이산수학

증명법

○ 연역법 : 탑 다운방식

○ 귀납법 : // ex) 엄마가 얼룩소 , 아빠가 황소, 첫째자식 얼룩소, 둘째자식 얼룩소면 셋째자식도 얼룩소

5주차(16.04.07)

관계

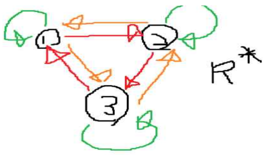
○ 도메인 : 관계(집합 등/ {a,b,c,d))

* 반사관계
  * **반사 관계** : 1,1 2,2 같이 자기한테로 돌아오는 것/ (전체다) 하나라도 아니면 반사관계가 아니다.
  * **비반사 관계** : (x,x)가 없는 것 / (1,1) (2,2) 가 하나도 없는 것
* 대칭관계
  * **대칭관계** : 원소 (a,b)가 존재하면 (b,a)가 반드시 존재함 / 반사와는 다르게 하나가 없어도 대칭관계이다.
    ex) x+y가 20일 때 (1,19)(19,1) 대칭관계가 맞다.
  * **반대칭관계** : (a,b)가 있으면 (b,a)가 **없어야**한다.
* 추이관계
  * **추이관계** : a->b->c 관계가 있는데 a->b, b->c가 있는데 a->c도 있으면 추이 관계
    / (1,2 2,1 이면 1,1이 있으므로 추이관계.)
    / 어떻게 되든 가는 길이 2개면 추이관계 성립
    //참고 : 추이 클로우저(closur은 닫는다, 폐쇄라는 뜻.) 중요하진 않음

R의 추이클로우저 $R^+ = \bigcup_{n=1}^{\infty} R^n = R \cup R^2 \cup \cdots$

만약 (a,b)=R이면 (a,b)=R+이다.

만약(a,b)=R+이고, (b,c)=R이면 (a,c)=R+ // a,c를 R+에 추가시킨다. // 더 이상 추이관계가 없을 땐 폐쇄한다.



대칭, 반사, 추이관계가 전부 적용되면 동치관계

(R+(R대거/플로스가 아님)는 1~무한대 / R*(R아스트리스크)은 0~무한대)(,)(,)

R = {(1,1)(1,3)(2,2)(2,4)(3,1)(3,3)(4,2)(4,4)} 동치관계 / A={1,2,3,4}

○ **동치류** / [x]={1,3,2,4}

○ **동집합** : : 동치류의 모임을 몫집합이라 함 A/R={{1,2,3,4}} //집합의 집합

○ **분할** : 집합을 쪼개서 교집합이 안생기면 됨. // x mod(modulo) y / 7 mod 3 = 1

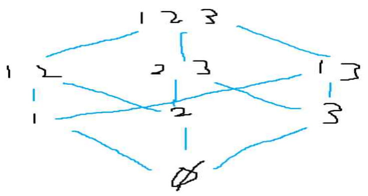
합동 : x=(줄세개임)y(mod m) x,y를 m으로 나눈 나머지가 같을 때 합동이라고함.

○ **부분 순서 관계** : a<~b : a가 b에 앞선다 ex) a=1, b=2 일 때 a는 b에 앞선다.

//1,1 2,2 3,3 도 포함함(반사관계도 포함)

[= 반순서관계 참고:(알고리즘수업 때 Sorting정렬 : 무작위 숫자를 순서대로 정렬). 토너먼트형식같은 것.]

- 멱집합(Pa) : 집합 A에서 만들 수 있는 모든 집합들의 집합



멱집합 : {공집합,{1},{2},{3},{1,2},{2,3},{1,3},{1,2,3} } // {1,2,3}으로 집합 A도 포함

함수

❑ **함수** : 어떤 객체와 진리 값의 집합이다.

ex) y=f(x)=x+1={(1,2)(2,3)(3,4).....} 무한대로 늘어가는 집합

F : X -> Y f는 X에서 Y로 사상한다(상이 맺힌다.)

○ **정의역** : : X의 원소 //정의역에서는 항상 입력하는 것을 줘야함

○ **공변역** : : Y의 모든 값

○ **치역** : : X에서 Y를 사상해서 나오는 값 / ex)x=123 , y=abcd 일 때 d는 아무 상관없음)

* 함수가 되는 것은 일대일 / 다대일 // 함수가 안되는 것

○ 함수의 종류

- **단사함수** : 일대일 함수
- **전사함수** : 치역과 공변역이 동일한 함수 / 공변역에서 버리는 함수가 하나도 없다.
- **전단사 함수** : 남는 것도 없고 일대일로 대응되는 것(단사함수+전사함수)

* 합성함수 : 추이관계와 같다.

a->b에서 받은 함수를 b->c에서 다른 함수로 쓸 수 있는 것 ex)a를 입력하면 3이 a가되고 a를 입력하면 y가됨

* 항등함수 : 값이 같아지는 것 (시험x)

* 역함수 : 논항 값이 함수 값이 됨 (시험x)

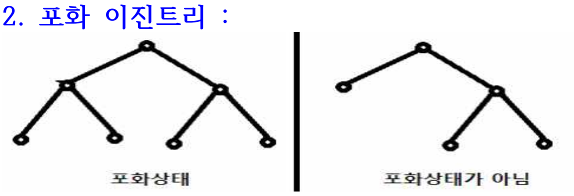
- **특성함수** :

- **올림함수** : x보다 크거나 같은 정수 값 중 가장 작은 값을 나타냄 [3.5]라 치면 4가 됨.
- **내림함수** : x보다 작거나 같은 정수 값 중 가장 큰 값을 나타냄 [3.5]라 치면 3이 됨

중간고사 시험문제!!!!

1. 드모르간 법칙 : ~(p∩q) -> (~p)∪(~q)

2. 포화 이진트리 :

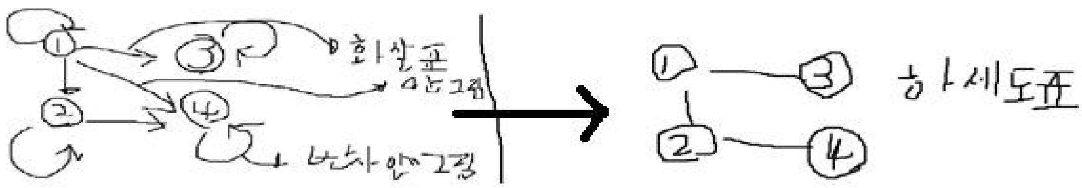


3. 대우 증명법 : p->q = ~q-> ~p 일 때 두 개 중 한 개만 증명하면 된다. (짝수 일 때 2k를 쓰자)

4. 카티시안 곱 : AXB, A={1,2,3}, B={a,b} 일 때 카티시안 곱(AXB)은 3*2 = 6이다.

5. 다음관계가 반사관계인지 대칭관계인지 추이관계인지 서술하시오.

6. 하세도표 로 바꾸시오. : 필요 없는 화살표를 다 지움 (당연한 화살표)



7. 함수의 종류
   - **단사함수** : 일대일 함수
   - **전사함수** : 치역과 공변역이 동일한 함수 / 공변역에서 버리는 함수가 하나도 없다.
   - **전단사 함수** : 남는 것도 없고 일대일로 대응되는 것(단사함수+전사함수)
   - **올림함수** : x보다 크거나 같은 정수값 중 가장 작은 값을 나타냄 [3.5] 라치면 4가 됨.
   - **내림함수** : x보다 작거나 같은 정수 값 중 가장 큰 값을 나타냄 [3.5]라치면 3이 됨

○ **깊이 우선 탐색 : 왼쪽부터 차례로 들어가며 탐색**

○ **넓이 우선 탐색 : 옆으로 자식 꼭지점의 개수를 세면서 탐색**

* 트리탐색 용어 : 꼭지점 이동 : traversel / 꼭지점 뒤로 이동 : backtracking

시험범위 논리연산~그래프 **중간고사 범위 끝**

===

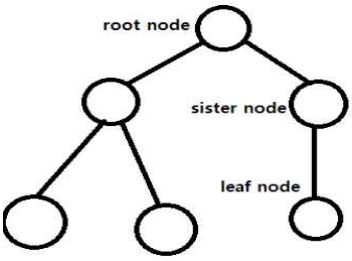
6주차(16.04.14) 7주차(16.04.21)

그래프

❑ **그래프** : G = (V, E) / 유한개의 개수를 정점 또는 노드들의 집합인 V와 연결선 또는 에지라고 불리는 정점들의 쌍들의 집합

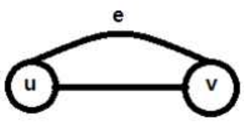
ex) 1 -> 2 -> 3 -> 4 // 1은 2에 대해 선행자 / 2는 1에 대해 후속자.

○ **트리**



- 시작 노드(root node/ parent node) : 밑에 자식 노드가 있음 / 비종말 노드(non-terminal node)
- 중간 노드(sister node) : 중간 노드 중 동등한 위치에 다른 노드가 있음 / 비종말 노드(non-terminal node)
- 이파리 노드(leaf node) : 밑에 더 이상 자식노드가 없음 / 종말 노드(terminal node)

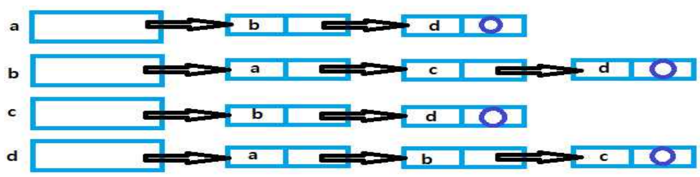
○ **연결선**

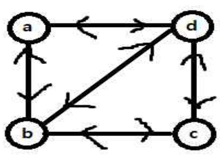


- (u,v) E E에 대해 u와 v를 연결하는 연결선 e는 u와 v에 접했다(사건이 일어남), u와 v는 인접했다.

○ **인접 리스트 (Linked List)**

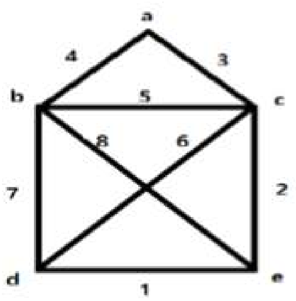
- 각 장점에 대해 포인터가 주어지고, 그 점으로부터 연결된 장점들을 차례로 연결 리스트(Linked List)로 표시함
- 같은 리스트 내에서는 순서에 관계가 없음



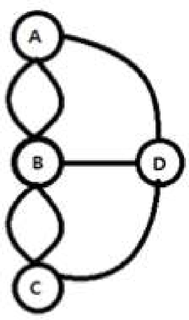


	a	b	c	d
a	0	1	0	1
b	1	0	1	1
c	0	1	0	1
d	1	1	1	0

○ **오일러 경로 : 각 연결선을 단 한 번씩만 통과**



오일러 경로

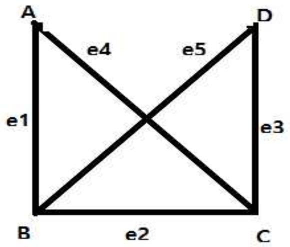


오일러 경로가 아님

<- 홀수로 나가는 선이 2개를 넘었으므로 오일러 경로가 없다.

○ **오일러 순회 : 각 연결선을 가서 다시 돌아옴**

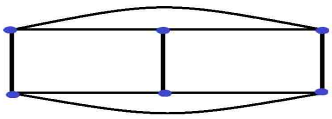
○ 차수 : 정점과 연결된 연결선의 개수
- 모든 정점들의 차수의 합은 연결선들 개수의 2배이다.



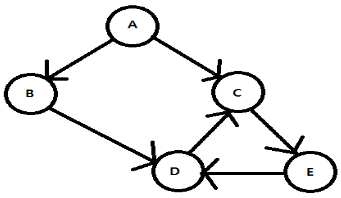
연결선의 개수(5) *2 = 차수의 개수(10)

○ 해밀턴 경로 : 각 꼭지점을 단 한 번씩만 통과하지만 시작점으로 돌아오지 않음
○ 해밀턴 순회 : 각 꼭지점을 단 한 번씩만 통과하고 시작점으로 돌아옴

○ 시험 : 연결된 평면 그래프에서 꼭지점의 수를 v, 연결선의 수를 e, 면의 수를 f라고 할 때 v-e+f=2이다.
ex) 꼭지점의 개수 = 6, 연결선의 개수 e=9, 면의 수 f=5(바깥면도 친다), v-e+f=2의 오일러 정리가 성립함.

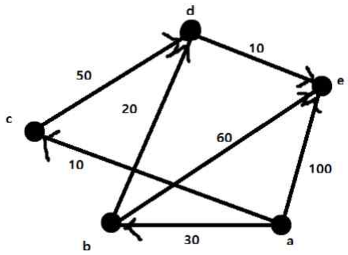


○ 연결선
○ 완전 그래프
- 완전 그래프에서 연결선의 개수는 n(n-1)/2개이다. (n은 꼭지점) ex) n=5이면 5*4/2 10개
○ 방향 비사이클 그래프 : 순환이 없다. // 자매끼리 leaf node의 화살표를 주고 받는다



<- 트리, DAG, 사이클을 가진 방향 그래프

○ 다익스트라 알고리즘 : 최단경로를 찾는 알고리즘, 시작 지점에서 최단경로 꼭짓점을 찾아내는 식으로 구함

ex)


단계	탐색 전 최단경로 집합 (S)와 남은 꼭짓점(V)	D[b]	D[c]	D[d]	D[e]	최단경로 꼭짓점	탐색 후 최단경로 집합 {S}와 남은 꼭지점{V}
1	S={a} V={b,c,d,e}	30	10	∞	100	c	S={a,c} V={b,d,e}
2	S={a,c} V={b,d,e}	30	10	60	100	b	S={a,c,b} V={d,e}
3	S={a,c,b} V={d,e}	30	10	50	90	d	S={a,c,b,d} V={e}
4	S={a,c,b,d} V={e}	30	10	50	60	e	S={a,c,b,d,e} V={ }

행렬과 행렬식

❑ **행렬(매트릭스) : 가로(밑) = 행, 세로(옆) = 열**

- 정방행렬 : 행과 열의 수자 같다
- A_{ij} = i(행) j(열)

○ **행렬의 합 : 각각의 번호를 더함**

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} = \begin{pmatrix} 3 & 3 \\ 4 & 6 \end{pmatrix}$$

○ **행렬의 곱 : 왼쪽 행과 오른쪽 열의**

$$C_{11} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj}$$

ex) $\begin{pmatrix} 2 & 3 \\ 3 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 3 \\ 4 & 2 \end{pmatrix} = \begin{pmatrix} (2\times1+3\times4) & (2\times3+3\times2) \\ (3\times1+1\times4) & (3\times3+1\times2) \end{pmatrix} = \begin{pmatrix} 14 & 12 \\ 7 & 11 \end{pmatrix}$

○ **대각행렬** : 정방행렬에서 **대각선을 제외한 모든 항이 0**인 행렬

○ **항등행렬 : 대각선이 모두 1로 이루어진 행렬, 역행렬을 구할 때 쓰임**

- $A * A^{-1} = I$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

○ **전치행렬 : 행과 열을 90도로 뒤집음(행이 열이되고 열이 행이 됨) (뒤집기임)**

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$$

○ **대칭행렬 : 행렬과 전치 행렬의 값이 똑같다.**

$$\begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$$

○ **교대행렬** : A=-AT(전치행렬)를 만족한다

$$\begin{pmatrix} 1 & -2 \\ 2 & 3 \end{pmatrix}$$

○ **삼각행렬**

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

○ **가우스 조단 소거법** : 정 [3 2 4 |5] = $3x_1 + 2x_2 + 4x_3 = 5$
[2 4 8 |9] = $2x_1 + 4x_2 + 8x_3 = 9$
[q w e |t]
이걸 1 0 0 로 만드는 것
0 1 0
0 0 1

○ **역행렬** : 행렬(A) * 역행렬(A-1) = 항등행렬(I) 이 되는 것.
2a + 5c =x1, 2b + 5d =x2, 1a + 3c = y1, 1b + 3d = y2 가 되는 것

○ 가우스 조단 역행렬 구하는 알고리즘 : 절차가 존재함 정 [A | I]

$$\left[\begin{array}{cc|cc} 3 & 4 & 1 & 0 \\ 2 & 3 & 0 & 1 \end{array}\right] \quad (\frac{1}{3}) * R1 \rightarrow R1$$

$$\left[\begin{array}{cc|cc} 1 & \frac{4}{3} & \frac{1}{3} & 0 \\ 2 & 3 & 0 & 1 \end{array}\right] \quad (-2) * R1 + R2 \rightarrow R2$$

$$\left[\begin{array}{cc|cc} 1 & \frac{4}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & -\frac{2}{3} & 1 \end{array}\right] \quad 3 * R2 \rightarrow R2$$

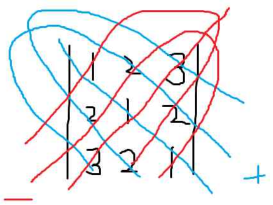
$$\left[\begin{array}{cc|cc} 1 & \frac{4}{3} & \frac{1}{3} & 0 \\ 0 & 1 & -2 & 3 \end{array}\right] \quad (-\frac{4}{3}) * R2 + R1 \rightarrow R1$$

$$\left[\begin{array}{cc|cc} 1 & 0 & 3 & -4 \\ 0 & 1 & -2 & 3 \end{array}\right]$$

1. 첫 번째 줄에 1/3을 전부 곱해줌 -> 첫 번째 줄에 값 입력 -> 1*1이 1이 됨
2. 첫 번째 줄에 -2를 전부 곱하고 R2에 전부 더해줌 -> 두 번째 줄에 값 입력 -> 2*1이 0이 됨
3. 두 번째 줄에 3을 전부 곱해줌 -> 두 번째 줄에 값 입력 -> 2*2가 1이 됨
4. 두 번째 줄에 (-4/3)을 곱하고 첫 번째 줄을 더해줌 -> 첫 번째 줄에 값 입력 -> 1*2가 0이 됨

○ 사루스의 공식 : 3X3 이하의 행렬에서 대각선으로 (왼쪽위->오른쪽아래)는 플러스 (오른쪽위 ->왼쪽밑)은 마이너스

$$|A| = \begin{vmatrix} 2 & 1 \\ 1 & 4 \end{vmatrix} = 2 \times 4 - 1 \times 1 = 8 - 1 = 7$$



-> 각각을 곱해서 나온 숫자를 다 더해줌.

○ **도치** : 열의 수를 봐서 j의 수가 뒤에 수보다 클 때 도치라고 함 (도치의 개수가 짝수(우치환)냐, 홀수(기치환)냐에 따라 바뀜)

배열	j1,j2의 도치수	부호	치환	a1j1 a2j2
1 2	0	+	우치환	a11 a22
2 1	1	-	기치환	a12 a21
1 2 3	0	+	우치환	
1 3 2	1	-	기치환	
2 1 3	1	-	기치환	
2 3 1	2	+	우치환	
3 1 2	2	+	우치환	
3 2 1	3	-	기치환	

(시험 안나옴) ○ 소행렬 : r번째 행과 s번 째 열을 제거해서 얻은 행렬

- 여인수 : I+j가 짝수이면 +, 음수이면 -

$$A=\begin{bmatrix}1&2&-1\\-1&3&5\\2&4&3\end{bmatrix}\quad M_{11}=\begin{bmatrix}3&5\\4&3\end{bmatrix}$$

$$\underset{\text{여인수}}{A_{11}}=(-1)^{i+j}\times\begin{bmatrix}3&5\\4&3\end{bmatrix}$$

$$=(+1)\times(3\times3-5\times4)$$

$$=-11$$

- det(A) = a11(각각의 겹쳐지는 원소)*A11(남은행렬)+ // 어떤 행을 택해도 값은 똑같이 나옴

$$\det(A)=a_{11}A_{11}+a_{12}A_{12}+a_{13}A_{13}\qquad\begin{vmatrix}3&1&4\\2&0&3\\1&-1&2\end{vmatrix}$$

$$=3\begin{vmatrix}0&3\\-1&2\end{vmatrix}-1\begin{vmatrix}2&3\\1&2\end{vmatrix}+4\begin{vmatrix}2&0\\1&-1\end{vmatrix}$$

11주차(16.05.19)

오토마타, 형식, 언어, 문법

❑ 오토마타 : 오토마톤(자동기계)의 복수 (앨런 튜링이 만듦/ 튜링머신)

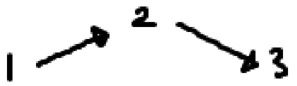
// 입출력이 유한한 크기의 한 줄로 되어있고, 그 안에서 왔다갔다함.

❑ 오토마타의 타입 // 단계가 올라갈수록 고급

1단계	유한 오타마타 (정규문법)	2단계	푸시다운 오타마타 (문맥자유 문법)	3단계	선형제한 오토마타 (문맥민감 문법)	4단계	튜링머신 (문맥자유 문법)

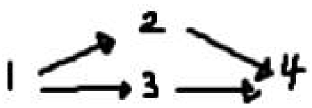
❑ 1단계 : 유한오타마타 : 방향그래프로 나타낼 수 있음 (Regular Language)

○ **결정적 오타마타** : 가는 단계가 결정되어 있는 것



○ **비결정적 오타마타** : 2가지 이상의 이동이 가능한 오타마타

- 출력여부에 따라 인식기와 트랜스듀서로 나누어짐

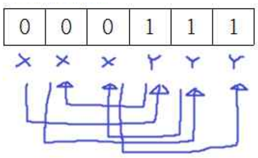


○ **오타마타 예제문제**

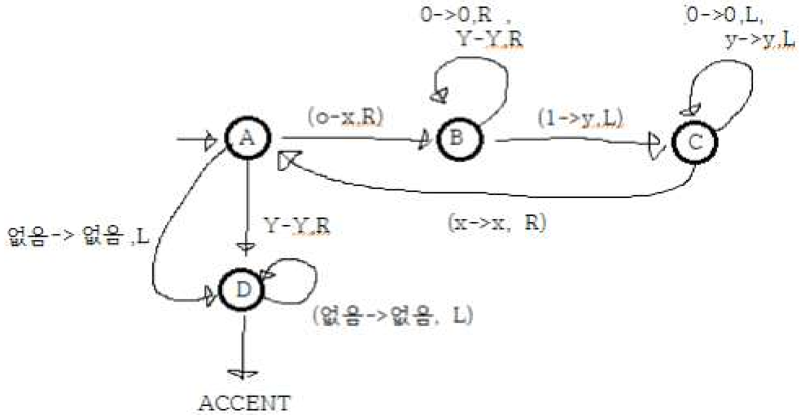
1번) L = 01*0(*은 개수가 무한) 일 때, 나올 수 있는 것이 00 / 010 / 0110 / 01110 / ,,, 로 삼각형형태로나옴

2번) L= 0n1n(n은 n승) 일 때, 나올 수 있는 것이 01 / 0011 / 000111 / ...

3번) 0 -> x, R 일 때 0을 X로 바꾸고 오른쪽으로 한칸, 1-> y, L 1을 y로 바꾸고, 왼쪽으로 한칸 일 때,



4번) 오토마타 그래프



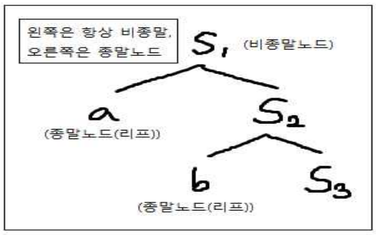
○ 전이 그래프

$$S_1 \xrightarrow{a} S_2 \xrightarrow{b} S_3$$

$$||$$

$$S_1 \rightarrow aS_2 \;,\; S_2 \rightarrow bS_3$$

$$=$$



* FSA(파이널 스테이 오타마타) = RG(정규언어) // 끝이 있는 오타마타

○ 결정적 유한 오타마타 : 다섯 개의 순서쌍으로 이루어짐

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

Q : 상태들의 유한 집합

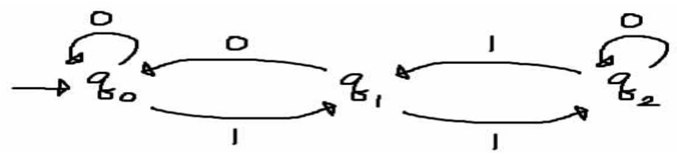
Σ : 0, 1을 쓸 것인지, a, b 등을 쓸 것인지

δ : 전이 함수의 집합

q_0 : 시작상태

F : 최종 단계 상태의 집합

○ 출력이 없는 유한 오타마타

DFA M = ({ q_0, q_1, q_2 } ,{0, 1}, δ, q_0, {q_1}) 일 때, δ : $\delta(q_0, 0) = q_0$, $\delta(q_0, 1) = q_1$ $\delta(q_1, 0) = q_0$, $\delta(q_1, 1) = q_2$ $\delta(q_2, 0) = q_2$, $\delta(q_1, 1) = q_1$ 일 때,	그래프로 나타내면 

// 해석 : { q_0, q_1, q_2 } 의 집합이 있고, {0, 1}의 숫자를 사용, q_0으로 시작하여, {q_1}이 최종단계

○ 유한 오토마타의 응용 : 철자 교정기 등을 짤 수 있음

<id> -> <letter><rest>

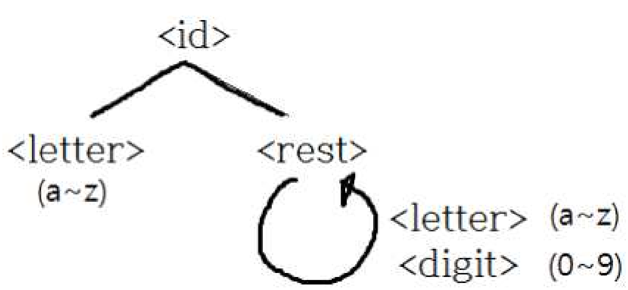
<rest> -> <letter><rest> | <digit><rest> | λ

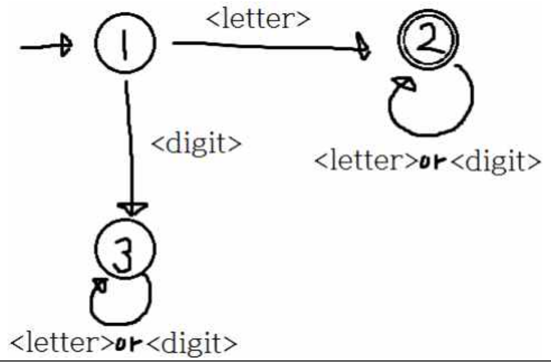
<letter> -> a | b | … | z | _

<digit> -> 0 | 1 | … | 9

//BNF에서는 <id>::=<letter><rest>로 표기

id -> rest로 갈 때 letter가 필요한데, letter가 종말 루트 여야 하므로,
letter-> a|b|...|z|_에서 불러와 a~z가 설정됨.





12주차(16.05.26)

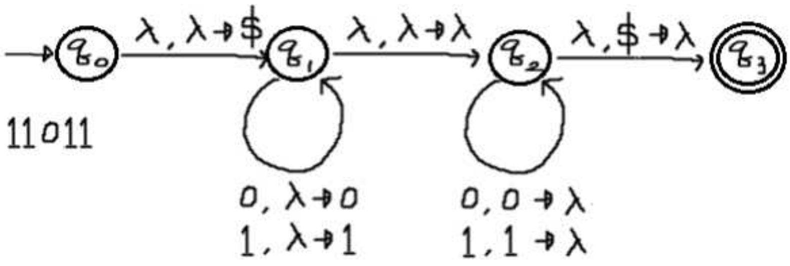
○ **스트링에서의 연산**

* 연결 : 말그대로 스트링 두 개를 연결하는 것 // ex) 동명 + 대학교 = 동명대학교
* 역 : 스트링을 뒤집는 것 // ex) 대학교^R = 교학대
* Σ^* : 0승부터 시작 // Kleene star
* Σ^+ : 1승부터 시작 // Dagger

○ **배커스 나우어 표기법**

❑ 2단계 : 푸시다운 오타마타(문맥자유 문법(CFL))

2단계 : 푸시다운 오토마타[문맥자유 문법]



q1에서 11011을 쌓고 q2에서 역으로 다시 빼줌

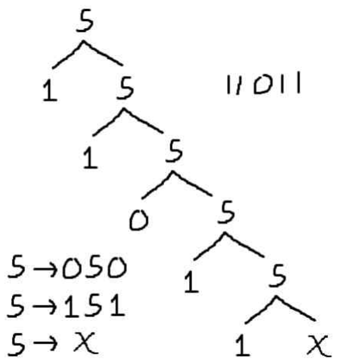
1	→	X
1	→	X
0	→	X
1	→	X
1	→	X
X	→	X

○ **팰린드롬** : 거꾸로해도 똑같은 것 // ε X ∅ : 공문자 / $: 푸쉬(넣다) / 팝(꺼내오다)

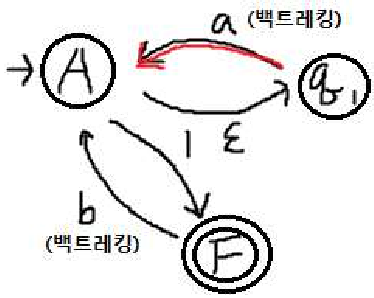
ex) 0, X ->0 이면, 0을 읽으면 공문자를 팝하고 0을 푸쉬하라/ 공문자를 팝하는건 아무일도 안한다는 것

○ **백 트레킹** : 비결정 오타마타에서 오른쪽으로 갔다가 파이널이 없으면 되돌아오는 것

팰린드롬



백 트레킹



❑ 4단계 : 튜링머신(문맥자유 문법(CFG / r.e Language))

문맥자유 문법(CFG) [튜링머신]

$A \rightarrow aB$ 오타마타 형식

$\alpha A \beta \rightarrow \alpha \gamma \beta$ 문법 자유 형식

○ 튜링머신 : 다섯 개의 순서쌍으로 이루어짐

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

Q : 상태들의 유한 집합

Γ : 허용되는 테이프 심볼들의 유한 집합

B : Γ에 속하는 블랭크

Σ : 0, 1을 쓸 것인지, a, b 등을 쓸 것인지

δ : 전이 함수의 집합

q_0 : 시작상태

F : 최종 단계 상태의 집합

// 여담) Halting Problem(정지문제) : 프로그램이 정지하는지 무한루프인지 알 수 있는 알고리즘

13주차(16.06.02)

알고리즘을 통한 문제 해결

❑ 알고리즘 : 입력, 출력, 유한성, 정확성, 확정성, 일반성, 효율성

- 입력, 출력, 유한성, 정확성, 확정성, 일반성, 효율성

○ 알고리즘 계산 복잡도 : 빅오(O)로 표현,

ex) 3n+2 : n의 복잡도 / 2x(제곱)+2 : n(제곱)의 복잡도

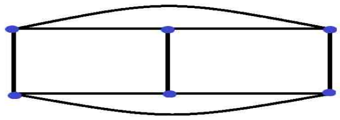
- 순차탐색 알고리즘 : 버블소트
- 분할 점령 알고리즘 : 퀵 소트 // $k = 1+\log_2 n = O(\log_2 n)$

기말고사 시험문제!!!!

1. 다음의 그래프를 인접 행렬과 연결 그래프로 나타내시오

2. 오일러 경로, 오일러 순회, 해밀턴 경로, 해밀턴 순회인지 설명하시오

3. 오일러의 정리 : 항상 v - e + f = 2 이다.
   연결된 평면 그래프에서 꼭지점의 수를 v, 연결선의 수를 e, 면의 수를 f라고 할 때 v-e+f=2이다.
   ex) 꼭지점의 개수 = 6, 연결선의 개수 e=9, 면의 수 f=5(바깥면도 친다), v-e+f=2의 오일러 정리가 성립함.



4. 최단 경로 : 항상 v - e + f = 2 이다.

5. 행렬의 곱 : 왼쪽 행과 오른쪽 열의 곱

$$C_{11} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj}$$

ex) $\begin{pmatrix} 2 & 3 \\ 3 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 3 \\ 4 & 2 \end{pmatrix} = \begin{pmatrix} (2\times1+3\times4) & (2\times3+3\times2) \\ (3\times1+1\times4) & (3\times3+1\times2) \end{pmatrix} = \begin{pmatrix} 14 & 12 \\ 7 & 11 \end{pmatrix}$

6. 가우스 조단 역행렬 구하는 알고리즘 : 절차가 존재함 정 [A | I]

$\begin{bmatrix} 3\,4 | 1\,0 \\ 2\,3 | 0\,1 \end{bmatrix}$ $(\frac{1}{3})* R1 \rightarrow R1$ 문제 : $\begin{bmatrix} 1\,0 | \\ 0\,1 | \end{bmatrix}$로 만들어줘야함.

$\begin{bmatrix} 1 & \frac{4}{3} | & \frac{1}{3} & 0 \\ 2 & 3 | & 0 & 1 \end{bmatrix}$ $(-2)* R1 + R2 \rightarrow R2$

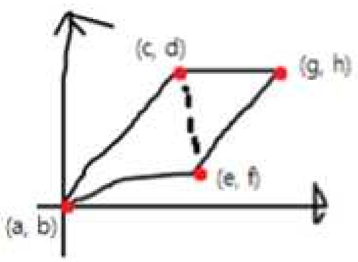
$\begin{bmatrix} 1 & \frac{4}{3} | & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} | & -\frac{2}{3} & 1 \end{bmatrix}$ $3 * R2 \rightarrow R2$

$\begin{bmatrix} 1 & \frac{4}{3} | & \frac{1}{3} & 0 \\ 0 & 1 | & -2 & 3 \end{bmatrix}$ $(-\frac{4}{3})* R2 + R1 \rightarrow R1$

$\begin{bmatrix} 1\,0 | & 3 - 4 \\ 0\,1 | & -2\ \ 3 \end{bmatrix}$

1. 첫 번째 줄에 1/3을 전부 곱해줌 -> 첫 번째 줄에 값 입력 -> 1*1이 1이 됨
2. 첫 번째 줄에 -2를 전부 곱하고 R2에 전부 더해줌 -> 두 번째 줄에 값 입력 -> 2*1이 0이 됨
3. 두 번째 줄에 3을 전부 곱해줌 -> 두 번째 줄에 값 입력 -> 2*2가 1이 됨
4. 두 번째 줄에 (-4/3)을 곱하고 첫 번째 줄을 더해줌 -> 첫 번째 줄에 값 입력 -> 1*2가 0이 됨

7. 4개의 좌표 값을 임으로 정하고 이것을 이용해 행렬식을 구하고 넓이를 구하라.!!!!! (삼각형 외적구해서 구함)
임의의 데카르트 2차원 좌표에 임의의 4개를 좌표를 그리고 사각형의 넓이를 행렬식으로 구하라!!



사각형 넓이 : (a,b) (c,d) (e,f) (g,h) 일 때

(1) x | 0 c e | y | 0 d f |

(2) x | c e g | y | d f h |

// a_{ij} : $(-1)^{i+j}$

1/2 $|(-1)^{1+1} * (cf - ed) + (-1)^{1+2} * (0f - e0) + (-1)^{1+3} * (0d - c0)|$ +
1/2 $|(-1)^{1+1} * (eh - gf) + (-1)^{1+2} * (ch - gd) + (-1)^{1+3} * (cf - ed)|$

기말고사 시험문제2!!!!

8. 출력이 없는 유한 오타마타를 그래프로 입력하시오.

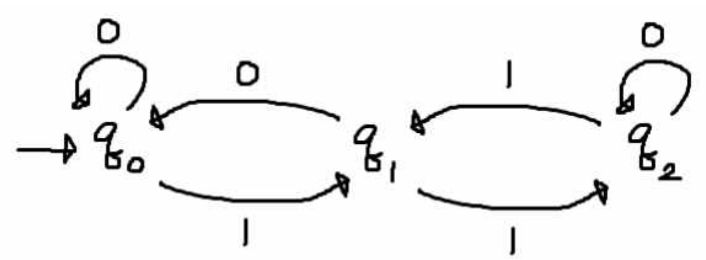
○ 출력이 없는 유한 오타마타

DFA $M = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\})$ 일 때,

δ : $\delta(q_0, 0) = q_0$, $\delta(q_0, 1) = q_1$

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

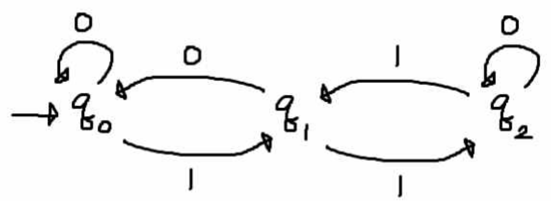
$\delta(q_2, 0) = q_2$, $\delta(q_1, 1) = q_1$ 일 때, 그래프로 나타내면



해석 : $\{q_0, q_1, q_2\}$의 집합이 있고, {0, 1}의 숫자를 사용, q_0으로 시작하여, $\{q_1\}$이 최종단계

9. 8추가로 0011100 등 숫자를 주었을 때, 인식하는 문자인지 아닌지 설명.

10. 8번의 유한상태기계의 그래프를 행렬로 나타내어라.
유한상태기계를 나타내는 매트릭스를 작성하라.



->

	q_0	q_1	q_2
q_0	(T, 1)	(T, 1)	(F, ╳)
q_1	(T, 0)	(F, ╳)	(T, 1)
q_2	(F, ╳)	(T, 1)	(T, 0)

//T,F로 나타내고, 필요한 푸쉬 입력

11. 튜링머신 : 다섯 개의 순서쌍으로 이루어짐 (많아도 2문제)

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

Q : 상태들의 유한 집합

Γ : 허용되는 테이프 심볼들의 유한 집합

B : Γ에 속하는 블랭크

Σ : 0, 1을 쓸 것인지, a, b 등을 쓸 것인지

δ : 전이 함수의 집합

q_0 : 시작상태

F : 최종 단계 상태의 집합
