n) \in V^n$. Then

*$\tilde{x}_1 \wedge \ldots \wedge \tilde{x}_n$ is a **minterm**, if $\tilde{x}_i = x_i$ or $\tilde{x}_i = \neg x_i$.*

*$\tilde{x}_1 \vee \ldots \vee \tilde{x}_n$ is a **maxterm**, if $\tilde{x}_i = x_i$ or $\tilde{x}_i = \neg x_i$.*

There are 2^n minterms and maxterms, resp.

***Disjunctions**($\vee$) of minterms whose f value is 1 is **equivalent** to f.*

$$\vee_{f(x_1 x, \ldots, x_n)=1} \tilde{x}_1 \wedge \ldots \wedge \tilde{x}_n$$

$$= (\tilde{x}_1 \wedge \ldots \wedge \tilde{x}_n) \vee \ldots \vee (\tilde{z}_1 \wedge \ldots \wedge \tilde{z}_n)$$

disjunctive normal form

***Conjunctions**($\wedge$) of maxterms whose f value is 0 is **equivalent** to f.*

$$\wedge_{f(x_1 x, \ldots, x_n)=0} \tilde{x}_1 \vee \ldots \vee \tilde{x}_n$$

$$= (\tilde{x}_1 \vee \ldots \vee \tilde{x}_n) \wedge \ldots \wedge (\tilde{z}_1 \vee \ldots \vee \tilde{z}_n)$$

conjunctive normal form

12.8 Propositional Calculus

*A **proposition** is a statement that may be either be **true**(T) or **false**(F)*

Consider a boolean lattice, $(\{F, T\}, \{F \le T\})$

*A **bolean algebra** $(\{F, T\}, \vee, \wedge, \neg)$ defined by a boolean lattice $(\{F, T\}, \{F \le T\})$.*

disjunction	$\vee$
conjunction	$\wedge$
negation	$\neg$

tautology	$\Leftrightarrow$ T(1)	
contradiction	$\Leftrightarrow$ F(0)	
disjunction	$\Leftrightarrow$ ***join***($\vee$, lub)	
conjunction	$\Leftrightarrow$ ***meet***($\wedge$, glb)	
negation	$\Leftrightarrow$ ***complement***($\neg$)	
condition	$\Leftrightarrow$ ***complement meet***	
$p \to q$	$\neg p \vee q$	$p \le q.$
biconditional	$\Leftrightarrow$ ***complement meet join***	
$p \leftrightarrow q$	$(p \wedge q) \vee (\neg p \wedge \neg q)$	$p = q.$
	$\equiv (\neg p \vee q) \wedge (p \vee \neg q)$	

12.9 Digital Network

$OR_n: \{0, 1\}^n \to \{0, 1\}$ *n-input OR gate*

$OR_n(x_1, \ldots, x_n) = x_1 \vee \ldots \vee x_n.$

$AND_n: \{0, 1\}^n \to \{0, 1\}$ *n-input AND gate*

$AND_n(x_1, \ldots, x_n) = x_1 \wedge \ldots \wedge x_n$.

$NOT_1: \{0, 1\} \to \{0, 1\}$ *1-input NOT gate*

$NOT_1(x) = \neg x$.

Let $f: \{0, 1\}^n \to \{0, 1\}$ *be a boolean function. Then f is equivalent to the composition of three gates.*

Proof

Let ***disjuntive*** *normal form has* $d(\leq 2^n)$ ***minterms***

$1 \leq {}^\forall j \leq d$, *we define minterm* $i_j = AND_n(\hat{x}_1, \ldots, \hat{x}_n)$

where $1 \leq {}^\forall k \leq n$, $\hat{x}_k = x_k$, *if* $\tilde{x}_k = x_k$; $NOT_1(x_k)$ *if* $\tilde{x}_k = \neg x_k$.

Then f is equivalent to $OR_d(i_1, \ldots, i_m)$.

Let ***conjuntive*** *normal form has* $c(\leq 2^n)$ ***maxterms***

$1 \leq {}^\forall j \leq c$, *we define maxterm* $t_j = OR_n(\hat{x}_1, \ldots, \hat{x}_n)$

where $1 \leq {}^\forall k \leq n$, $\hat{x}_k = x_k$, *if* $\tilde{x}_k = x_k$; $NOT_1(x_k)$ *if* $\tilde{x}_k = \neg x_k$.

Then f is equivalent to $AND_c(i_1, \ldots, i_m)$.

Furthermore,

f is equivalent to the composition of ***three*** *gates,* OR_2, AND_2, *and* NOT_1.

f is equivalent to the composition of ***two*** *gates,* NOR_2, $NAND_2$.

algebraic system
$(A, \oplus)$

poset
$(A, \leq)$

semigroup
$(A, \oplus)$

monoid
$(A, \oplus, e)$

group
$(A, \oplus, \oplus^{-1}, e)$

ring
$(A, \oplus, e, \oplus^{-1}, \otimes)$

field
$(A, \oplus, \oplus^{-1}, \otimes, \otimes^{-1}, 0, 1)$

lattice
$(A, \leq, \vee, \wedge)$

boolean algebra
$(A, \leq, \vee, \wedge, \neg, 0, 1)$

finite boolean algebra
(power set lattice)
$(2^S, \subseteq, \cup, \cap, ^-, \emptyset, S)$

predicate calculus
$(\{f, t\}, \leq, \vee, \wedge, \neg, f, t)$

XI 9/1(화) 제1강 강의소개 와 ~~이산수학 복습~~. 1/1

본강의는 모국어로 이루어진다. home page adamant.kaist.ac.kr/cs322

수학은 언어이다.

공학, 자연과학 — 수학으로 대화한다

인문, 사회과학 — 모국어

전산학 — 프로그래밍 언어(?)

Dijkstra
A humble programmer
P.L. for human-being vs. machine.

최광무
P.L. for human-being vs mathematics

피타고라스의 슬픔 $\sqrt{2}$

18xx Cantor's diagonal argument.
191x Russel's Paradox $R = \{x \in R \mid x \notin x\}$
1939? Gödel's Incompleness Theorem
? Halting Problem 증명 ⇔ 정리 ⇔ 프로그램
? 이발사의 고민 이글은 개똥이야
? 거짓말장이 "내가 했다" 개똥이가 있는데

→ ~~deci~~ Denial of Self-recursion
자기자신에 대한 부정

finite representation of infinite sets.
"infinite wisdom"
↳ Recursion (Basis) … 수학적 귀납법

좋은책: 이희재 "번역의 탄생"

1) (1)

제2강 (9/13 木) 이산수학 복습.

set

1. 원소나열법

$\{1, 2, \dots, n\} = \mathbb{N}_n$

$\{1, 2, \dots\} = \mathbb{N}_0$

$\{0, 1, \dots\} = \mathbb{N}_1$

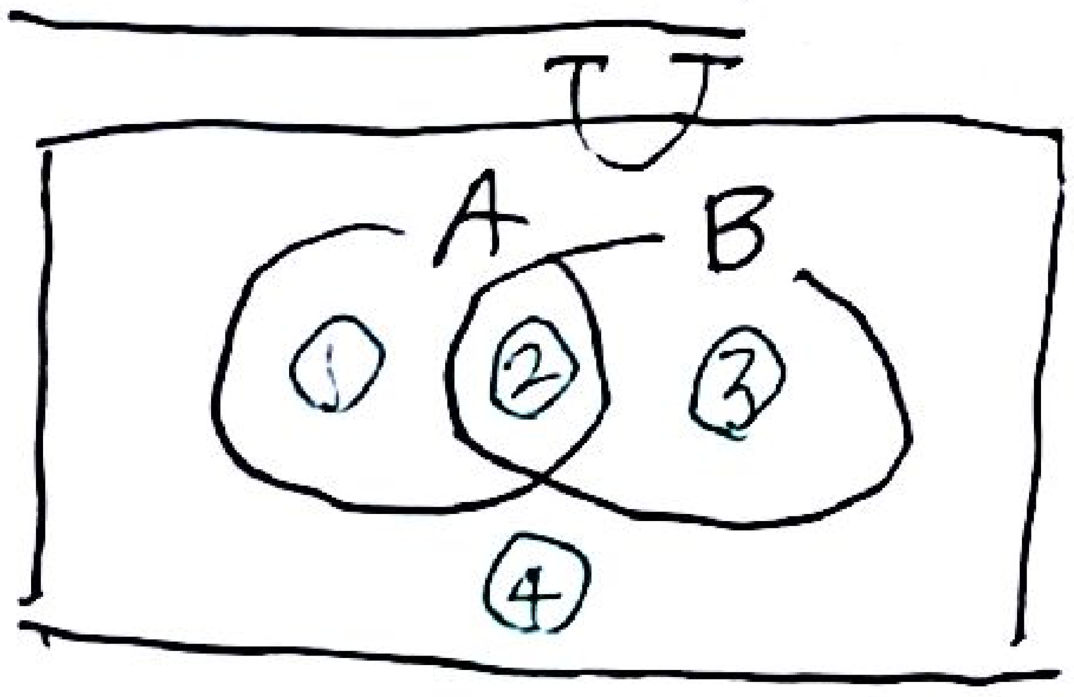
2. 조건제시법

$A = \{x \in U \mid p(x)\}$

predicate cond. statement

조건명제 $\xrightarrow{\forall, \exists}$ 명제

for all . there exist...

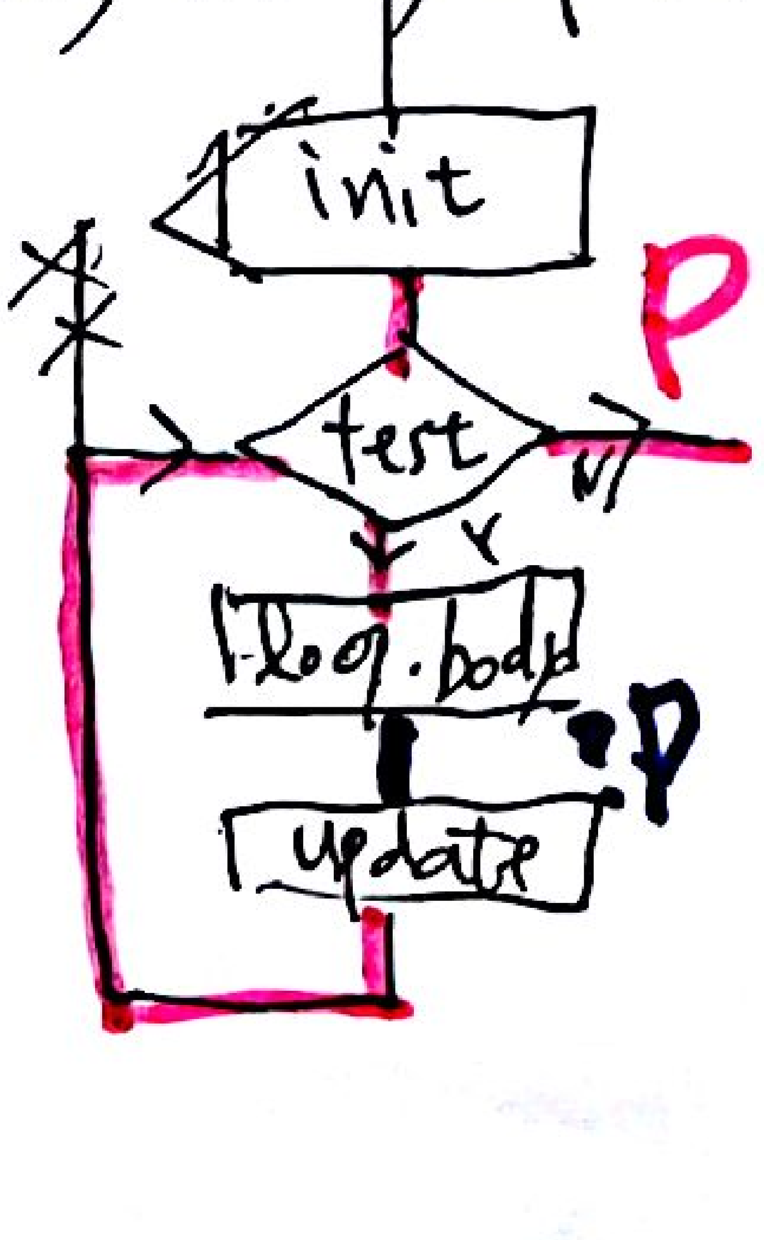


partition $\longleftrightarrow$ equivalence relation

recursion $\longleftrightarrow$ loop < invariant cond. P / termination

C

for(init; test; update) loop_body



제3강 9/8(土) 이산수학 II.

Cartician Product of two sets A and B.

$A \times B \triangleq \{(a,b) \mid a \in A, b \in B\}$

순서쌍

ordered pair

A binary relation (이진 관계) from a set A to B.

R

$R \subseteq A \times B$. $(a,b) \in R$ aRb

a set of ordered pairs

binary relation from A to B

Ex) $\leq \subseteq \mathbb{N} \times \mathbb{N}$

$R: A \times B \to \{t, f\}$

$2^A = \{B \mid B \subseteq A\}$ Ex. $A = \{a_1, a_2, \ldots, a_n\}$

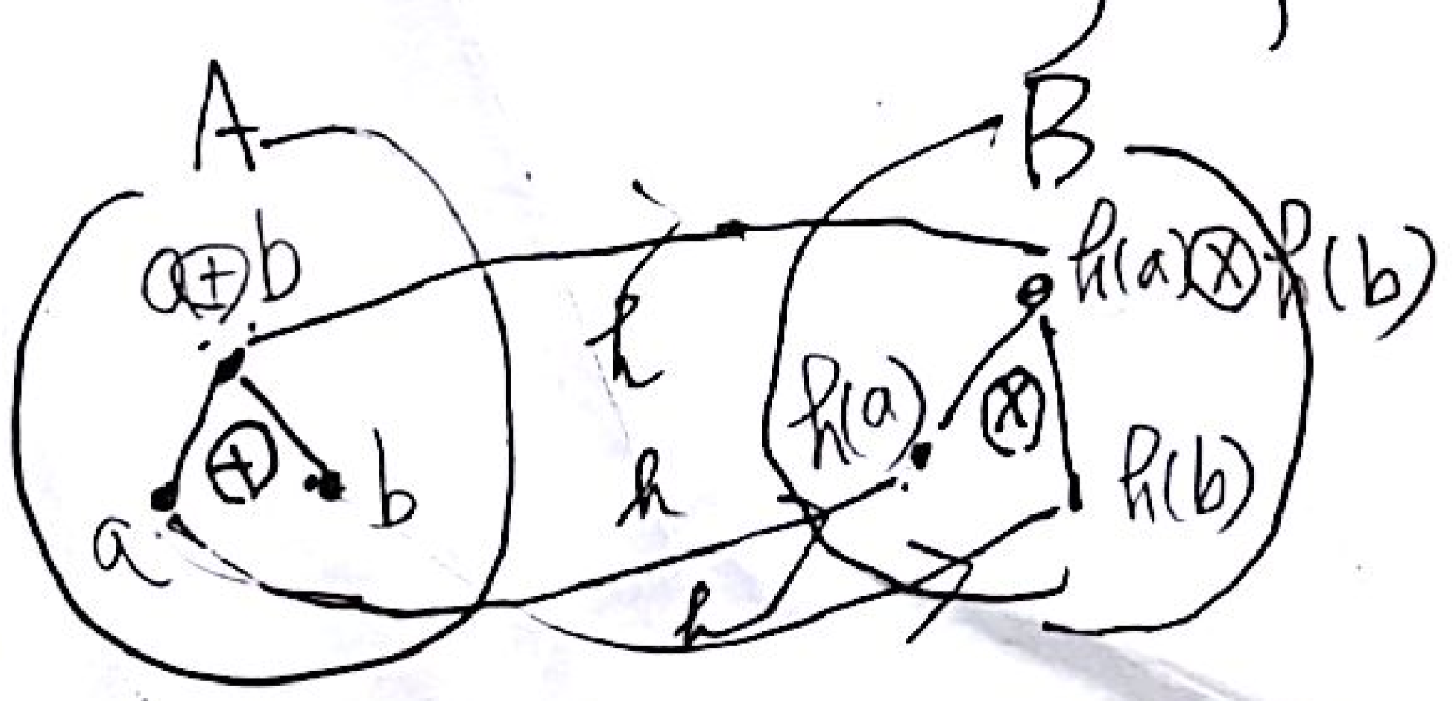
$|2^A| = ?$

$|1\ 1\ 0 \cdots 0| = n$

$\updownarrow$

$\{a_1, a_2\}$

$\oplus : A \times A \to A$ a binary operation (연산) on A.



(9/10 목)

제 4강 set isomorphism, ~~infin~~ infinite sets, 4 Terminologies

~~If~~ $|A| = |B|$ ~~then~~ $A \overset{?}{=} B$ X

iff $A \cong_f B$

정의, characterization, ~~성질~~ property

$\Leftrightarrow$ $\Rightarrow$

$\mathbb{N} \times \mathbb{N} \xleftrightarrow{f} \mathbb{N}$

$(0,0), (0,1), (0,2), \ldots$ sum first, lexicographic order

$(1,0), (1,1), (1,2), \ldots$ (사전식 순서)

$(2,0), (2,1), (2,2), \ldots$ 군수열

Def Let Σ be a set of symbols
vocabulary, alphabet,

$\Sigma^0 \overset{\Delta}{=}_B \{\varepsilon\}$

$\Sigma^{n+1} \overset{\Delta}{=}_R \Sigma \cdot \Sigma^n \quad (n \geq 0)$ vs $\Sigma^n = \Sigma \cdot \Sigma^{n-1} \quad (n \geq 1)$

$\Sigma^* \overset{\Delta}{=} \bigcup_{i \in \mathbb{N}_0} \Sigma^i$

Def empty string ε ... identity element for concatenation op. $\cdot$,

$|\varepsilon| = 0$, $\forall x \in \Sigma^*$ $\varepsilon \cdot x = x \cdot \varepsilon = x$

$(\Sigma^*, \cdot, \varepsilon)$ is a monoid. where $\cdot : \Sigma^* \times \Sigma^* \to \Sigma^*$

Def $B^A \overset{def}{=}$ ~~f:~~

$$\{f \in A \times B \mid f : A \to B\}$$

Example $B^{\{1,2,\dots,n\}}$ ... string of length n over $\{a,b,\dots,z\}$

$$f : \{1,2,\dots,n\} \to \{a,b,c,\dots,z\} \to$$

Thm $|B^A| = |B|^{|A|}$

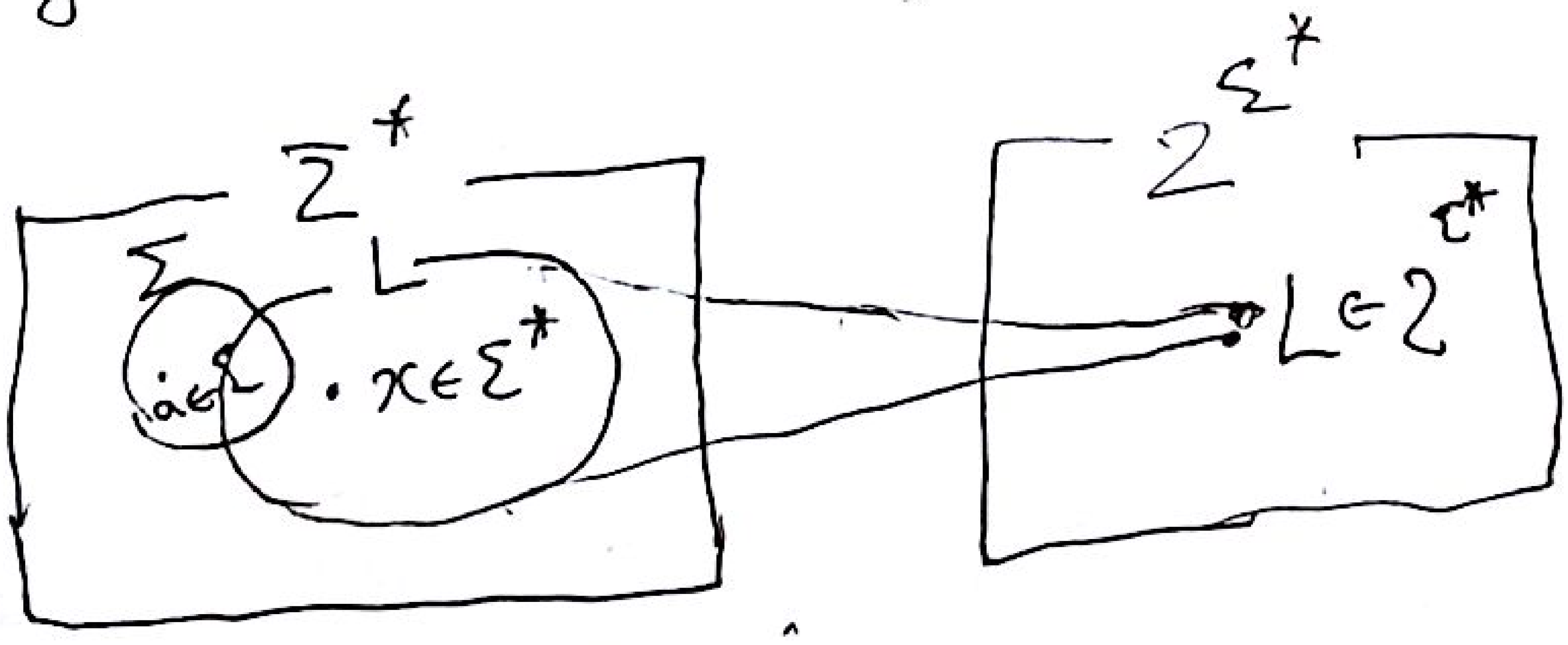
string ... 문자열　　symbol ... 문자

a sequence of symbols

문자가 줄서면 문자열이 된다

a set of strings　　　a set of symbols

language.　　　　　vocabulary (어휘)



Four Terminologies

집합

sequencing

$a\in\Sigma$ (symbol, 문자)	Σ (vocabulary, 어휘)
$x\in\Sigma^*$ (string, 문자열)	$L\subseteq\Sigma^*$ (language, 언어)

제15강 9/15(화) Uncountable & Deterministic Finite (state) Automata (DFA).

$|\Sigma^*| = |\mathbb{N}|$

Cantor's Diagonal argument.

$\{0,1\}^{\mathbb{N}}$ $\quad f: \mathbb{N} \to \{0, 1\}$ infinite binary strings.

$\mathcal{P}(\mathbb{N}) = 2^{\mathbb{N}} = \{B \mid B \subseteq \mathbb{N}\}$

$(1 0 1 0 0 1 0 0 0 \ldots) \overset{1:1}{\underset{\text{onto}}{\longleftrightarrow}} \{0, 2, 5\}$

$|\{0,1\}^{\mathbb{N}}| = |2^{\mathbb{N}}|$ $\quad 3^{\mathbb{N}}$? $= \{0,1,2\}^{\mathbb{N}}$

Assume $2^{\mathbb{N}}$ is countable

$2^{\mathbb{N}} = \{(b_{00}, b_{01}, b_{02}, \ldots, b_{0i}, \ldots)$ $\quad \forall i,j \in \mathbb{N} \; b_{ij} \in \{0, 1\}$

$(b_{10}, b_{11}, b_{12}, \ldots, b_{2i}, \ldots)$

$\ldots$

$(b_{i0}, b_{i1}, b_{i2}, \ldots, b_{ii}, \ldots)$

$\ldots\}$

Consider $\bar{a} = (\bar{b}_{00}, \bar{b}_{11}, \bar{b}_{22}, \ldots, \bar{b}_{ii}, \ldots)$ $\quad \bar{b}_i = \begin{cases} 1 & \text{if } b_{ii} = 0 \\ 0 & \text{if } b_{ii} = 1. \end{cases}$

$\bar{a} \notin 2^{\mathbb{N}} \neq \{(\;)(\;)\ldots(\;)\ldots\}$

$\bar{a} \in 2^{\mathbb{N}}$ Contradiction

Uncountable !

Denial of Self recursion

$\longleftrightarrow$ Complement of diagonal !

Chap. 2 Finite Automata

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

$\delta : Q \times \Sigma \to Q$

$q_0 \in Q$, $F \subseteq Q$

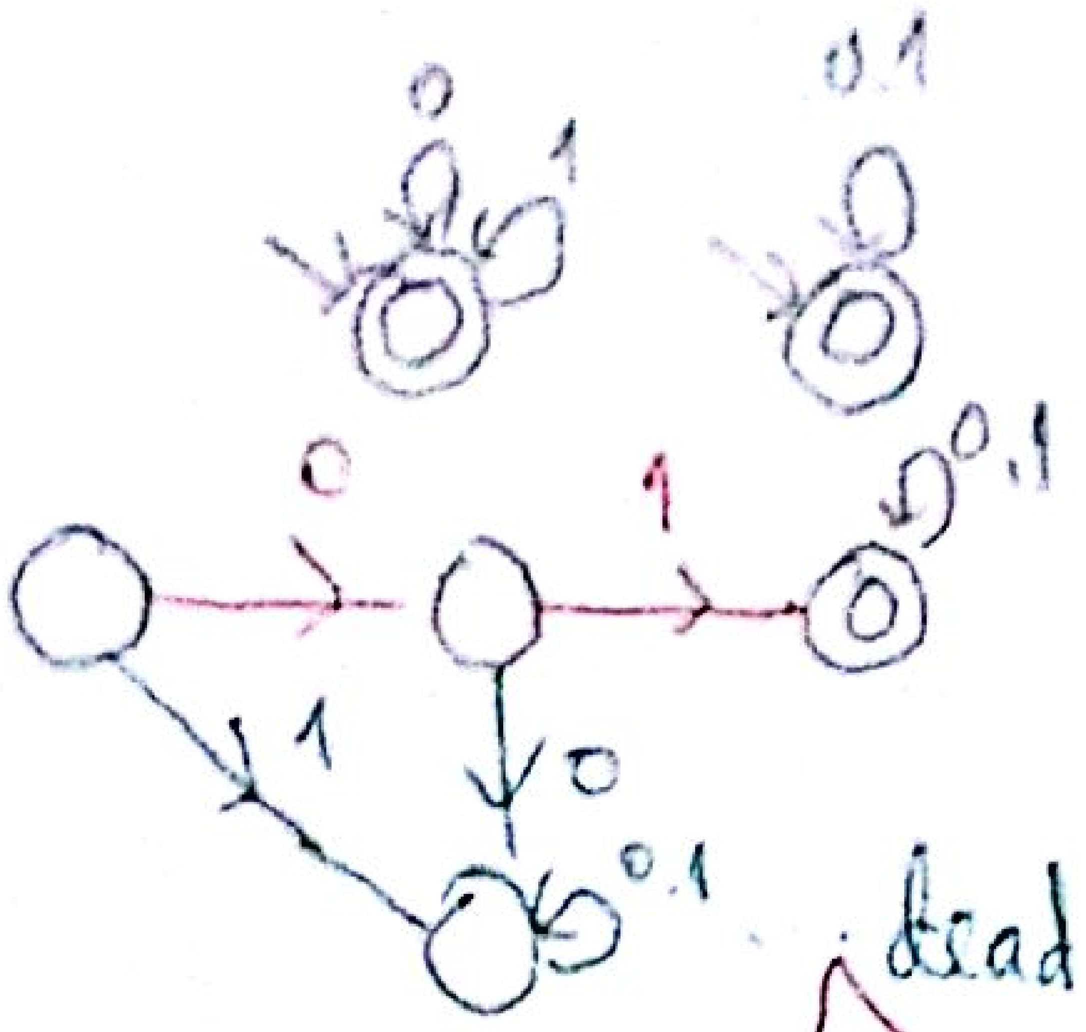
$\in \{0,1\}^*$

Ex. 2.1 $\Sigma = \{0, 1\}$, $L = \{x01y \mid x, y \in \Sigma^*\}$

제6강 DFA - Extension of DFA (9/11)

K's Book
Ex 2.6 $(0+1)^*$

Prefix 01



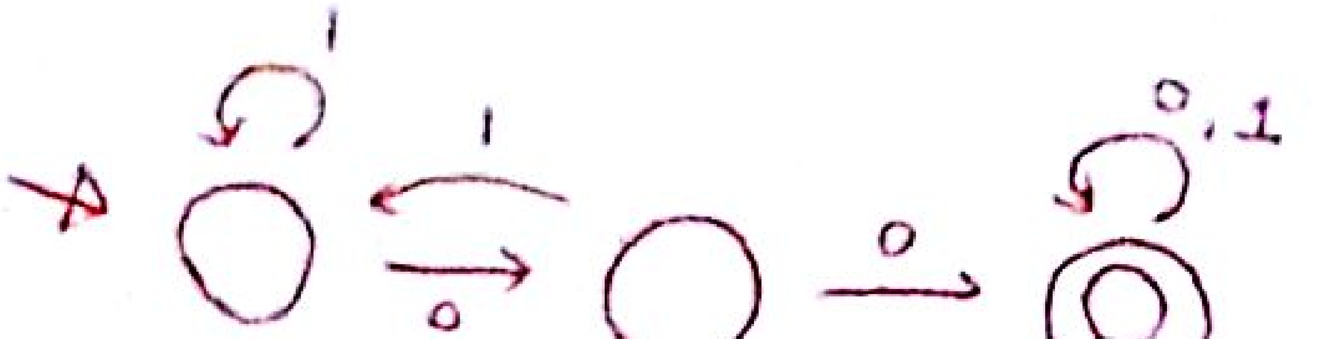
$01(0+1)^*$

dead state → $Q \times \Sigma \to Q$

Eliminate

partial ftn.

Ex 2.8

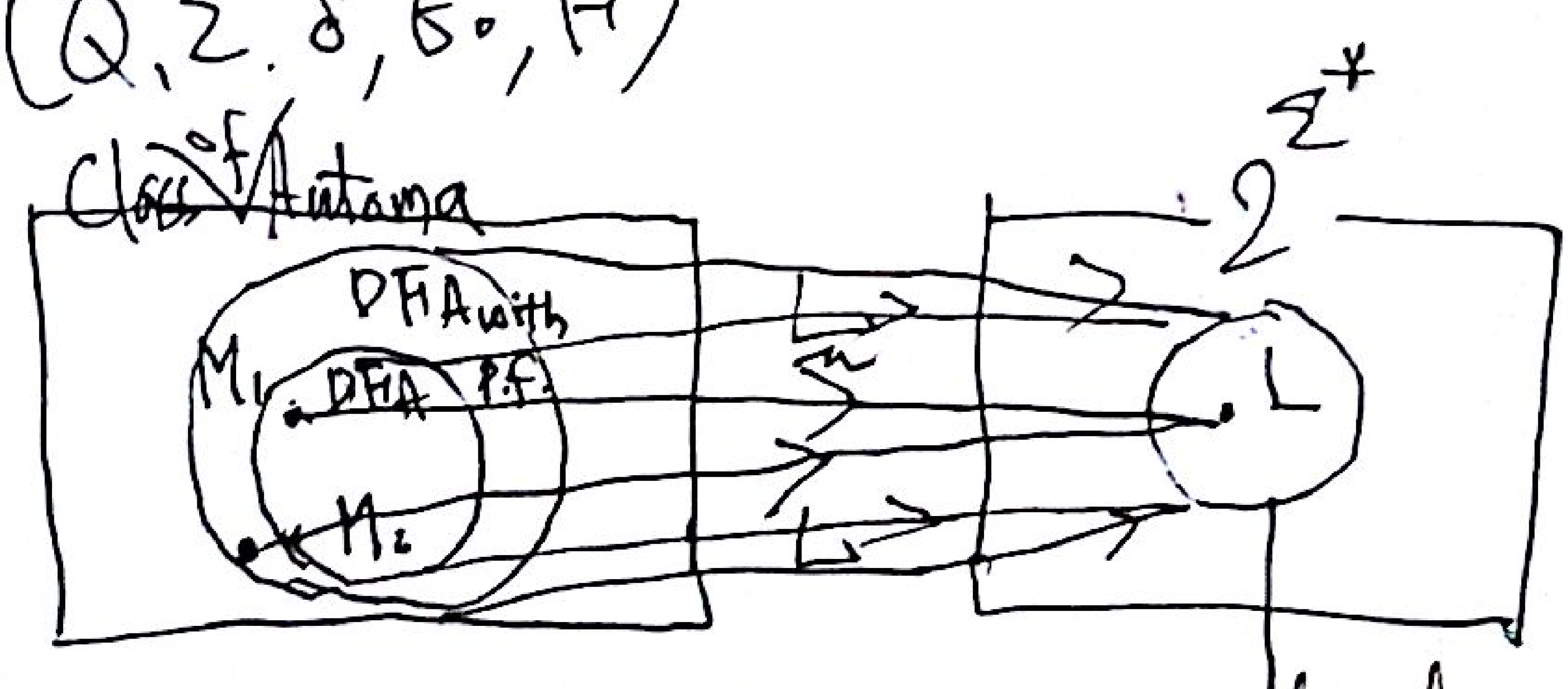


$\delta : Q \times \Sigma \to Q$ $\quad \Sigma \subsetneq \Sigma^* \therefore \delta \subsetneq \delta^*$

$\delta^* : Q \times \Sigma^* \to Q$

$L(D) = \{x \in \Sigma^* \mid \delta^*(q_0, x) \in F\}$ Well-defined well-formed

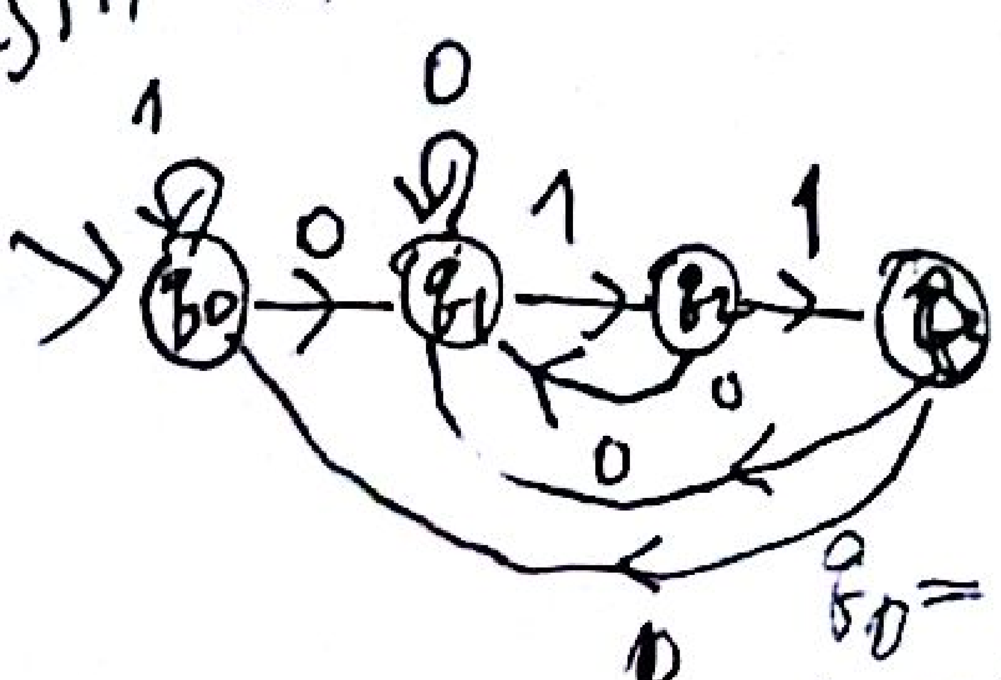
$D = (Q, \Sigma, \delta, q_0, F)$

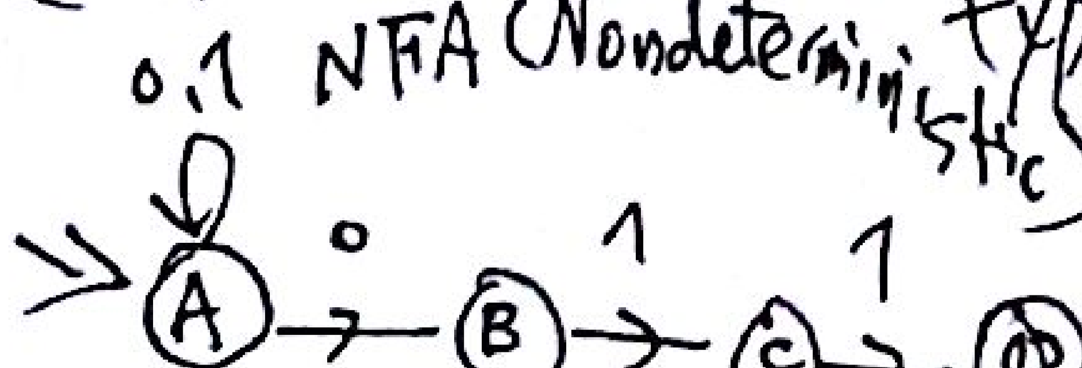


Regular languages type-3 class

Suffix 011 $\quad (0+1)^*011$

0,1 NFA (Nondeterministic)



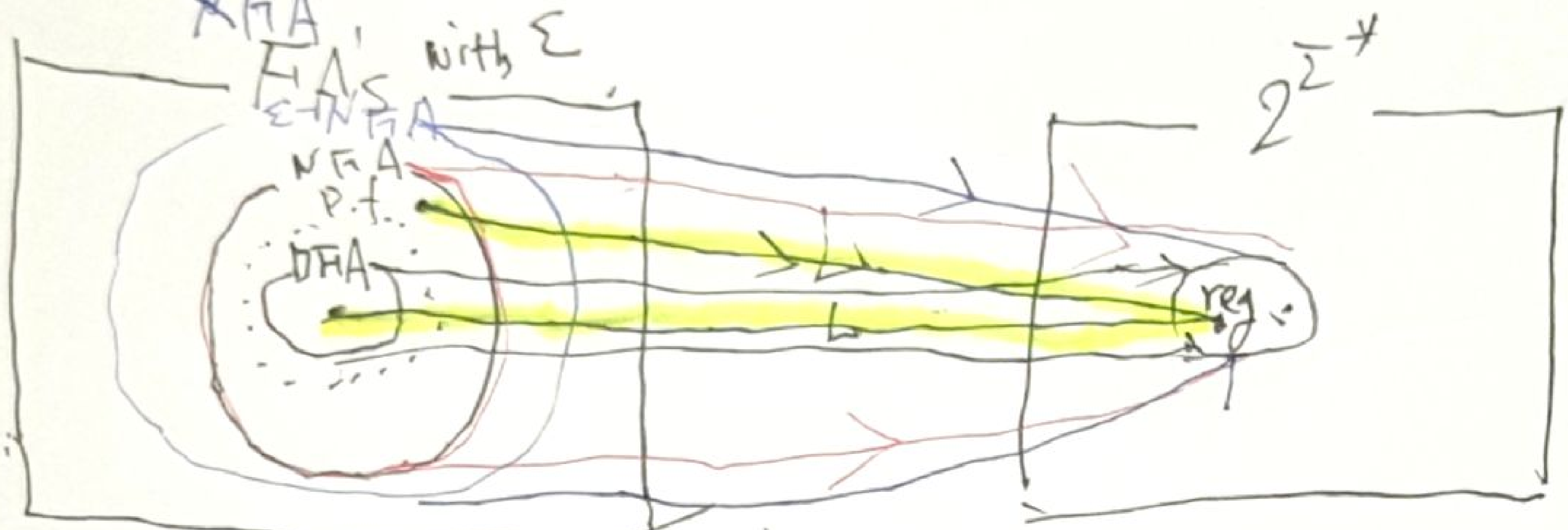


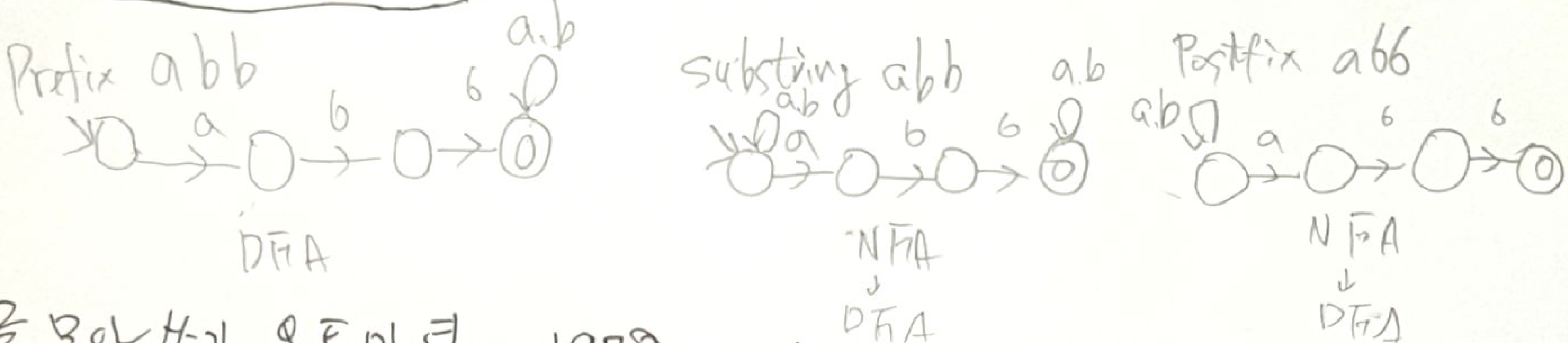
	0	1
$q_0 = \{A\}$	$\{A,B\}$	$\{A\}$
$q_1 = \{A,B\}$	$\{A,B\}$	$\{A,C\}$
$q_2 = \{A,C\}$	$\{A,B\}$	$\{A,D\}$
$q_3 = \{A,D\}$	$\{A,B\}$	$\{A\}$

	0	1
A	$\{A,B\}$	$\{A\}$
B	$\emptyset$	$\{C\}$
C	$\emptyset$	$\{D\}$
D	$\emptyset$	$\emptyset$

1)

제7강 9/22(화) 한글모아쓰기오토마타, ε-NFA





한글모아쓰기 오토마타 1978. 2. 한국과학원 석사논문

FA with output $(Q, \Sigma, \Pi, \delta, \lambda, q_0)$

Q, Σ, δ, q_0 는 DFA나 NFA와 같다

Π : output vocabulary

$\lambda_{Mo}: Q \to \Pi$

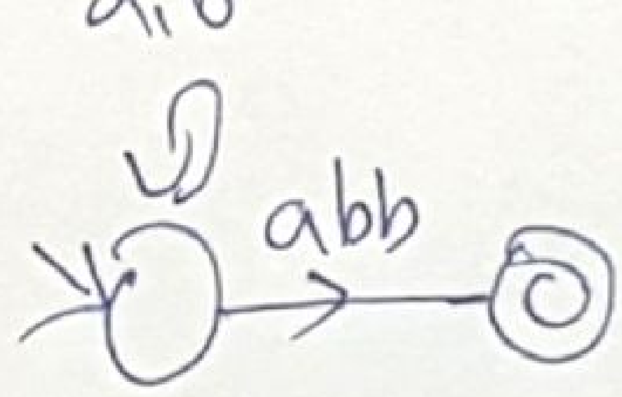
$\lambda_{Me}: Q \times \Sigma \to \Pi$

2.5 FA with ε-moves

$\delta: Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$

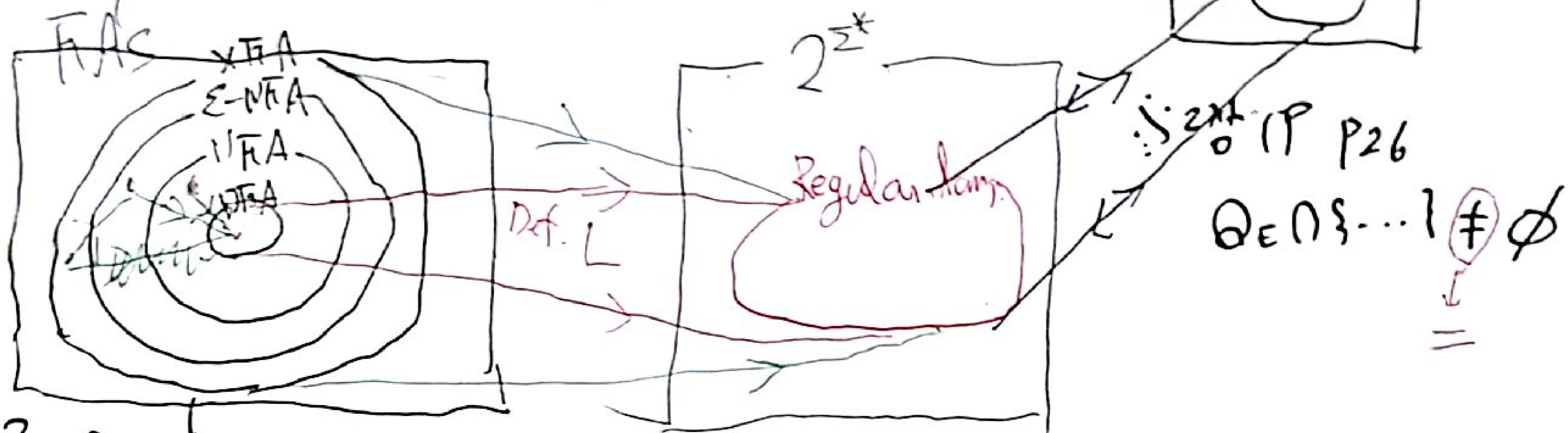
2.A XFA

$\delta: Q \times \Sigma^* \to 2^Q$



제 8강 9/24(木) Regular expression (Chap. 3)

Review



Chap 3. Regular expressions & Regular Language

(정규식) (정규언어)

$boy^R = yob$

3.1. R.E over Σ

B1. $\varepsilon \longrightarrow \{\varepsilon\}$

$\phi \longrightarrow \{\}$ or ϕ

B2. $a \in \Sigma \longrightarrow \{a\}$

R1. $E, F \Rightarrow E+F \longrightarrow L(E) \cup L(F)$

R2 $E, F \Rightarrow E \cdot F \longrightarrow L(E) \cdot L(F)$

R3 $E \Rightarrow E^* \longrightarrow (L(E))^*$

R4 $E \Rightarrow (E) \longrightarrow L(E)$

Ex) $\Sigma = \{0, 1\}$

$$\underbrace{(\varepsilon + \phi + 1 + \phi + 1)^*}_{\{\varepsilon, 1\}^*} \cdot \underbrace{\phi}_{\{\}}$$

$= \{1\}^*$

$= 1^*$

CFG - Chap 5

식 = 문자 | 수 | 식 + 식 | 식 − 식 | (식)

| 식 * 식 | 식 ÷ 식 | 식 / 식 | 식 mod 식

... | 식^식 | sin 식

※ palindrome (chap 5)

generalized math. induction

$P(n) \Rightarrow P(n+k)$

k개의 Bases 가 필요함.

Equivalence of R.E's

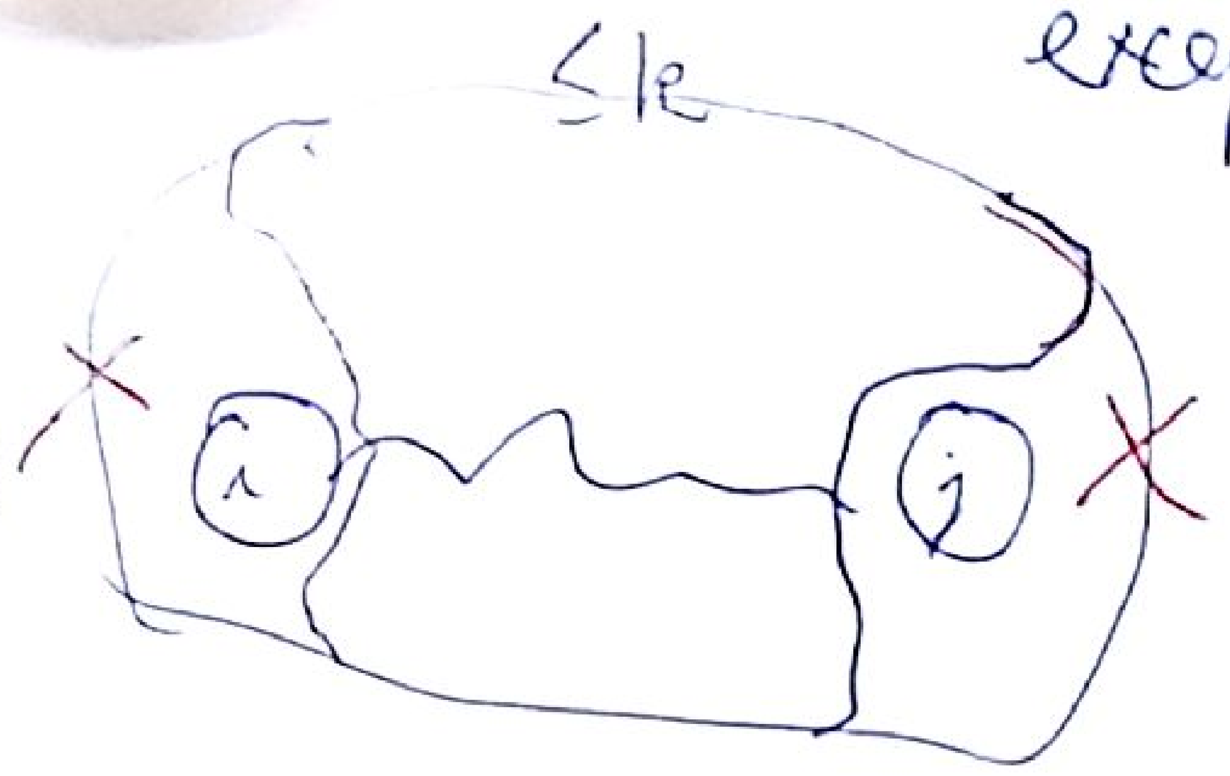
Let R and S be R.E's. If $L(R) = L(S)$, then

$R = S$

2)

DFA ⟹ R.E.

방법1 R_{ij}^{k} (R_{ij})



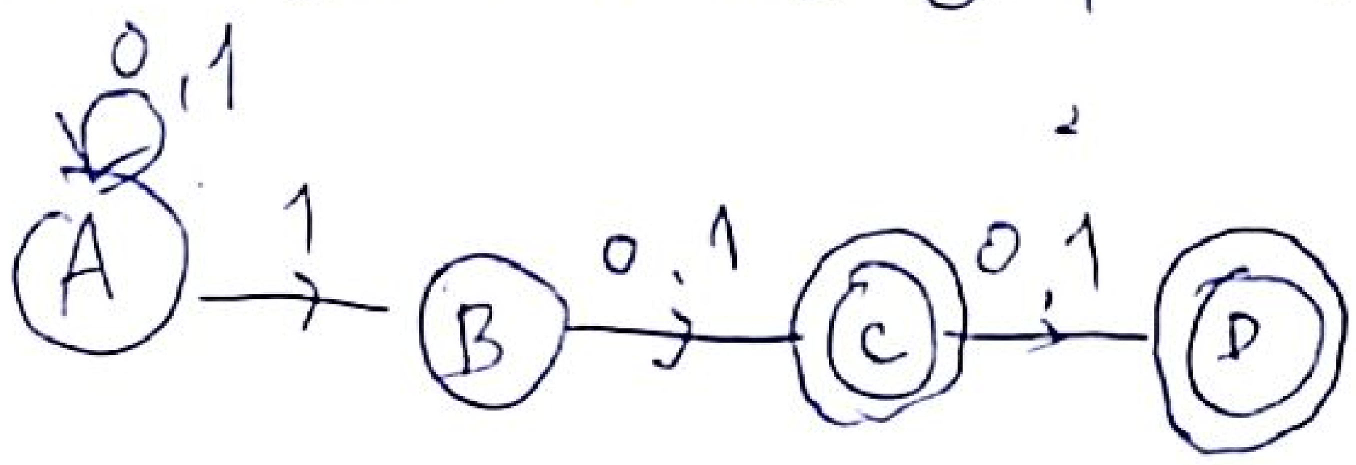
except i and k.

DFA ⟹ RE

Basis

Inducta

방법2. 연립방정식을 푼다. FA ⟹ RE



RE ⟹ FA.

1)

2)

★ 10/1 (목) 제9강 FA와 R.E.

한글 모아쓰기 오토마타 P#1

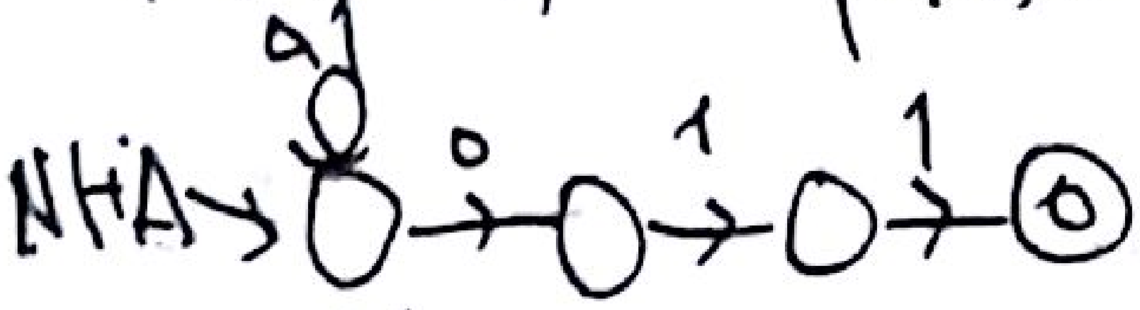
RE ⇒ FA Thm 3.7 Easy to program

FA ⇒ RE Thm 3.4 손으로 풀기 어렵다 HW#3.3

연립방정식

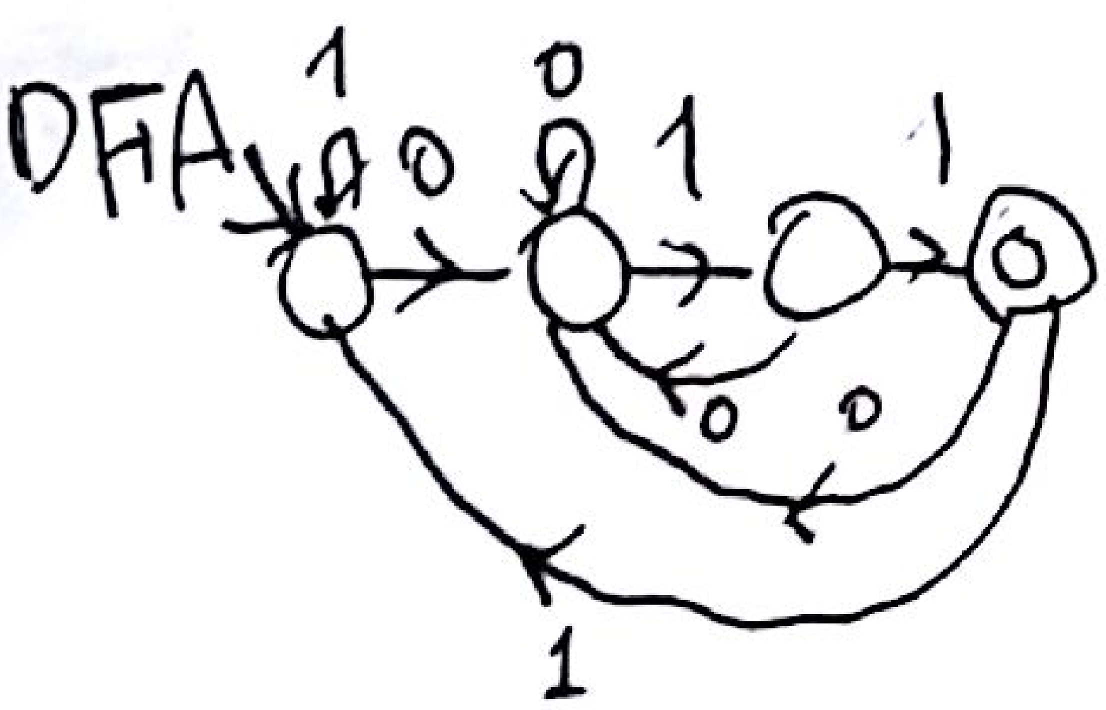
$\boxed{A=\alpha A+\beta \Leftrightarrow A=\alpha^*\beta}$

$(0|1)^*011$ postfix 011



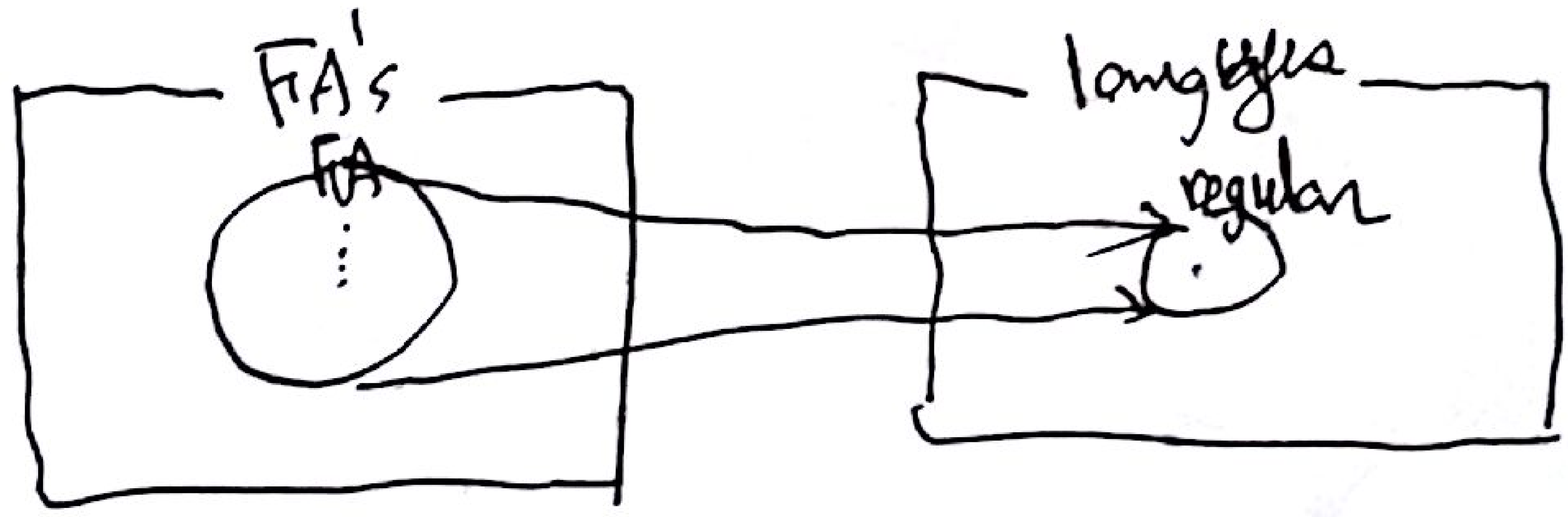
$$A=(0+1)A+0B=(0+1)A+011=(0+1)^*011$$
$$B=1C=11$$
$$C=1D=1$$
$$D=\varepsilon$$

$$A=(0+1)A+011D$$
$$D=\varepsilon$$



$$A=1A+0B$$
$$B=0B+1C$$
$$C=0B+1D$$
$$D=0B+1A+\varepsilon$$

한글 강의 노트 C3.2

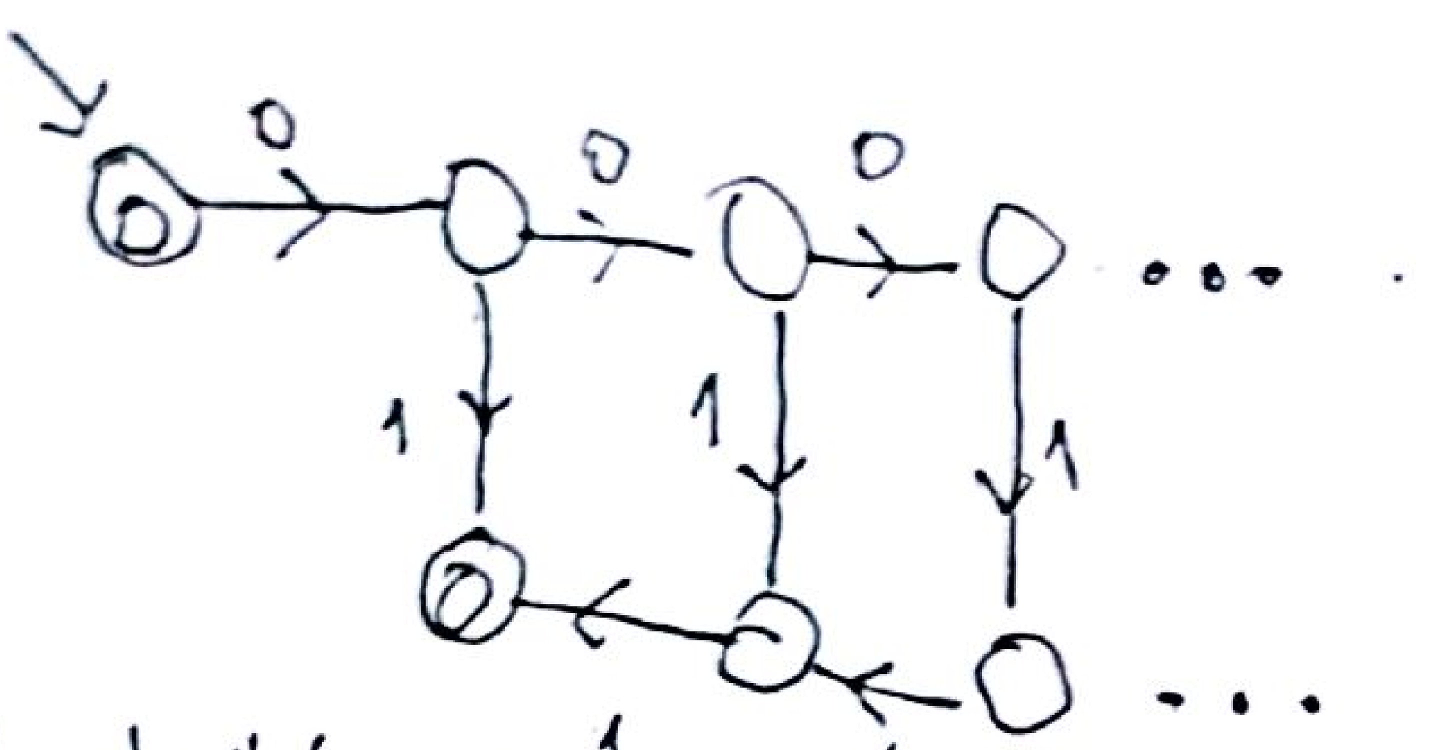


Chap. 7. Properties of Regular Languages.

$L = \{0^n 1^n \mid n \geq 0\}$

$= \{\varepsilon, 01, 0011, 000111, \cdots\} \subset L(0^* 1^*)$

FA



infinite state automata

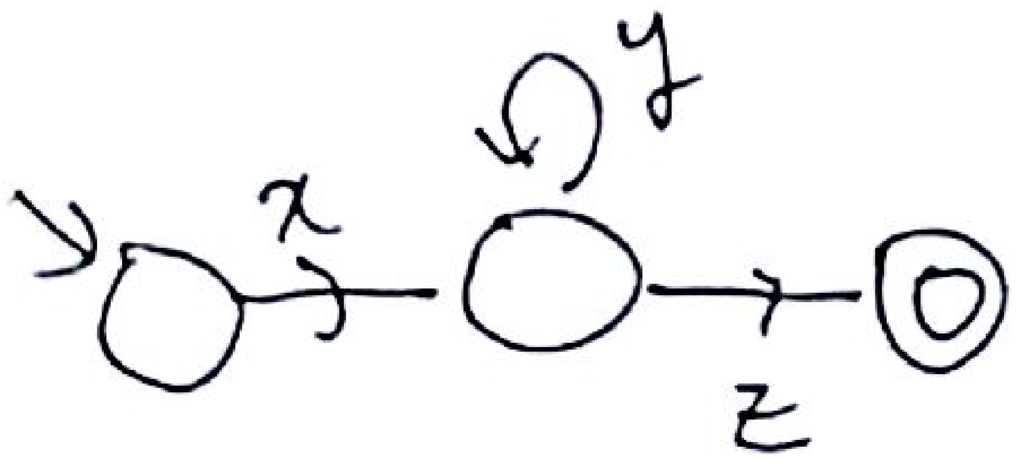
~~FA~~ !

3가지 기계 (automata)

- FA ... no memory except state!
- FA + stack (memory) ... Pushdown Automata
- FA + memory (tape) ... Turing Machine.

Pumping Lemma for 정규언어.

n-states ~~…~~ m-length string $m \geq n$? $>$

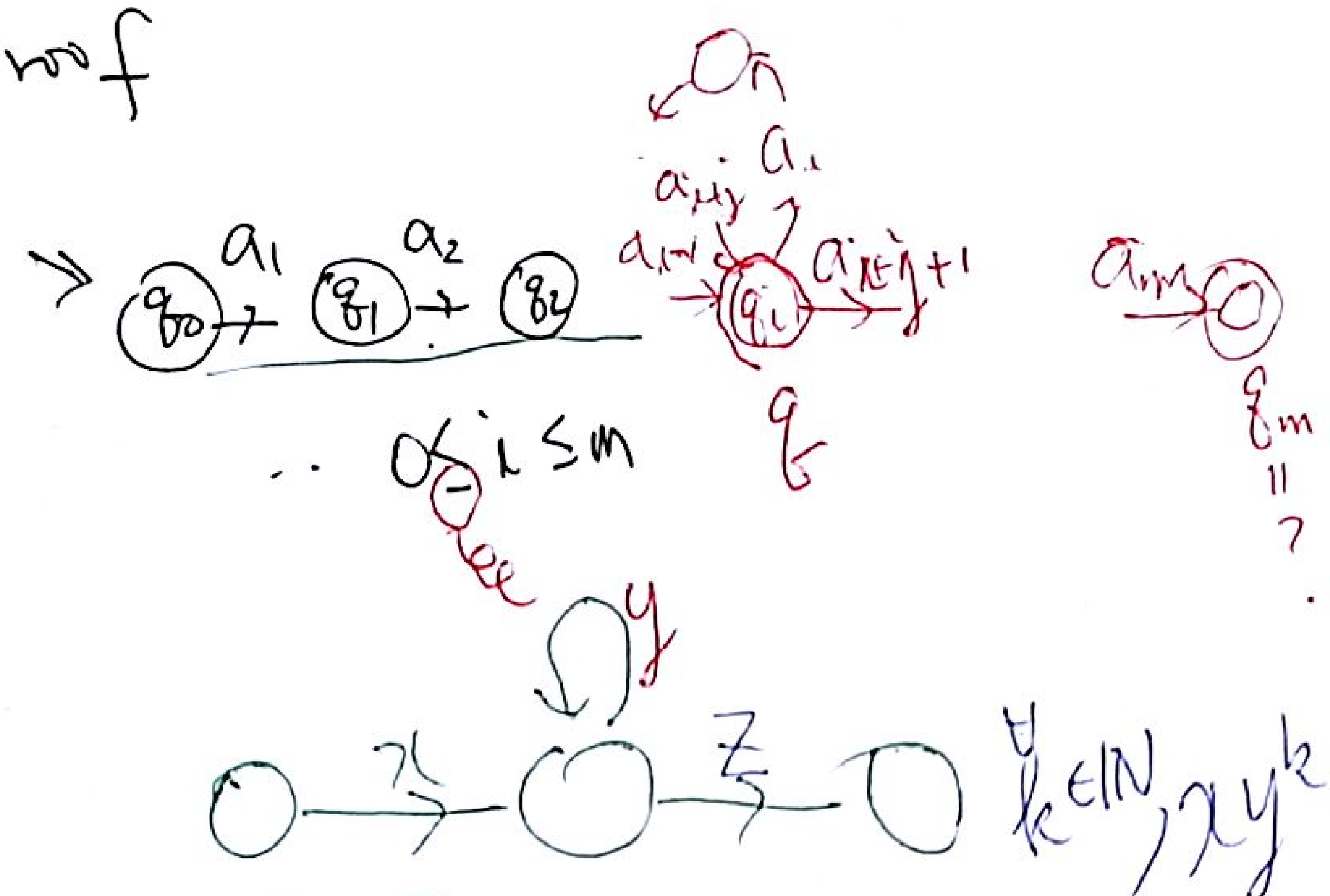


xy^*z

1) 제10강 10/6(木) Pumping lemma 와 Non-regular lang. 1)

Thm 4.1

proof



$\forall k \in \mathbb{N}, xy^k z \in L$

대우명제: $(\forall) \exists (\forall) \exists$ $\notin L \Rightarrow L$ is not regular

변수로

4.2 closure properties of regular languages

$L \cup M$, $L \cdot M$, L^*

$L \cap M$ < De Morgan's Law / product construction

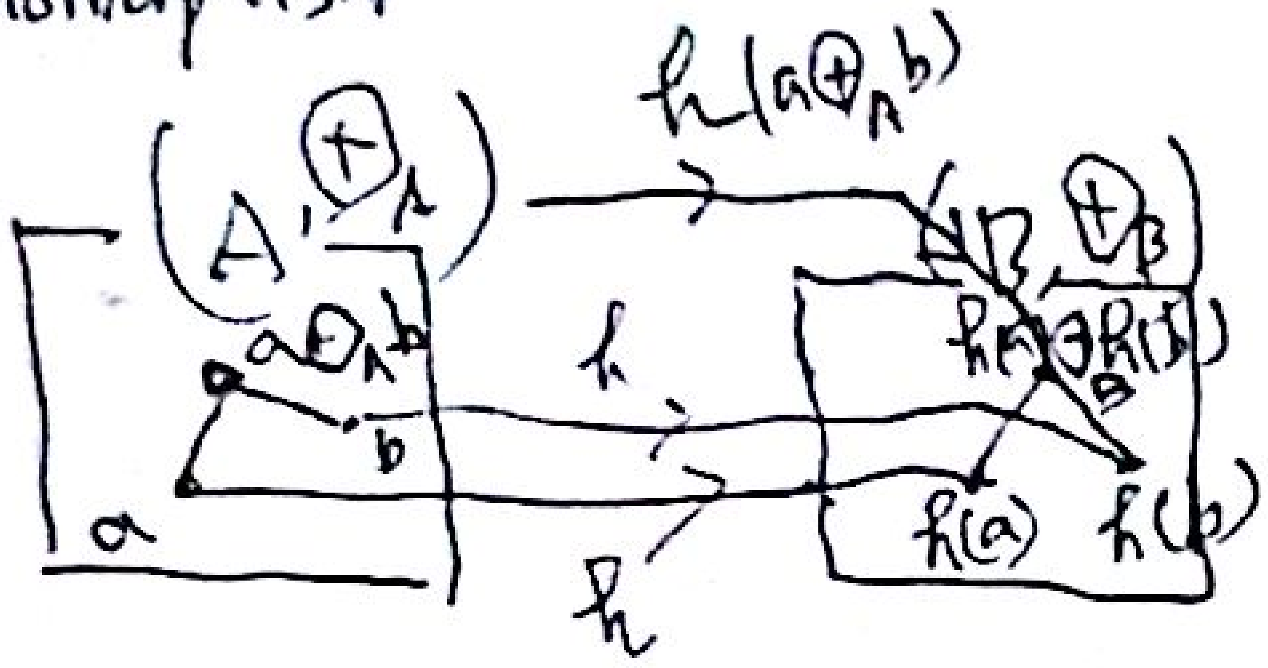
$\bar{L}$, $L - M$

Reversal

1/

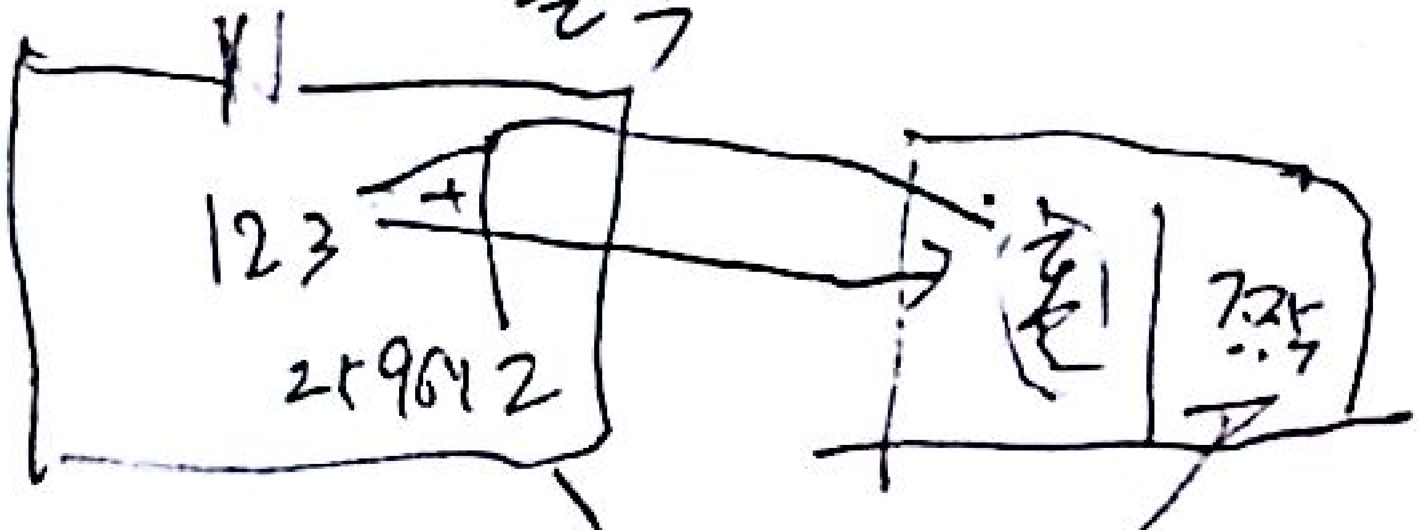
제 11강 10/8(木) Homomorphism, Closure Prop. of RL, minimal-state DFA

Homomorphism



h: onto $|A| \geq |B|$

예) $(\mathbb{N}, +) \xrightarrow{\text{홀짝}} (\{\text{홀}, \text{짝}\}, \oplus_{\text{홀짝}})$ abstract interpretation



String homomorphism

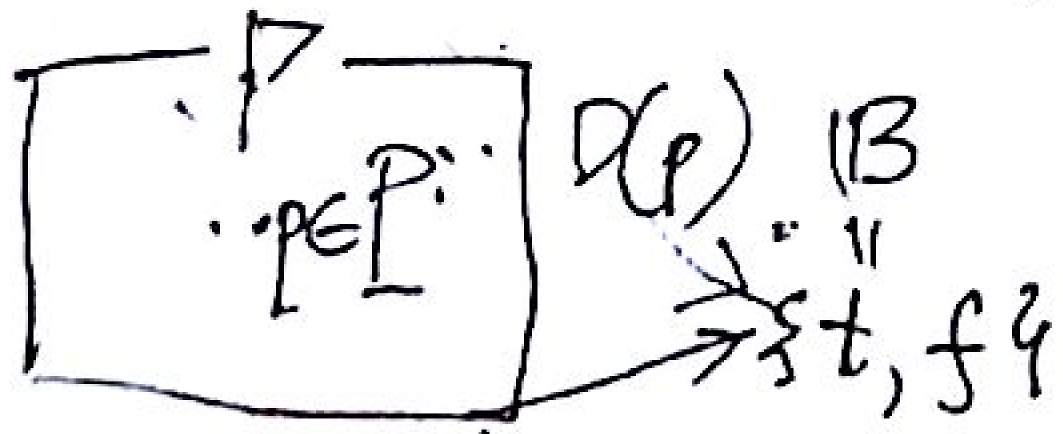
$\oplus_A = \cdot : A^* \times A^* \to A^*$ $\quad \oplus_B = \cdot : B^* \times B^* \to B^*$

$\cdot : 2^{A^*} \times 2^{A^*} \to 2^{A^*}$ $\quad : 2^{B^*} \times 2^{B^*} \to 2^{B^*}$

4.3 Decision Properties of Regular Languages

결정 문제

yes-no



4.4 Equivalence and Minimization of Automata

$\neg(p \equiv q) = p \not\equiv q$

repeat until no change (n states)

$\frac{n(n-1)}{2}$ pairs

Chap 5. Context-free grammar의 간단한 소개
(문맥 자유? 문법)

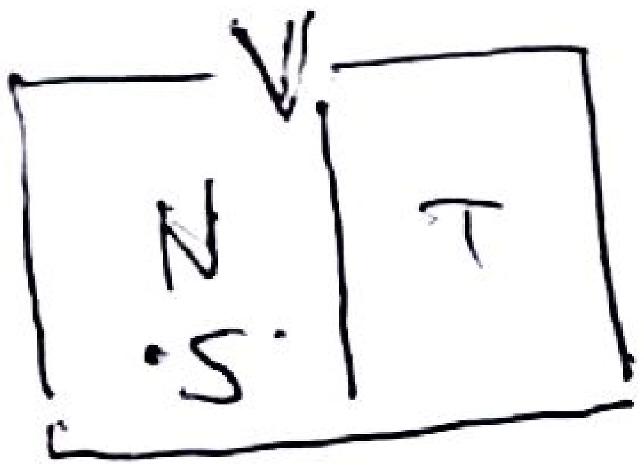
1)

제12강 (10/13, 화) 문맥자유문법 (cfg)

Def. Context-free grammar $G = (N, T, P, S)$

$N \cap T = \emptyset$, $V = N \cup T$. $P \subseteq N \times (N \cup T)^*$, $S \in N$. 學而之

銘名



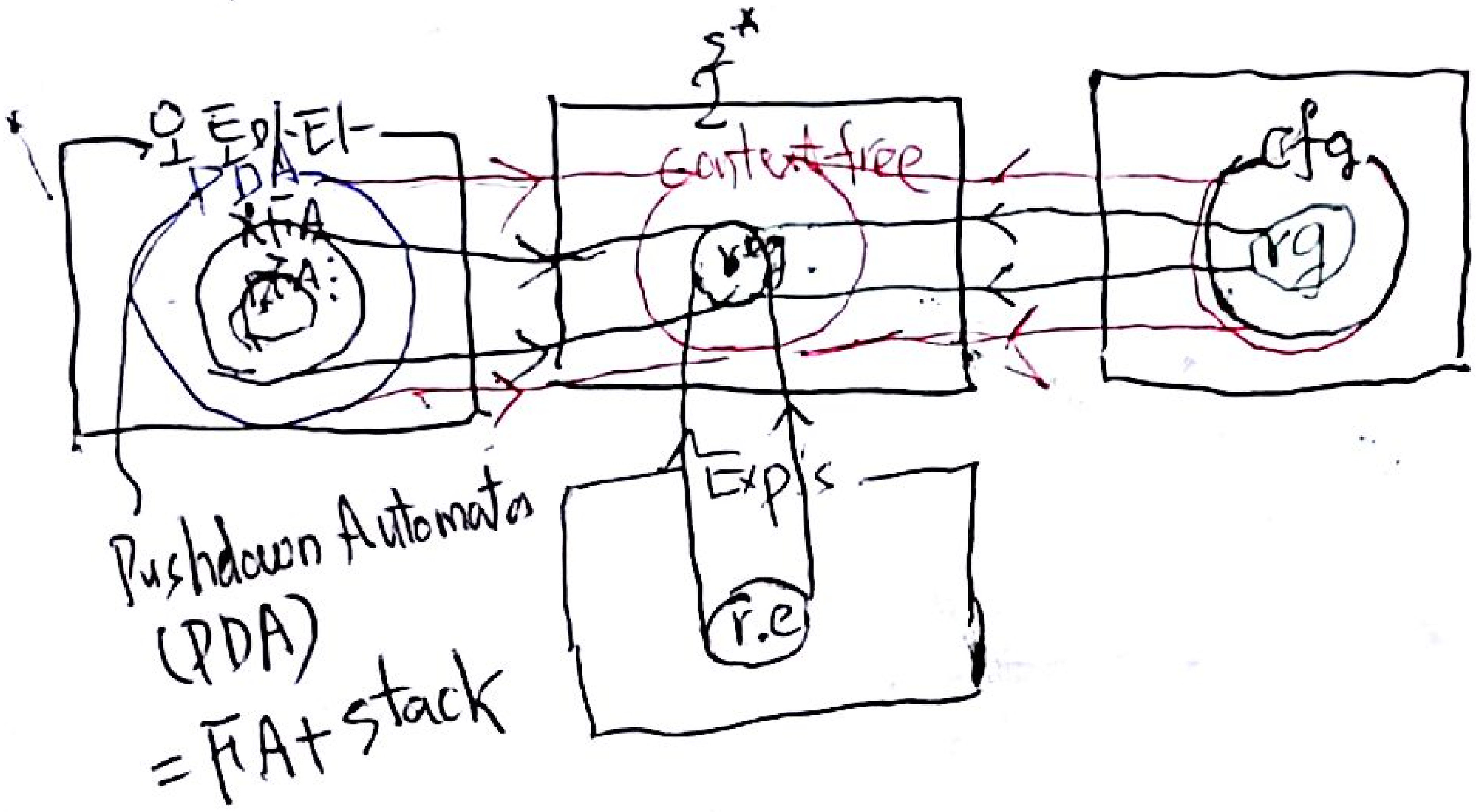
$(A, \alpha) \in P$.

$A \to \alpha \in P$.

名可名 非常名

—노자—

$\to, \Rightarrow$ compare $\Rightarrow^*$ vs. Σ^*



Pushdown Automata (PDA)

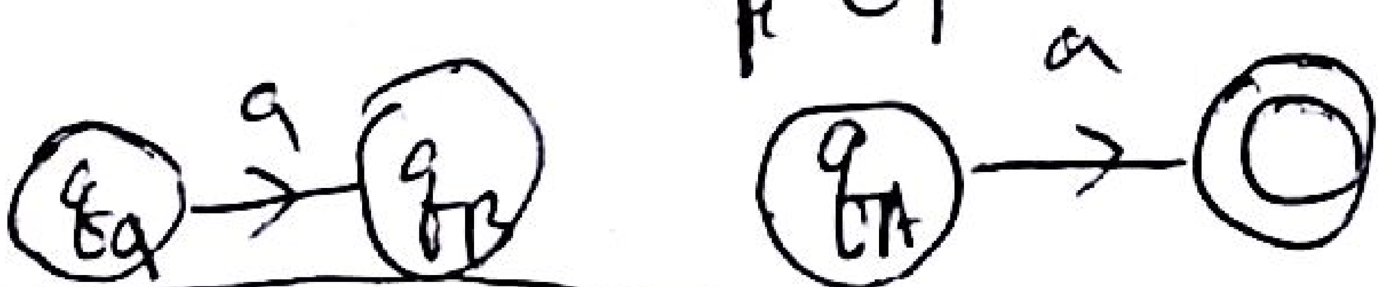
= FA+stack

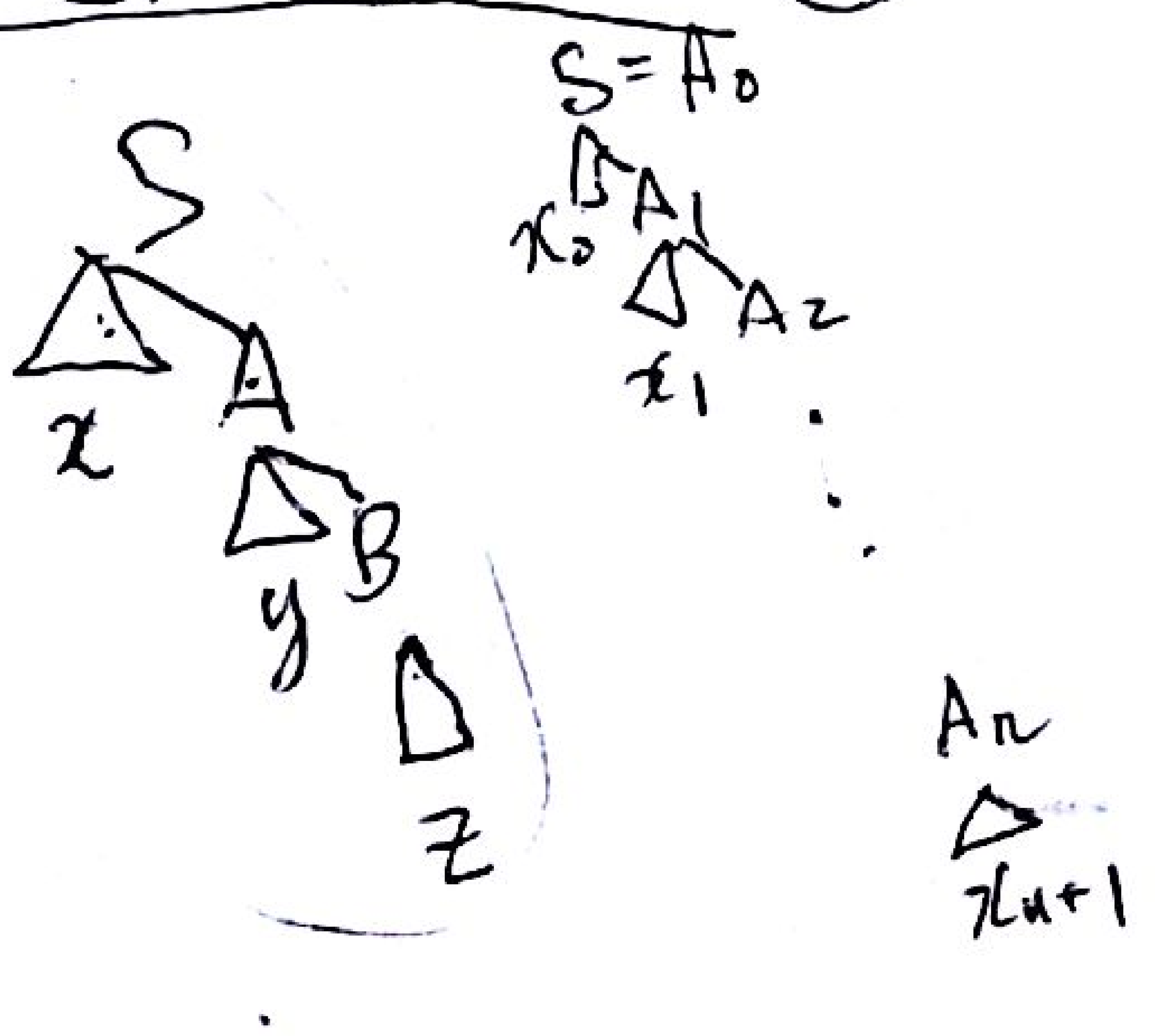
regular grammar

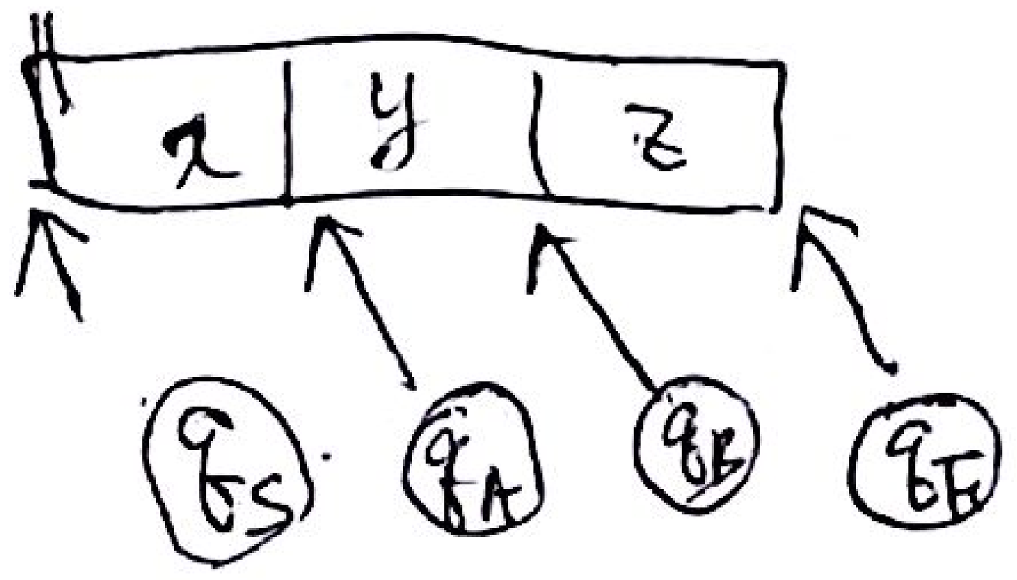
$A \to aB$ or $A \to Ba$ where $A, B \in N$, $a \in \Sigma$.

$q_B \in \delta(q_A, a)$ $f \in \delta(q_A, a)$

$f \in F$







1)

제13강 (10/15 목) $\Rightarrow_{lm}$, $\Rightarrow_{rm}$, Parse Tree, Ambiguity of CFG.

leftmost derivation, rightmost der.

틀릴 수 — 배우는 자
왜? 모르니까
모르면 질문한다

$G=(N,\Sigma,P,S)$ be a cfg.

(Grammar is a rewriting system)

derive, sentential form, sentence. language

Ex) G_1: $E \to E+E \mid E*E \mid a \mid (E)$

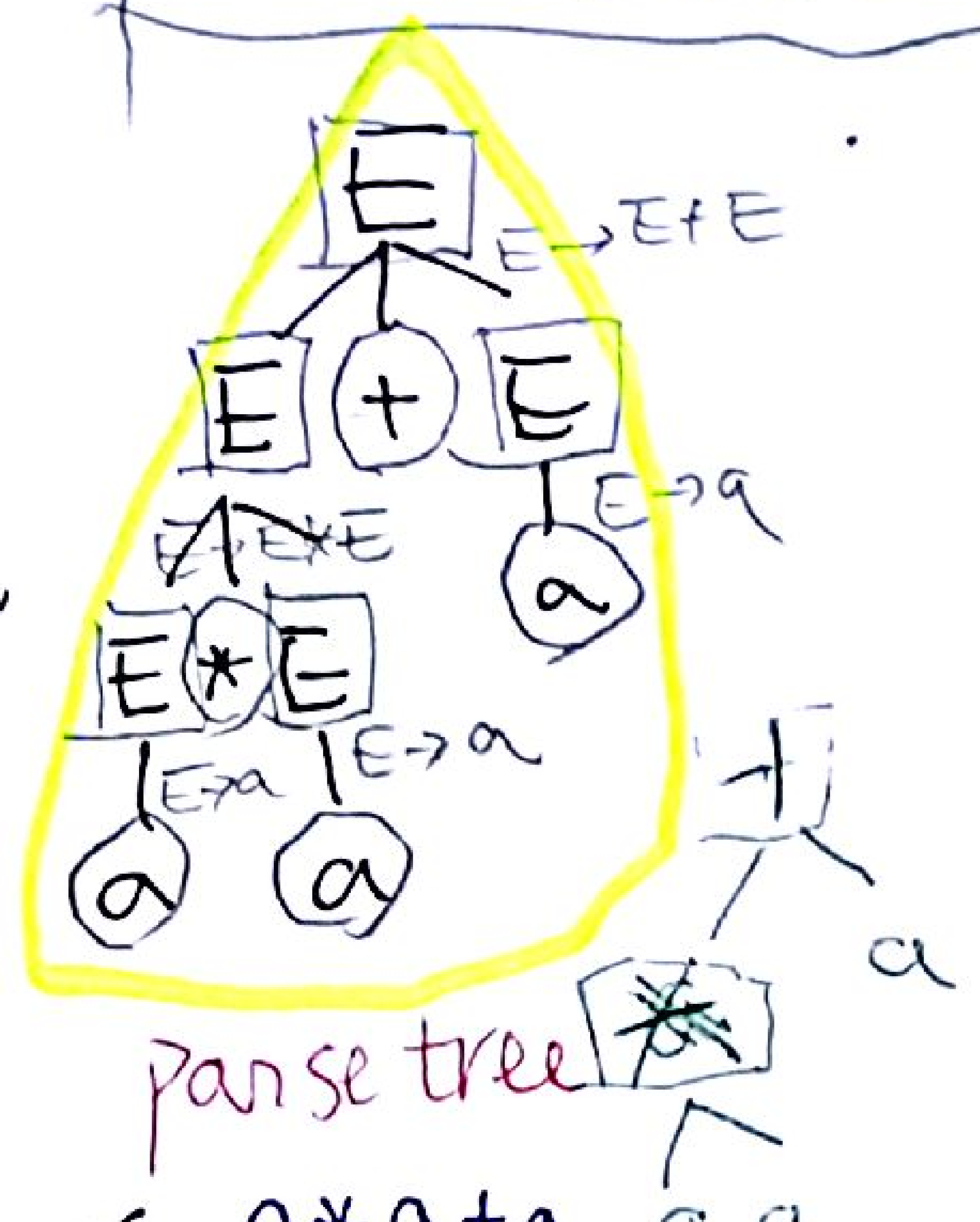
$$E \overset{E\to E+E}{\Rightarrow} E+E \overset{E\to E*E}{\Rightarrow} E*E+E \overset{(E\to a)^3}{\Rightarrow} \underline{a*a+a}$$

Left parse of $a*a+a$

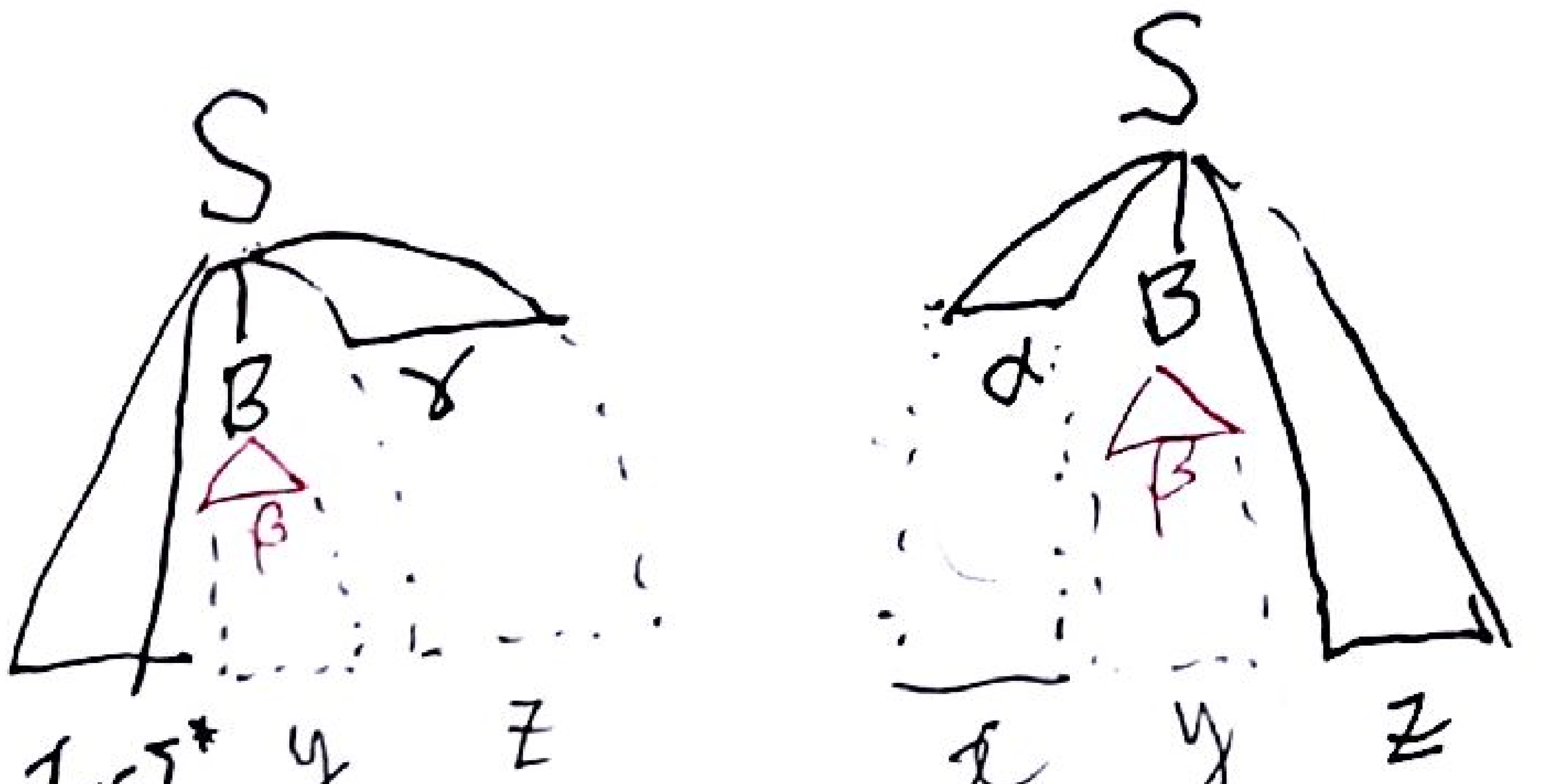
$$E \Rightarrow_{lm}^{E\to E+E \cdot E\to E*E \cdot E\to a \cdot E\to a \cdot E\to a} a*a+a$$

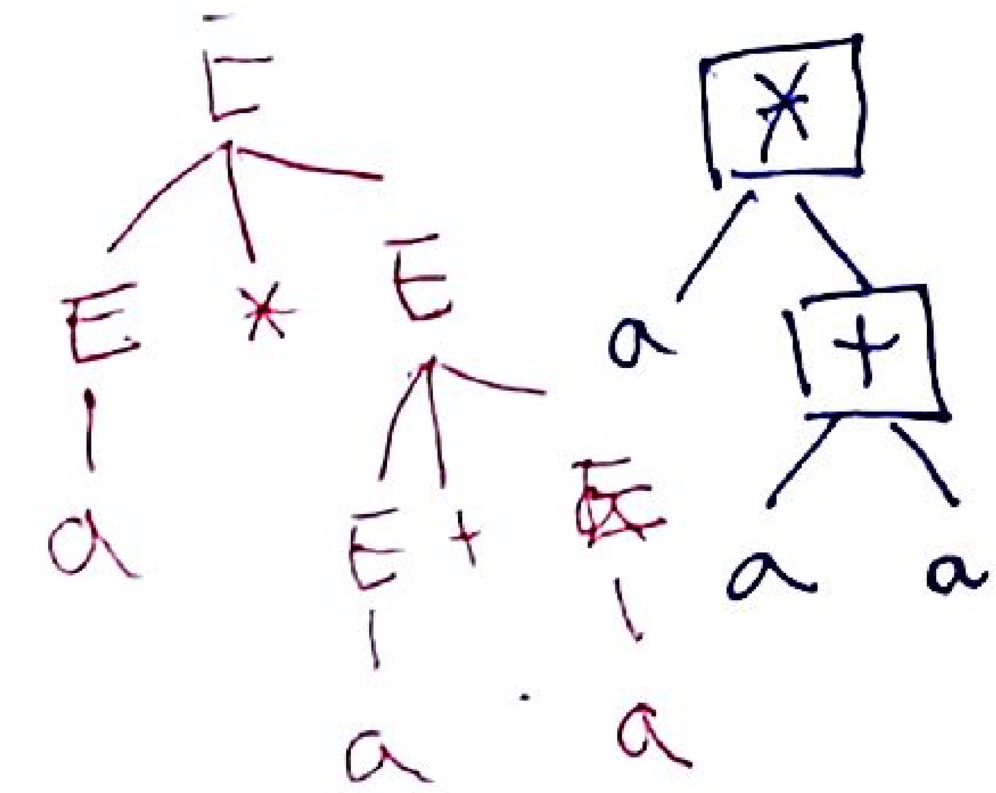
right parse ″

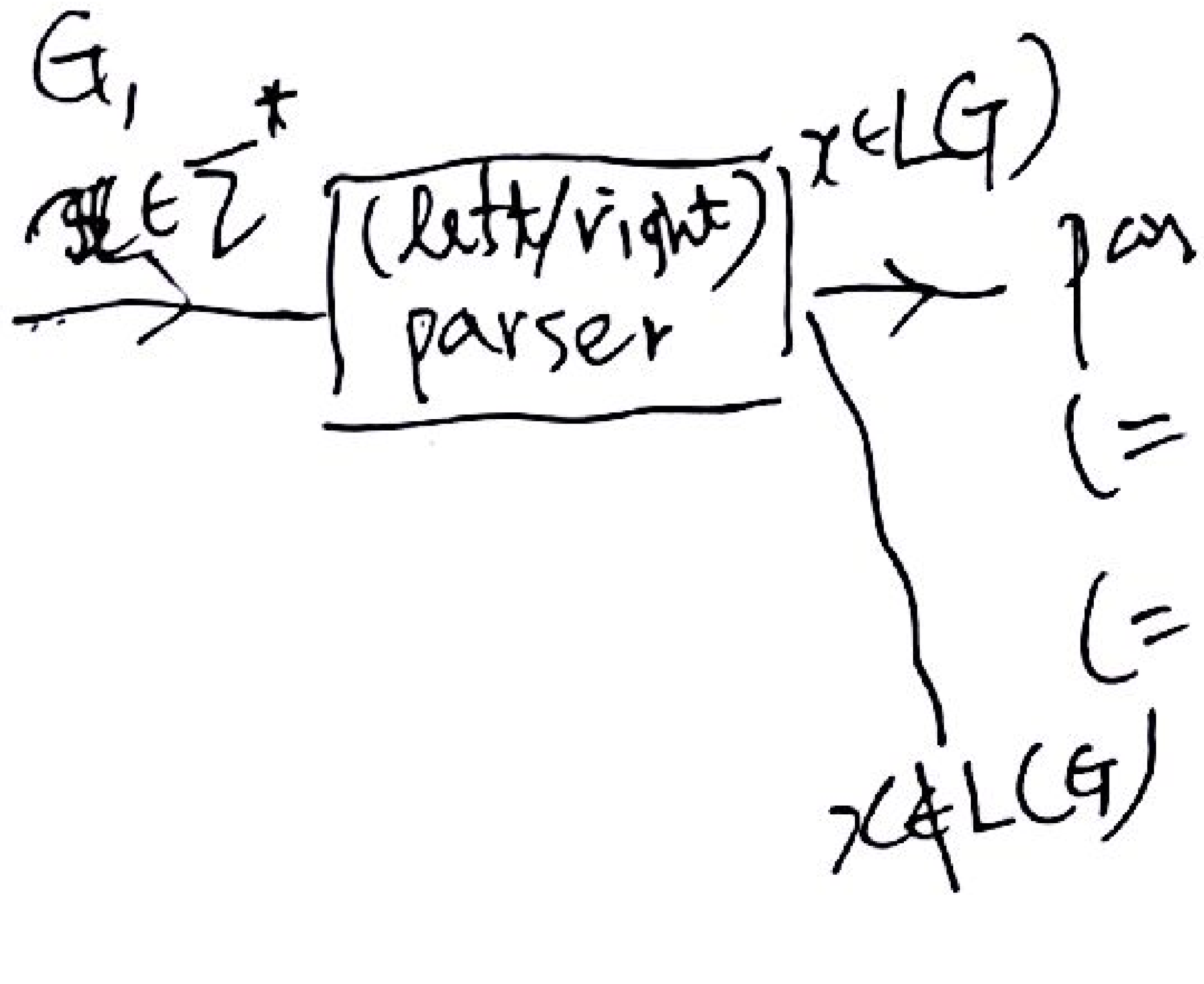
$$E \Rightarrow_{rm}^{E\to E+E \cdot E\to a \cdot E\to E*E \cdot E\to a \cdot E\to a}$$



parse tree of $a*a+a$







parse tree of x
(= left parse)
(= right ″)

문장의 같으나 ~~Trees~~ 문법 나무가 다르다!

⇒ ambiguous
(애매하다)

숙제 #1

출제자: 황진환, 박현지, 이의현

1. $\overline{A\cup(B\cap C)}=(\overline{C}\cup\overline{B})\cap\overline{A}$임을 증명하시오.
(단, 벤 다이어그램을 이용한 증명은 인정하지 않음)

$$
\begin{aligned}
\overline{A\cup(B\cap C)} &= \overline{A}\cap\overline{(B\cap C)} && \rightarrow \text{드모르간법칙}\\
&= \overline{A}\cap(\overline{B}\cup\overline{C}) && \rightarrow \text{드모르간법칙}\\
&= (\overline{B}\cup\overline{C})\cap\overline{A} && \rightarrow \text{교환법칙}\\
&= (\overline{C}\cup\overline{B})\cap\overline{A} && \rightarrow \text{교환법칙}
\end{aligned}
$$

2. R_1과 R_2을 집합 S에 대한 두 관계(Relation)라 하자. 다음 명제들이 참인지 거짓인지 적고, 증명하시오.

A. R_1과 R_2가 각각 Reflexive하면, $R_1\cup R_2$ 또한 그렇다.
참. $a\in S$라 했을 때, R_1과 R_2는 Reflexive하므로, (a,a)는 R_1과 R_2 모두에 포함되어 있다. 따라서 $(a,a)\in R_1\cup R_2$이다.

B. R_1과 R_2가 각각 Symmetric하면, $R_1\cup R_2$ 또한 그렇다.
참. $(a,b)\in R_1\cup R_2$이고, $(a,b)\in S\times S$라 하자. 그러면 (a,b)는 R_1또는 R_2에 포함되어 있고, R_1과 R_2가 Symmetric하기 때문에 (b,a) 또한 R_1또는 R_2에 포함되어 있다. 따라서 $(b,a)\in S\times S$이다.

C. R_1과 R_2가 각각 Transitive하면, $R_1\cup R_2$ 또한 그렇다.
거짓. $S=\{a,b,c\}, R_1=\{(a,b)\}, R_2=\{(b,c)\}$라 하면, R_1과 R_2는 Transitive 하다. 하지만 $R_1\cup R_2=\{(a,b),(b,c)\}$는 (a,c)를 포함하지 않기 때문에 거짓이다.

3. 어떤 집합 S에 대한 Equivalence Relation R이 S를 Disjoint Equivalence Classes로 나눔을 증명하시오.

1) $[x]=\{e\in S\mid xRe\}$라 할 때, $\bigcup_{x\in S}[x]=S$ 이다.

Equivalence Relation은 Reflexive하기 때문에, $x\in[x]$이다.

$S\supseteq\bigcup_{x\in S}[x]\supseteq\bigcup_{x\in S}x=S \qquad \therefore \bigcup_{x\in S}[x]=S$

2) $[x]\cap[y]\neq\varnothing\rightarrow[x]=[y]$이다.

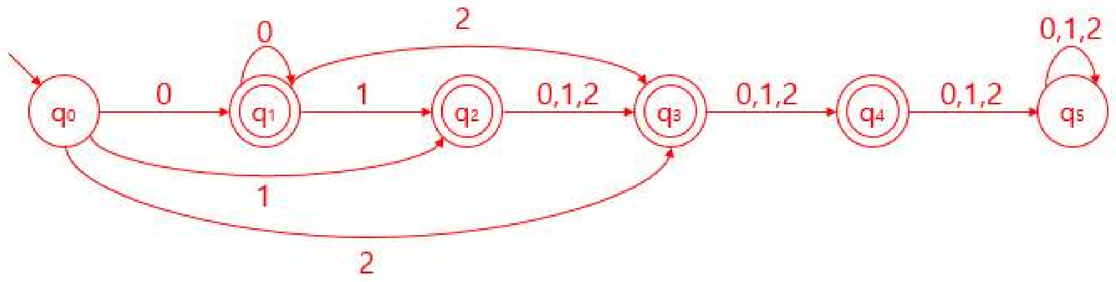
$[x]\cap[y]\neq\varnothing\rightarrow[x]\neq[y]$라 가정하자. $\exists z\in[x]\cap[y]$에 대해서, xRz 그리고 yRz이며, $\forall w\in[y]$에 대해서, zRw이다. R이 Transitive하기 때문에, $xRw\rightarrow[y]\subseteq[x]$. $[x]\subseteq[y]$ 역시 같은 방법으로 증명되고, 따라서 가정에 모순되기 때문에, $[x]=[y]$다.

1)과 2)에 의해, $\bigcup_{x\in S}[x]=S$이고 $[x]\neq[y]\rightarrow[x]\cap[y]=\varnothing$이기 때문에, R이 S를 Disjoint Equivalence Classes로 나눈다.

숙제 #2

출제자: 김성훈, 이의현

1. $\Sigma = \{0, 1, 2\}$일 때, 나타낼 수 있는 문자열 중 200보다 작은 수를 나타내는 문자열은 모두 받아들이고, 그렇지 않은 문자열은 받아들이지 않는 DFA를 그리시오.



1) 000..을 0으로 보고, ϵ을 0으로 보지 않는(input string으로 허용하지 않는) 경우

-> 위 그림

2) 000..을 0으로 보고, ϵ을 0으로 보는 경우

-> 위 그림에서 q_0을 삭제하고, q_1을 initial state로 한다

3) 000..을 허용하지 않고, ϵ을 0으로 보지 않는(input string으로 허용하지 않는) 경우

-> 위 그림에서 q_1에서 0을 보고 q_5로 transition

4) 000..을 허용하지 않고, ϵ을 0으로 보는 경우

-> 위 그림에서 q_0을 final state에 포함시킨다

위 1), 2), 3), 4) 경우 모두 정답 처리했습니다.

2. DFA는 $M = (Q, \Sigma, \delta, q_0, F)$로 표현할 수 있다. 이 때, $\Sigma = \{a, b\}$일 때, prefix가 $bbab$인 모든 문자열만을 받아들이는 DFA의 상태(Q), 상태변화함수(δ), 처음 상태(q_0), 마지막 상태(F)를 각각 정의하시오.

$Q = \{A, B, C, D, E, G\}$

$q_0 = A$

$F = \{E\}$

δ	a	b
A	G	B
B	G	C
C	D	G
D	G	E
E	E	E
G	G	G

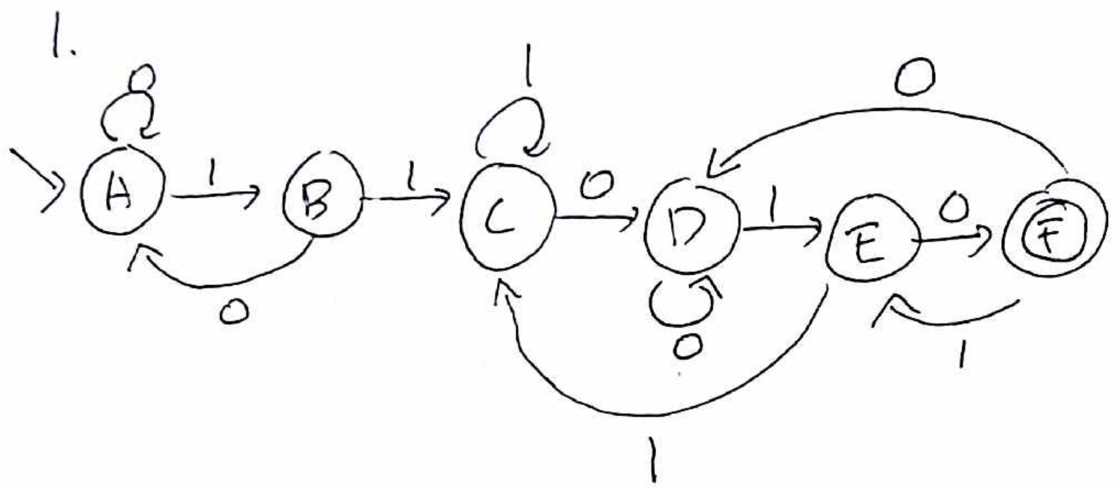
채점기준 : Q, q_0, F가 틀린 경우 각 1점씩 감점했습니다.

δ의 transition table이 틀린 경우 row당 1점씩 감점했습니다.

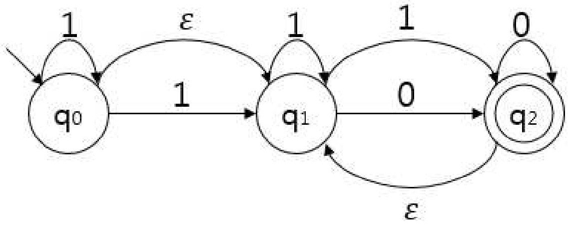
숙제 #3

출제자: 김성훈, 이의현

1. $\Sigma = \{0, 1\}$일 때, 11을 포함하고, 010으로 끝나는 문자열을 받아들이는 DFA를 그리시오.

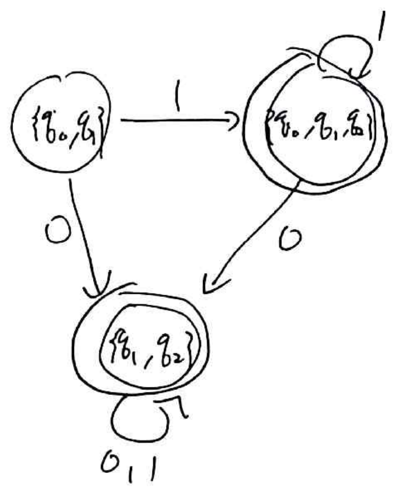


2. 다음 ϵ-NFA를 DFA로 변환하시오. ($\Sigma = \{0, 1\}$)

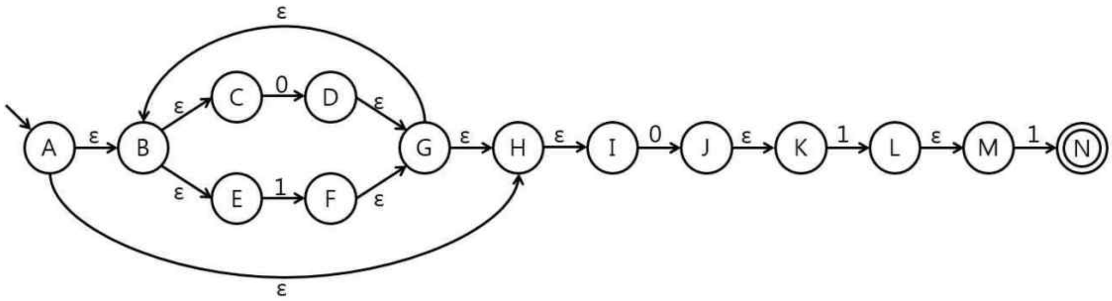


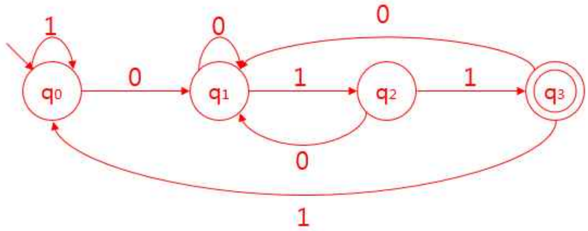
2.

	0	1
→ $\epsilon^*(q_0)$	$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$
* $\{q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_1, q_2\}$
* $\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$



3. 다음 ϵ-NFA를 DFA로 변환하시오. ($\Sigma = \{0, 1\}$)





숙제 #4

출제자: 김성훈, 이의현

1. $(a+b)^*a(a+b)^{n-1}$을 받아들이는 DFA D의 state 최소 개수가 2^n개 임을 증명하시오.

가정) 2^n개의 state보다 적은 state를 갖는 DFA $D=\{Q,\{a,b\},\delta,q_0,F\}$가 존재

then $\exists q$ s.t. $q=\delta^*(q_0,a_1\cdots a_n)=\delta^*(q_0,b_1\cdots b_n)$ (but, $a_1\cdots a_n \neq b_1\cdots b_n$)

then $\exists i, a_i \neq b_i$ suppose $a_i=a$, then $b_i=b$

case 1) if $i=1$

$$\delta^*(q_0,a_1\cdots a_n)\in F, \delta^*(q_0,b_1\cdots b_n)\notin F$$

$$\delta^*(q_0,a_1\cdots a_n)\neq \delta^*(q_0,b_1\cdots b_n)$$

case 2) if $i > 1$

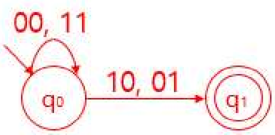
$$\exists p=\delta^*(q_0,a_1\cdots a_n a^{i-1})=\delta^*(q_0,b_1\cdots b_n b^{i-1})$$

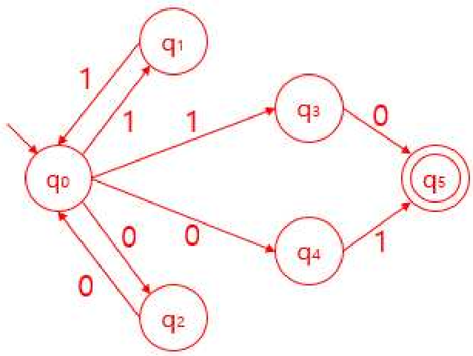
$$\Rightarrow p\in F, \text{ but } p\notin F$$

$$\Rightarrow \delta^*(q_0,a_1\cdots a_n)\neq \delta^*(q_0,b_1\cdots b_n)$$

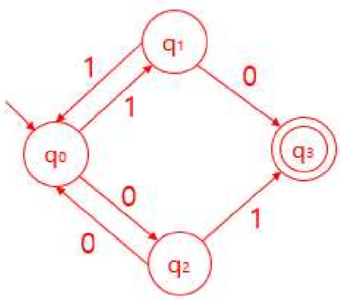
contradiction

2. Regular Expression $(00+11)^*(10+01)$과 동치인 XFA, ϵ-NFA, partial function을 허용하는 DFA, total function만을 허용하는 DFA, minimal state DFA를 그리시오.

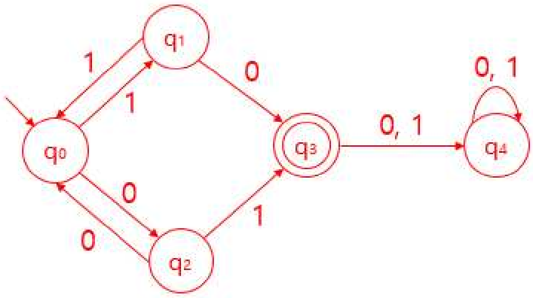
XFA: 

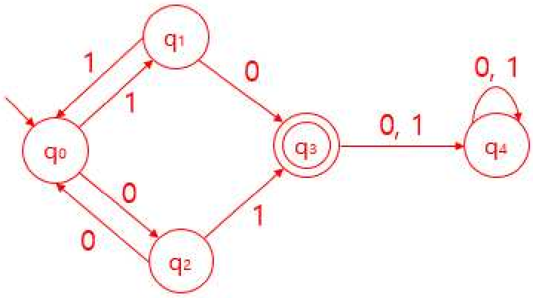


ϵ-NFA:

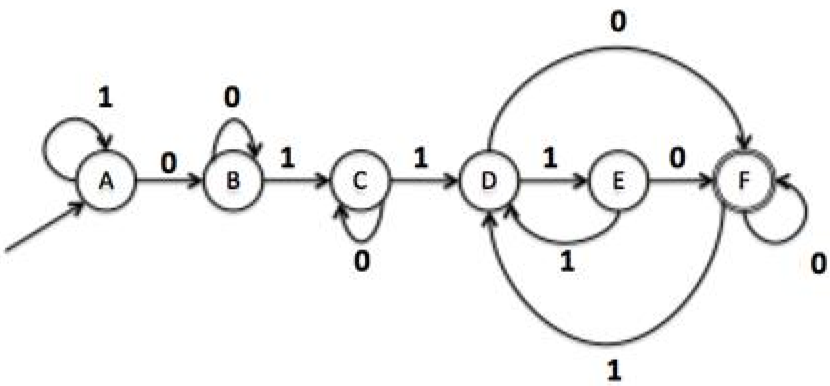


partial function을 허용하는 DFA:

total function만을 허용하는 DFA: 

minimal state DFA: 

3. 다음 DFA가 받아들이는 언어를 표현(denote)하는 Regular Expression을 연립방정식을 이용하여 푸시오.



A = 0B + 1A

B = 0B + 1C

C = 0C + 1D

D = 0F + 1E

E = 0F + 1D

F = 0F + 1D + e = E + e

F = E + e

D = 0(E + e) + 1E = (0 + 1)E + 0

E = 0(E + e) + 1(0E + 1E + 0)

  = 0E + 0 + 10E +11E + 10

  = (0 + 10 + 11)E +0 +10

= (0 + 10 + 11)*(0 + 10)

D = (0 + 1) (0 + 10 + 11)*(0 + 10) + 0

C = 0C + 1((0 + 1) (0 + 10 + 11)*(0 + 10) + 0)

= 0*1((0 + 1) (0 + 10 + 11)*(0 + 10) + 0)

B = 0B + 10*1((0 + 1) (0 + 10 + 11)*(0 + 10) + 0)

= 0* 10*1((0 + 1) (0 + 10 + 11)*(0 + 10) + 0)

A = 00*10*1((0 + 1) (0 + 10 + 11)*(0 + 10) + 0) + 1A

= 1*00*10*1((0 + 1) (0 + 10 + 11)*(0 + 10) + 0)

4. $L = \{a^p | p\text{는 소수}(prime)\}$이 Regular Language가 아님을 보이시오.

가정) $L = \{a^p | p\text{는 소수}(prime)\}$가 Regular하다고 가정하자.

그러면 L을 받아들이는 어떤 DFA $D = \{Q, \{0\}, \delta, q_0, F\}$가 존재한다/

$s = |Q|$라 할 때, $p > s$인 어떤 prime number p를 가정하자.

$0^p \in L$이고 $|0^p| = p > s$이기 때문에, pumping lemma에 의해서, $0^p = uvw$라 쓸 수 있다.

$(v \neq \epsilon$이고 $uv^*w \subseteq L$이다.)

$i = |u| + |w|$이고 $j = |v|$라 하자. 그러면, $uv^*w \subseteq L$는 $\forall k \geq 0, uv^kw = 0^{i+kj} \in L$과 같다.

(또는 $\forall k \geq 0, i + kj$는 $prime$)

$k = 0$일 때, i는 $prime$이다.

$k = i\ (i \geq 2)$일 때, $i(1+j)$는 $prime$이다.

이 때, $v \neq \epsilon$이기 때문에, $j = |v| \geq 1$이고, 따라서 $i(1+j)$는 $prime$이 아니다.

contradiction

5. 다음 DFA를 minimal DFA로 변환하시오. (풀이 과정을 쓰시오.)