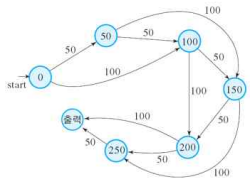
13.1 오토마타란 무엇인가?

교과목명	이산수학	분반		담당교수	김 외 현
학부(과)		학번		성명	

정의 **오토마타**

- 디지털 컴퓨터의 수학적 모델인 오토마톤의 복수형으로서 '자동기계 장치'란 뜻을 가짐
- 일반적으로 입력장치, 출력장치, 저장장치, 제어장치를 가지고 있으므로 현대적인 디지털 컴퓨터가 작동하는 이론적인 메커니즘임
- 단순한 형태의 오토마타는 기원전 3천 년 무렵부터 만들어졌는데, 고대 이집트인들이 사용했던 모래시계나 물시계 등도 넓은 의미의 오토마타임
- 일반적인 뻐꾸기 시계는 정해진 시각이 되면 뻐꾸기가 안에서 튀어나와 울지만, 요즘의 뻐꾸기 시계는 빛을 감지할 수 있는 센서를 통하여 밤이 되면 울지 않는 기능도 가진 여러 가지 형태의 시계 오토마타를 나타냄

참고 **간단한 자판기 오토마타의 다이어그램**



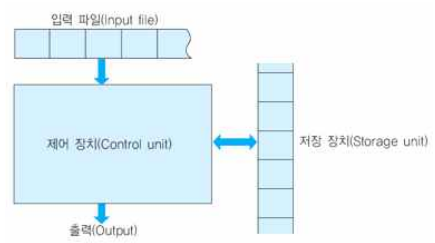
정리 **오토마타의 특성**

(1) 오토마타는 입력 데이터를 읽을 수 있는 입력 기능을 가지고 있음

(2) 오토마타는 특정 형태의 출력 기능을 가지고 있음

(3) 오토마타는 무한개의 셀들로 이루어진 임시 저장 장치를 가질 수 있음

(4) 오토마타는 유한개의 내부 상태를 제어할 수 있는 제어 장치를 가지고 있음



참고 **오토마타의 기본 개념**

- 오토마타는 이산적인 시간 단위로 작동됨을 기본 가정으로 함
- 어느 특정 시각에 제어 장치는 특정의 상태에 놓여 있음
- 입력 기능은 입력 파일상의 어떤 특정한 심볼을 읽음
- 제어 장치의 다음 상태는 전이 함수에 의해 결정됨
- 전이 함수는 현재의 제어 상태, 입력 심볼, 임시 저장 장치 내의 정보들에 의해 결정됨

정의 **결정적 오토마타**

전의에 의한 다음 상태가 현재의 형상에 따라 유일하게 결정되는 오토마타

정의 **비결정적 오토마타**

어떤 상태에서 하나의 입력을 받았을 때 다음 상태가 2개 이상인 오토마타

참고 **출력 여부에 따른 오토마타**

(1) 인식기

주어진 입력에 대해 인식하거나 기각할 수 있는 기능만을 가짐

(2) 변환기

주어진 입력에 대응하는 새로운 문자열을 출력할 수 있는 기능도 가짐

상태에 따른 출력을 내는 무어기계, 전이에 따른 출력을 내는 밀리 기계 등이 있음

예제

1. 오토마타에서 일반적으로 가지고 있지 않은 장치는?

① 입력장치

② 출력장치

③ 계산장치

④ 제어장치

13.2 오토마타 학습의 필요성과 유한 상태 시스템

교과목명	이산수학	분반		담당교수	김 외 현
학부(과)		학번		성명	

참고 **오토마타를 학습하는 주된 이유**

(1) 오토마타 이론이 전산학이나 전자공학 등의 학문을 이해하는데 있어서 필요한 기본적인 개념과 원리를 제공하고, 여러 가지 응용 분야에 적용할 수 있기 때문임

(2) 오토마타 이론을 통하여 컴퓨터와 관련된 이론적인 바탕과 작동의 원리를 보다 정확하게 이해함으로써 하드웨어나 소프트웨어 각 분야에 대한 직관을 넓힐 수 있으며, 보다 포괄적인 통찰력을 제공해줌

(3) 오토마타 이론의 직접적인 응용. 이를 통하여 디지털 컴퓨터의 디자인, 프로그래밍 언어, 컴파일러, 신경생리학, 통신, 신경망, 언어론 등 다양한 분야에 직접 활용할 수 있음
특히 컴파일러나 프로그래밍언어를 정확하게 이해하기 위해서는 오토마타 이론의 이해가 필수적임

(4) 추상적 모델의 개념적 이해. 복잡한 현상들을 간략하고 정확하게 추상화시킴으로써 연구의 폭을 넓히고 정교한 학문적 탐구가 가능해짐

정의 **유한 상태 시스템(FSM)**

유한개의 상태를 가진 오토마타

참고 **유한 상태 시스템의 응용**

(1) 엘리베이터의 제어
엘리베이터의 제어 방식은 탑승자들이 과거에 요청했던 모든 요구들을 기억하지 않고, 단지 현재의 층수와 엘리베이터의 운동 방향 및 탑승자들이 요청한 것들 중에서 아직까지 이루어지지 않은 요구들만을 기억하고 있는 유한 상태 시스템임

(2) 컴퓨터의 제어 장치와 같은 논리 회로
논리 회로는 통상 0과 1로 표시되는 2가지 조건 중 하나의 상태인 유한개의 게이트들의 집합으로 구성됨. 따라서 n개의 게이트를 가진 논리 네트워크의 상태는 2^n개 중 하나. 이런 점에서 논리 회로는 유한 상태 시스템임

(3) 문서 편집기와 어휘분석기에도 적용됨
문서 편집기는 입력을 추가할 때마다 상태가 변화되고, 어휘분석기는 컴퓨터 프로그램에서 기호들을 차례로 읽어서 문자 스트링들을 변수, 상수, 예약어 등으로 대응시킴. 이 과정에서는 단지 유한개의 정보를 기억하기만 하면 됨. 이와 같이 유한 상태 시스템은 오토마타 이론을 적용하여 여러 가지 형태의 스트링들을 효과적으로 처리하는 데 있어서 매우 유용함

(4) 컴퓨터도 일종의 유한 상태 시스템임
컴퓨터는 중앙처리장치, 주기억장치, 보조기억장치 등 상태의 개수는 매우 많지만 여전히 유한개의 상태로 이루어져 있음. 여기에서 우리는 한정된 수의 디스크, 드럼, 테이프 등이 사용된다고 가정함

예제

2. 다음 중 오토마타 이론을 비교적 직접 이용하지 않은 것은?
① 컴파일러
② 프로그래밍 언어
③ 언어론
④ 수학의 증명

3. 다음 중 오토마타의 응용과 관련이 가장 적은 것은?
① 컴파일러
② 문서편집기
③ 행렬
④ 엘리베이터

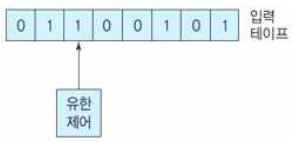
13.3 유한 오토마타

교과목명	이산수학	분반		담당교수	김 외 현
학부(과)		학번		성명	

정의 유한 오토마타의 기본 개념

(1) 유한 오토마타
- 이산적인 입력과 출력을 가지는 시스템의 수학적 모형임
- 이 시스템은 유한한 개수의 내부 상태들을 가지고 있는데, 시스템의 상태는 다음에 들어올 입력에 대한 시스템의 행동을 결정하는데 필요한 과거의 정보들을 요약함
- 인식기로서의 유한 오토마타를 고찰함
- 유한 오토마타의 주요 특징으로는 임시 저장 장치를 가지지 않는다는 점과 입력 테이프는 단지 읽힐 수만 있으며 그 내용이 변경될 수 없다는 점임
- 유한 오토마타의 기능은 작동 과정에서 정보를 기억하는 데 있어서 상당한 제약을 받게 됨

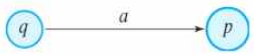
(2) 유한 제어
- 유한 오토마타의 입력과 상태들을 제어함
- 유한 오토마타는 여러 종류의 오토마타들 중에서 가장 단순한 형태로서 입력과 유한 제어를 가진 유한 오토마타를 나타냄



(3) 전이 함수
- 유한 오토마타는 유한한 상태들의 집합과 전이 함수들의 집합으로 구성됨
- 전이란 알파벳 ∑로부터 선택된 입력 기호에 의해 생기는, 상태에서 상태로의 변화임
- 입력에 따라 상태는 항상 변할 수 있으며, 원래의 상태로 다시 돌아가는 전이도 있음

(4) 시작 상태, 최종 상태
- q_0로 나타내는 상태를 시작 상태라 하는데, 이 시작 상태에서 오토마타가 시작됨
- 어떤 상태들은 최종 상태 또는 인식 상태들로 지정되는데, 그래프에서는 이중의 서클로 표시

(5) 전이 다이어그램
- 각 유한 오토마타마다 전이 다이어그램이라 불리는 방향이 있는 그래프는 전이 다이어그램과 같이 그려짐



- 전이 다이어그램의 각 노드들은 유한 오토마타의 상태에 해당
- 어떤 상태 q에서 입력 a를 받아서 어느 상태 p로 가는 변환이 있다면 전이 다이어그램에서는 상태 q에서 상태 p로 가는 a라는 연결선이 존재함
- 입력 스트링 x에 대해 시작 상태에서 최종 상태로 가는 경로가 존재한다면 유한 오토마타는 스트링 x를 인식하게 됨

(6) 정규 언어
- 유한 오토마타는 전이 방법에 따라 결정적 유한 오토마타와 비결정적 유한 오토마타로 구분됨
- 이 둘은 기능에 있어서는 동치이며 유한 오토마타에 대응하는 언어는 정규 언어임

정의 결정적 유한 오토마타(DFA)

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

$\Rightarrow$ Q: 상태들의 유한집합

Σ: 입력 기호들의 유한집합

δ: $Q \times \Sigma \rightarrow Q$인 전이 함수

즉, $\delta(q_i, a) = q_j$

q_0: $q_0 \in Q$인 시작 상태

F: $F \subseteq Q$인 최종 상태의 집합

참고 **DFA의 작동 개요**

- 처음에는 시작 상태가 q_0에 있고 유한 제어는 입력 스트링의 가장 왼쪽에 있는 심볼을 가리킴
- 오토마타의 작동에 따라 입력장치에서 한 기호씩 오른쪽으로 이동하면서 상태가 바뀜
- 스트링을 모두 읽고 난 후 DFA가 최종상태에 있으면 그 스트링이 '인식되고' 그렇지 않으면 '기각된다'.
- 입력 메커니즘은 왼쪽에서 오른쪽으로의 방향으로만 이동이 가능하며 각 단계에서 하나씩의 기호만을 읽을 수 있다는 점에 유의

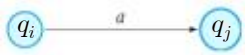
정의 **전이 그래프**

원: 상태, 연결선: 전이

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

$\Rightarrow$ 상태 개수$=|Q|$

if $q_i \in Q$에 대해 $\delta(q_i, a)=q_j$



최종 상태인 경우에는 두 개의 서클로 표시

참고 **출력이 없는 유한 오토마타**

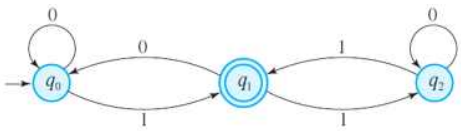
DFA $M=(\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\})$

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

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

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

$\Downarrow$



$\Rightarrow$ 스트링 $001, 01, 0101, 1101$ 등은 인식됨

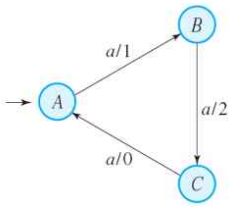
$00, 110, 011$ 등은 기각됨

예제

4. 짝수 개의 b를 가지는 $\Sigma=\{a, b\}$상의 모든 스트링들 인식하는 DFA를 디자인해보자.

참고 **출력이 있는 유한 오토마타**

- 어떤 입력의 개수를 읽는 순간마다 알 수 있는 것이 카운터란 장치
- Mod-3 카운터: 어떤 수를 3으로 나누어서 나머지를 출력해내는 오토마타



- 상태 A에서 a가 들어오면 1을 출력하면서 상태 B로 이동하고, 상태 B에서 입력 a가 들어오면 2를 출력하면서 상태 C로 이동함
- 상태 C에서 a가 들어오면 0을 출력하면서 상태 A로 이동하게 되며 추가적인 입력이 있을 경우에는 위의 과정을 반복함

참고 **유한 오토마타로 변수를 만드는 방법**

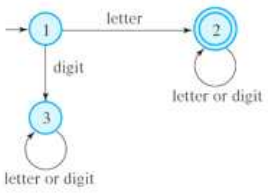
- C언어에서 모든 변수들의 집합은 하나의 언어
- 각 변수는 하나의 문자로 시작되어야 하며, 문자 다음에는 임의 개수의 문자나 숫자가 혼용하여 사용됨
- 다음의 문법은 변수에 대한 정확한 정의를 나타내는 규칙들임

```
<id>      →   <letter><rest>
<rest>    →   <letter><rest> | <digit><rest> | λ
<letter>  →   a | b | ··· | z | _
<digit>   →   0 | 1 | ··· | 9
```

$\Rightarrow$ 변수: <id>, <letter>, <rest>, <digit>

기호: $a, b, \cdots, z, \_, 0, 1, \cdots, 9$

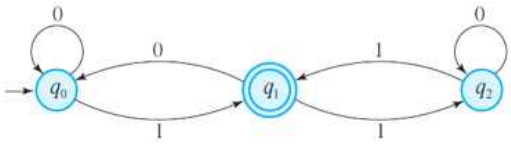
참고 **C언어의 변수를 인식하는 오토마타**



예제

5. 문제 해결을 위한 유한 오토마타의 예로 다음과 같이 강을 건너는 문제를 살펴보자. 어떤 사람이 늑대, 염소 그리고 양배추와 더불어 강의 왼쪽 기슭에 있다고 생각하자. 사람은 한 번에 늑대나 염소 또는 양배추 중 하나만 선택하여 강의 왼쪽 기슭이나 오른쪽 기슭을 왕복할 수 있다. 물론 사람 혼자서 건널 수도 있다. 만양 사람이 늑대와 염소를 어느 한쪽 기슭에 같이 남겨 둔다면 늑대는 사람이 없는 틈을 타서 양배추를 먹어 치우게 될 것이다. 그렇다면 어떻게 염소나 양배추가 먹히지 않고 사람에 의해 강을 무사히 건너갈 수 있을까?

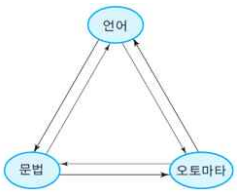
6. 다음의 스트링 0001, 01001, 0000110 중 어떤 것들이 다음과 같은 DFA에 의해 인식되는지를 판단해보자.



13.4 형식 언어와 문법

교과목명	이산수학	분반		담당교수	김 외 현
학부(과)		학번		성명	

참고 **오토마타 이론에서 가장 중요한 3가지 개념**



(1) **언어**

의사소통의 수단으로 자연어와 형식언어로 분류

정의 **알파벳**

Σ: 공집합이 아닌 기호들의 유한집합

정의 **문자열(스트링)**

알파벳 Σ에 있는 기호들을 나열한 유한수열

보기 $\Sigma = \{a, b\}$: 알파벳

$\Rightarrow$ $aa, ab, abab, bbaa \cdots$: 문자열

정의 **팰린드롬**

앞에서 읽으나 뒤에서 읽으나 똑같은 문자열

참고 **문자열의 연산**

(1) 연결(접합)

두 개의 문자열을 연결해 새로운 문자열을 만드는 연산

$w = a_1a_2a_3 \cdots a_n$, $v = b_1b_2b_3 \cdots b_n$

$\Rightarrow$ $wv = a_1a_2a_3 \cdots a_nb_1b_2b_3 \cdots b_n$

(2) 역

문자열 w를 이루는 요소들의 역순

$w = a_1a_2a_3 \cdots a_n$

$\Rightarrow$ $w^R = a_n \cdots a_3a_2a_1$

(3) 문자열의 길이

$|w|$= (문자열 w를 이루는 기호의 개수)

(4) 공문자열

λ: 공문자열 $\Leftrightarrow$ $|\lambda| = 0$

$\lambda w = w = w\lambda$

(5) 문자열의 반복

$w^n = \underbrace{www \cdots w}_{n}$, $w^0 = \lambda$

(6) Σ의 클리니 클로저, 포지티브 클로저

$\Sigma^* = \{\lambda, \Sigma$상의 기호들의 결합으로 만들어지는 모든 문자열$\}$

$\Sigma^+ = \Sigma^* - \{\lambda\}$

(7) 언어

L: Σ^*의 부분집합

보기 $\Sigma = \{a, b\}$: 알파벳

$\Rightarrow$ $L = \{a, ab, abab\}$: 언어

예제

7. 두 문자열 $u =$ race와 $v =$ car에 대해 uv와 vu를 구해보자.

8. 문자열 $w =$ dog의 역인 w^R을 구해보자.

9. 두 문자열 $w_1 =$ orange와 $w_2 =$ racecar의 길이를 구해보자.

10. 알파벳 $\Sigma = \{a, b\}$에 대해 Σ^*와 Σ^+를 구해보자.

(2) 문법

배커스-나우어 표기법(BNF)

문맥자유 문법(CFG)

정의 **형식 문법**

$G=(N, T, P, S)$

(1) N: 논터미널 기호들의 유한집합

(2) T: 터미널 기호의 유한집합

$N\cap T=\varnothing$, $N\cup T=\mathrm{V}$ (V:문법 기호)

(3) P: 생성 규칙들의 유한집합

$\alpha\rightarrow\beta$, $\alpha\in\mathrm{V}^+$, $\beta\in\mathrm{V}^*$

(4) S: N에 속하는 시작 기호

정의 **유도**

$\exists\ \alpha\rightarrow\beta$: 생성규칙, $\gamma, \delta\in\mathrm{V}^*$

$\gamma\alpha\delta\Rightarrow\gamma\beta\delta$

$\Rightarrow$(유도)는 α가 β로 대체되었음을 의미

$\overset{*}{\Rightarrow}$: 영 번 이상의 유도

$\overset{+}{\Rightarrow}$: 한 번 이상의 유도

정의 $G=(N, T, P, S)$: 문법

$L(G)=\{w\mid S\overset{*}{\Rightarrow}w, w\in T^*\}$

: G에 의해 생성된 언어

정의 **문장형태, 문장**

(1) w: 문장형태 $\Leftrightarrow$ $S\overset{*}{\Rightarrow}w$, $w\in\mathrm{V}^*$

(2) w: 문장 $\Leftrightarrow$ $S\overset{*}{\Rightarrow}w$, $w\in T^*$

정의 **동치**

G_1, G_2: 동치 $\Leftrightarrow$ $L(G_1)=L(G_2)$

참고 **4가지 타입의 오토마타**

문법	언어	오토마타
Type 0 (무제한 문법)	r.e Language	튜링머신
Type 1 (문맥민감 문법)	CSL	선형제한 오토마타
Type 2 (문맥자유 문법)	CFL	푸시다운 오토마타
Type 3 (정규 문법)	Regular Language	유한 오토마타

참고 **촘스키 포함 관계**

예제

11. 문법 G에 대하여 다음을 구해보자.

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

$P: S\rightarrow aA \quad S\rightarrow bS$

$A\rightarrow aA \quad A\rightarrow bA \quad A\rightarrow b$

(1) 문법 G와 생성 규칙 P가 다음과 같을 때 논터미널 기호, 터미널 기호, 시작 기호를 구하고, 생성 규칙의 개수를 구해보자.

(2) 문법 G에서 ab, abb, baab 등이 생성되는지 유도해보자.

(3) (2)의 abb에서 문장 형태와 문장을 구해보자.

12. 다음과 같은 문법 G로부터 생성되는 언어를 구해 보자.

$G = (\{S, A, B, C\}, \{a, b\}, P, S)$

$P : S \rightarrow A$

$A \rightarrow abC \mid aABC$

$bB \rightarrow bbb$

$bC \rightarrow bb$