

# 인공지능 워크북

## 학습 주안점

인공지능은 컴퓨터에게 지능적인 문제해결 능력을 부여하기 위한 제반 학문 분야를 통칭하는 것이다. 본 과목에서는 문제풀이 방식, 지식의 표현 및 사용, 퍼지이론, 전문가 시스템, 컴퓨터 시각, 패턴인식, 학습과 신경회로망 등 인공지능의 전반적인 분야에 대한 기본적인 이해를 하는 것을 목표로 하고 있다.

이 워크북은 인공지능 과목 학습을 돕기 위한 최소한의 학습내용 요약과 보조 학습내용을 담고 있어 교과목 전체의 내용이 포함되어 있지는 않다. 그러므로 지나치게 워크북 내용에만 의존하지 말고 각 장별 학습지침에 포함된 내용도 교재를 통해 학습하기 바란다.

중간시험은 교재 제1~3장(매체강의 1~4강) 범위에서 출제하며, 주로 탐색에 의한 문제풀이 방법의 개념과 주요 알고리즘을 이해하고 있는지를 평가하기 위한 문제와 인공지능의 기본 개념과 탐색에 대한 주요 용어에 대한 지식을 평가하는 문제가 출제된다.

기말시험은 교재 제4~10장(매체강의 5~15강) 범위에서 출제하며, 전 범위에 걸쳐 고루 출제된다. 강의를 중심으로 하여 각 강별 학습지침에 따라 학습하기 바란다.

## 제1장 인공지능의 개요

### ▶ 주요 학습내용

- 지능적 시스템을 구현하기 위한 접근방법의 이해
- 인공지능의 주요 분야
- 지능형 에이전트의 기본구조

### ▶ 학습지침

제1장은 인공지능 분야에서 지능의 구현방식에 대한 기본적인 아이디어를 소개하는 것이 주요 목적이다. 일반적인 정보처리 방식처럼 효율적인 알고리즘으로 정형화하기 어려운 문제에 어떻게 접근하고 있는지 살펴보기 바란다.

### 1. 생각하는 컴퓨터

- 지능이란 무엇인가?
  - ‘배우고 이해할 수 있는 능력, 새로운 상황에 대처할 수 있는 능력’ - 『웹스터 사전』
- 인공지능(artificial intelligence)
  - 지능적 행동의 일반적 의미에 대한 컴퓨터 관점에서의 이해 및 지능적 행동을 할 수 있는 인공물의 생성을 다루는 컴퓨터과학 및 컴퓨터공학의 분야
  - 계산심리학 관점 및 기계지능의 관점에서의 접근이 함께 이루어짐
  - 이 과목에서는 기계지능의 관점에서 접근함 : 컴퓨터를 이용하여 프로그래밍할 수 있는 영역을 인간이 수행할 수 있는 작업영역으로 확장하는 것에 목적을 둠

### 2. 인간과 지능

- 물리적 기호 시스템 가설 (Newell과 Simon, 1976) : 인공지능 시스템에서의 기본적인 가설
 

*물리적 기호 시스템은 일반적인 지능적 행동을 위한 필요 충분한 수단을 가지고 있다.*

  - ※ 물리적 기호 시스템 가설이 갖는 의미 : 인간이 행하는 지능적 작업을 수행하는 프로그램을 작성할 수 있다는 가능성에 대한 믿음의 근원
- 지능적 시스템 : **지식**을 바탕으로 하여 다음의 능력이 요구됨
  - 학습을 통해 새로운 지식을 습득함
  - 외부의 환경을 인식함으로써 문제를 **이해**할 수 있음
  - 지식을 기반으로 **추론**함으로써 대상문제의 해를 찾아냄

### 3. 인공지능 시스템

- 문제풀이 시스템 : 인공지능의 추론능력에 의하여 문제를 풀이
- 패턴이해 시스템
  - 외계의 사상(事象)을 보고 들을 수 있게 하는 시스템
  - 외부의 패턴을 인식하여 그 의미를 이해하고 판단
- 지식 베이스 : 지식을 구조화하여 저장
  - 지식공학 : 지식을 어떻게 체계화하고 지식 베이스에 축적하며, 축적된 지식을 어떻게 이용하는가를 연구하는 학문

#### 4. 학습하는 인공지능의 필요성

- 학습을 통한 지식 축적
  - 지도를 통한 지식의 축적
  - 행동에 대한 긍정적·부정적 자극을 통해 더 좋은 반응을 받을 수 있도록 지식을 강화
  - 컴퓨터가 스스로 이러한 과정을 통해 지식을 습득할 수 있도록 하는 것이 필요함

#### 5. 인공지능의 역사

- 지식공학
  - DENDRAL이라는 전문가 시스템 이후 인간의 사고에 대해 폭넓은 일반 법칙을 찾으려는 시도로부터 제한되고 특수한 분야의 지식을 획득해서 조직화하고 그것을 이용한다는 태도로 전환

#### 6. 지능형 에이전트

- 에이전트란 무엇인가?
  - 누군가를 위하여 정해진 일을 대신 해 주는 컴퓨터 프로그램을 통칭한 것
- 지능형 에이전트의 특성
  - 자율성 : 사용자를 위해 자율적으로 행동할 수 있음
  - 반응성 : 환경의 변화에 대해 시간적으로 적절하게 응답
  - 능동성 : 주도권을 가지고 목표지향적으로 행동함
  - 지속성 : 지속적으로 실행하는 데몬(demon)과 같은 프로세스
  - 사회성 : 사용자나 다른 에이전트와 상호작용을 할 수 있음
  - 적응성 : 스스로 배우거나 명시적으로 지시를 받는 과정을 통해 개별 사용자에게 맞게 적응할 수 있음
- 지능형 에이전트의 일반적 구조

- 에이전트는 자신이 속한 환경을 지속적으로 감지하여 상황에 맞는 적절한 행위를 함으로써 환경에 영향을 미침
- 감지기 : 소속된 환경의 상황을 입력
  - ⇒ 환경의 상태를 인식
  - ⇒ 정해진 방식에 따라 조건을 만족하는 규칙을 선택
  - ⇒ 효과기를 통해 행위를 취함으로써 소속된 환경에 영향을 줌

```

function Agent
  Environment e;           // 환경
  RuleSet r;              // 규칙의 집합

  while (true) do                // 지속적인 동작
    state = senseEnvironment(e);   // 환경의 감지
    a = chooseAction(state, r);    // 행동의 결정
    a.applyAction(e);             // 환경에 대해 결정한 행동을 실행
  end-while;
end-function;

```

● 목표기반 에이전트

- 자신이 놓여 있는 환경과 추구하는 목표 그리고 가능한 행동에 대한 명시적인 기호모델 (symbolic model)을 가지고 있다.
- 이 모델을 바탕으로 논리적 추론을 전개함으로써 목표를 달성하기 위해 현재 자신이 수행해야 할 행동을 결정

## 제2장 문제풀이

### ▶ 주요 학습내용

- 인공지능 문제의 특징
- 문제의 표현방법
- 문제풀이 방법 : 탐색
- 문제풀이 방법 : 문제축소

### ▶ 학습지침

제2장에서는 인공지능 문제풀이의 기본적인 접근방법인 탐색과 문제축소의 개념을 이해한다. 상태, 연산자, 상태공간 등의 용어를 익혀 둔다.

### 1. 인공지능과 문제

#### ● 문제풀이

- 주어진 문제를 해결하는 일반적인 방법을 연구하는 것

#### ● 인공지능 문제의 특징

- 탐색형 추론문제 : 알고리즘화하기 어려운 비정형적인 문제에 대하여, 해에 도달하는 수많은 경로를 시행착오적·경험적 방법을 동원하여 적절한 해를 찾음
- 주어진 문제의 상태 및 문제의 풀이를 위한 연산 등을 컴퓨터에서 처리할 수 있는 형태로 나타내야 함

### 2. 문제의 표현

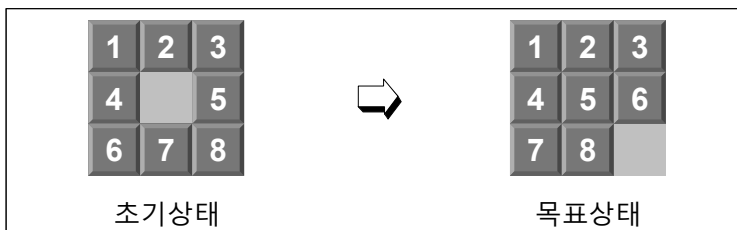
#### ● 상태(state) : 문제풀이 과정 중 어느 한 지점에서의 문제의 형태

#### ● 문제표현의 요소

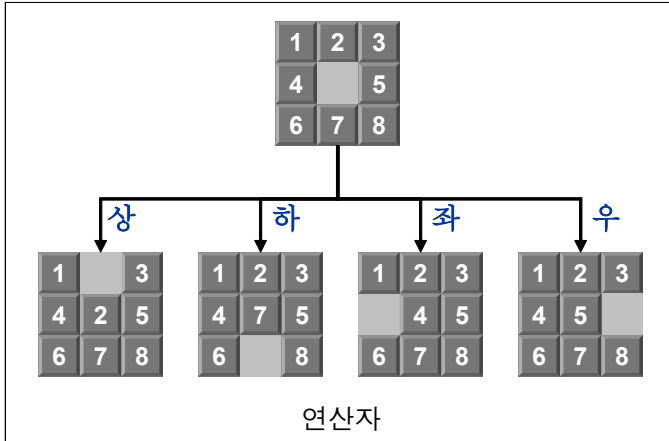
- 초기상태 : 주어진 문제를 나타내는 상태
- 목표상태 : 주어진 문제가 풀이된 결과를 나타내는 상태
- 연산자 : 어떠한 상태가 주어졌을 때 이를 새로운 상태로 변화시키는 역할을 하는 것

#### ● 문제표현의 예 : 8-퍼즐 문제

- 초기상태 및 목표상태



- 연산자 : 8개의 조각 중 하나를 움직이는 것
  - 조각을 움직이는 대신 빈칸을 이동시키는 것으로 표현할 수 있음



● 상태묘사

- 문제의 상태를 컴퓨터로 처리하기 위한 적절한 자료구조로 표현한 것
- 기호 열, 벡터, 다차원 배열, 트리, 리스트 등 문제에 따라 상태를 표현하는데 보다 자연스러운 표현방법을 사용
- 연산자를 효율적으로 정의할 수 있는 표현방법 사용
- 8퍼즐 문제의 예 : 2차원 배열 사용

<table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td><td>6</td></tr> <tr><td>7</td><td>8</td><td> </td></tr> </table>	1	2	3	4	5	6	7	8		<pre>struct PuzzleState{     int    blankX, blankY;     char  board[3][3]; };</pre>
1	2	3								
4	5	6								
7	8									

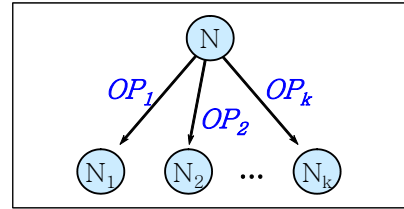
● 연산자의 정의

- 하나의 상태묘사를 다른 상태묘사로 변화시키는 일종의 연산능력을 지닌 함수
- 8-퍼즐 문제의 예 : 빈칸을 상향 이동하는 연산자

```
int opMvBlankUp(PuzzleState *s)
{
    if (s->blankY > 0) {
        s->board[s->blankX][s->blankY] =
            s->board[s->blankX][s->blankY-1];
        s->board[s->blankX][--s->blankY] = ' ';
        return 1;
    }
    else return 0;
}
```

● 상태공간(state space)

- 초기상태에서부터 정의된 연산자를 적용하여 만들어질 수 있는 모든 상태의 집합
- 일반적으로 방향성 그래프의 형태로 표현됨
- 노드 : 상태묘사를 나타냄
- 아크 : 연산자를 나타냄



● 문제의 정의 : 초기상태의 묘사를 정해진 방법으로 변화시켜 목표상태의 묘사로 바꾸는 것

↳ 문제 = (초기상태, 목표상태, 연산자 집합)

↳ 문제풀이 : 초기상태에서 시작하여 목표상태가 되도록 상태를 변화시키는 연산자의 적용순서를 찾는 것

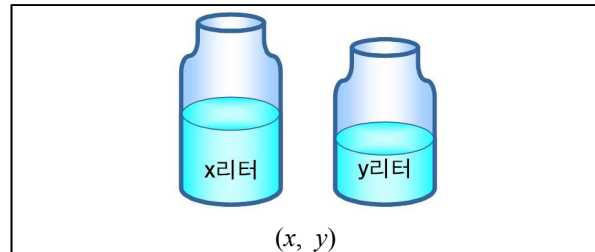
**3. 문제 표현의 예 - 물병문제**

**물병문제**

4리터들이 물병과 3리터들이 물병이 한 개씩 있다. 이 병에는 양을 측정할 수 있는 눈금이 없으며, 항상 물을 공급할 수 있는 펌프가 있다. 4리터들이 물병에 정확히 2리터의 물을 채워라.

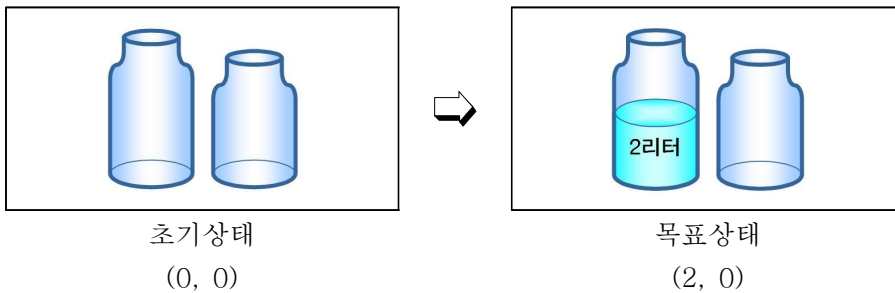
● 상태묘사

- 상태 : 4리터들이 및 3리터들이 물병에 들어 있는 물의 양
- ⇒ 두 통의 물의 양을 나열하여 상태를 묘사할 수 있음
- ⇒ 벡터  $(x, y)$



□ 초기상태 및 목표상태

- 초기상태 : 두 물병이 모두 비어 있는 상태 ⇨  $(0, 0)$
- 목표상태 : 4리터들이 물병에 2리터의 물이 있고, 3리터들이 물병은 비어 있는 상태 ⇨  $(2, 0)$

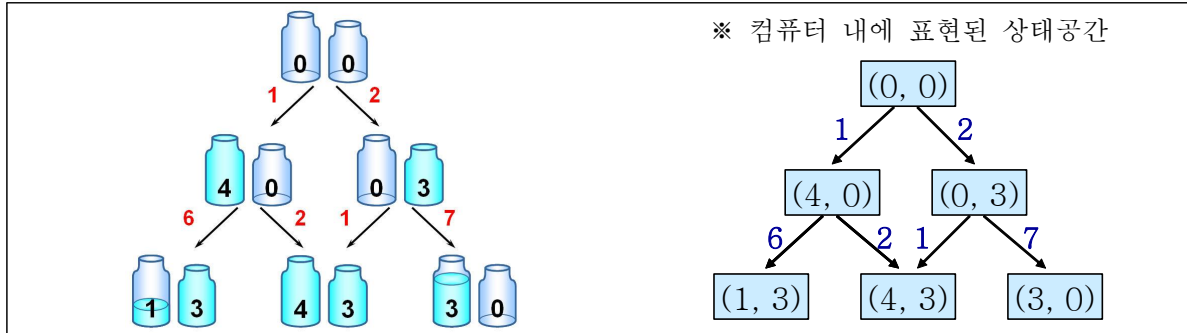


● 연산자의 정의

- 눈금이 없는 물병이므로 임의의 양의 물을 넣거나 옮기거나 버리는 연산은 의미가 없음
- 조건지정 : 불필요한 연산자를 반복 선택하여 적용하려는 노력을 줄일 수 있음
- 연산자 : 교재의 <표 2-1> 참조

● 상태공간

- 현재의 상태에 적용할 수 있는 연산자들을 이용하여 후계상태를 생성
- 초기상태로부터 생성될 수 있는 모든 상태가 연결된 그래프를 형성



4. 문제표현의 예 : 하노이 탑 문제

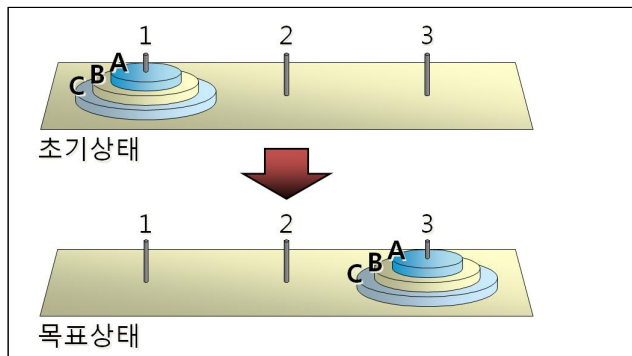
**하노이 탑 문제**  
 세 개의 말뚝 1, 2, 3이 있다. 1번 말뚝에 세 개의 구멍이 뚫린 각기 다른 크기의 원판 A, B, C가 있다. 가장 큰 원판 C는 제일 밑에 놓이고, 가장 작은 원판 A는 제일 위에 놓여 있다. 세 장의 원판을 모두 3번 말뚝으로 옮겨라. 단, 한 번에 어느 말뚝이든 제일 위에 있는 한 장의 원판만 옮길 수 있고, 큰 원판이 작은 원판의 위에 놓일 수 없다.

● 상태묘사

- $(i, j, k)$  : 가장 작은 원판 A가 말뚝  $i$ 에, 원판 B가 말뚝  $j$ 에, 가장 큰 원판 C가 말뚝  $k$ 에 놓여 있는 상태

● 초기상태와 목표상태

- 초기상태 = (1 1 1)
- 목표상태 = (3 3 3)



● 연산자

- 이동(ID, s, d), ID : 옮기고자 하는 원판  
 s : 원판의 현 위치  
 d : 원판을 옮길 목적지 말뚝



### 5. 상태공간 탐색에 의한 문제풀이

● 탐색에 의한 문제풀이

- 상태공간을 탐색하여 초기상태에서부터 목표상태로 이르는 경로를 찾음으로써 문제를 풀이
- 초기상태, 목표상태, 연산자로 문제를 정의
- 주어진 상태에 적용 가능한 연산자를 적용하여 차기상태를 생성함
- 미리 정의된 기준에 의거하여 조사되지 않은 차기상태의 집합에서 하나를 선택함
  - ↳ 정의된 기준을 만족하는 목표노드까지 도달하는 경로가 발견되면 문제풀이가 됨

※ 상태공간 탐색방법의 특성

- 시행착오를 반복하여 목표상태에 도달하는 경로를 탐색
- 시행착오를 최소화할 수 있는 효율적인 탐색방법이 중요

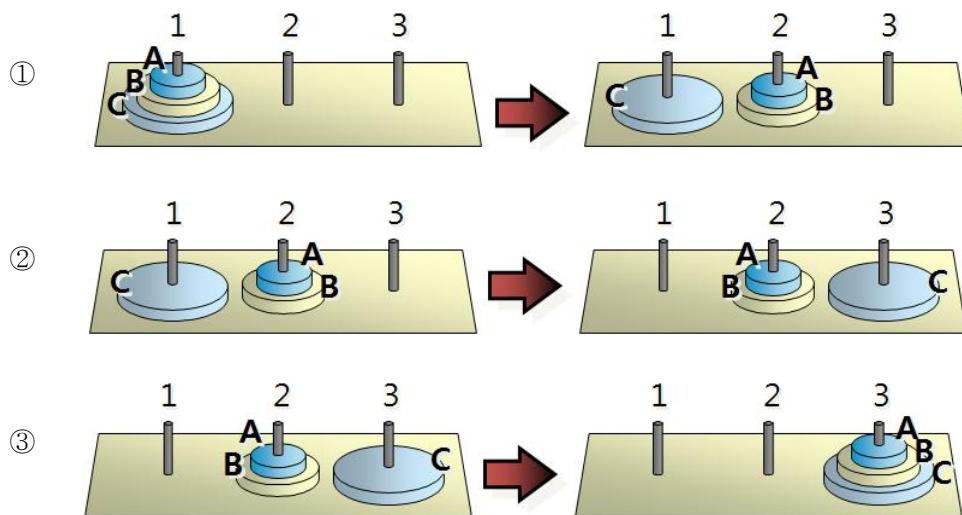
### 6. 문제축소에 의한 문제풀이

● 문제축소(problem reduction)

- 주어진 문제를 분석하여 보다 단순화된 부분문제로 분해함
- 문제를 분해하는 과정을 반복하여 아주 쉽게 풀이할 수 있는 작은 문제로 분해되면 원래의 문제를 해결할 수 있음

● 예 : 하노이 탑 문제

- 세 개의 부분문제로 축소할 수 있음



- ②의 문제는 단 한 번 원판이동을 함으로써 풀이할 수 있는 문제임 : 원시문제
- ①, ③의 부분문제는 또다시 더 작은 문제로 분해될 수 있음

## 제3장 탐색에 의한 문제풀이

### ▣ 주요 학습내용

- 그래프 탐색의 개요
- 맹목적 탐색방법 : 깊이우선 탐색 및 너비우선 탐색, 균일비용 탐색
- 경험적 탐색방법 : 언덕오르기, 모의 담금질, 최적우선 탐색, A\* 알고리즘 등
- 게임트리의 탐색

### ▣ 학습지침

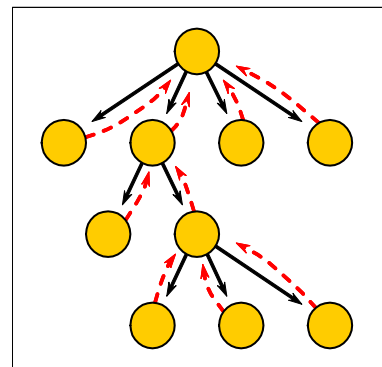
전반적인 탐색 알고리즘의 골격과 함께 OPEN 리스트에서 어떤 순서로 노드를 선택하여 확장하는가에 주의하여 학습한다. 균일비용 탐색에서는 비용을 어떻게 정의하는가, 탐색과정에서 노드의 비용은 어떻게 계산하는가를 잘 이해한다. 경험적 탐색에서는 경험적 지식을 반영하여 정의된 평가함수가 탐색에 활용되는 점에 유의하며, 각각의 탐색방법에서 어떤 탐색결과를 얻을 수 있는지도 알고리즘별로 비교하며 학습한다.

### 1. 그래프 탐색 과정

- ※ 상태공간에서 목표상태의 탐색 : 초기상태에서부터 시작하여 연산자를 이용하여 새로운 차기상태를 생성하고, 그중 적절한 대상을 선택하여 다시 차기상태를 생성하는 과정을 반복
  - ⇨ 상태 간의 의존관계가 형성
  - ⇨ 그래프로 표현

#### ● 탐색과정

- ① 주어진 상태에 적용할 수 있는 모든 연산자를 가하여 모든 차기상태를 생성
  - ⇨ 후계노드의 집합 생성
  - ※ 확장 : 어떠한 상태에 적용 가능한 모든 연산자를 적용하여 모든 후계상태를 생성하는 것
- ② 후계 노드에는 부모노드를 가리키는 포인터를 첨부 : 탐색경로를 알 수 있도록 함.
- ③ 정해진 기준에 따라 다음 노드를 선택
  - ※ 앞의 기본적 처리시스템에 탐색순서를 결정하기 위한 몇 가지 처리시스템이 포함됨



#### ● 탐색과정 중 사용하게 될 자료구조

- OPEN : 앞으로 확장해야 할 노드를 저장하는 리스트
- CLOSED : 이미 확장된 노드를 저장하는 리스트

#### ● 탐색에 사용되는 정보에 따른 분류

- 맹목적 탐색 : 목표노드의 위치에 관계없이 단순히 미리 정한 순서에 따라 탐색하는 방법

- 경험적 탐색 : 목표노드의 위치에 대한 경험적 정보를 사용하여 가장 타당한 노드를 다음 확장대상으로 선정
  - ※ 경험적 지식 : 많은 시행착오를 거쳐 경험적으로 얻은 지식으로, 항상 맞아떨어지는 완벽한 것은 아니지만, 많은 경우 유용하게 사용될 수 있음

● 탐색의 목표에 따른 분류

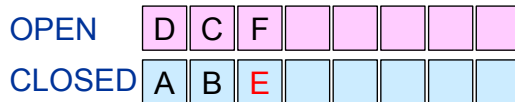
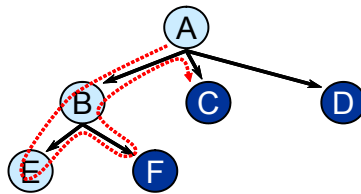
- ① 어떤 경로를 거치는 목표상태에 도달하는 경로를 찾기만 하면 되는 문제
- ② 초기상태로부터 목표상태에 도달하는 최적의 경로를 탐색하는 문제

정보 사용 \ 목적	임의경로 탐색	최적경로 탐색
맹목적 탐색	깊이우선 탐색 너비우선 탐색	균일비용 탐색
경험적 탐색	언덕오르기 탐색 최적우선 탐색	A* 알고리즘

**2. 맹목적 탐색방법 : 깊이우선 탐색 및 너비우선 탐색**

● 깊이우선 탐색

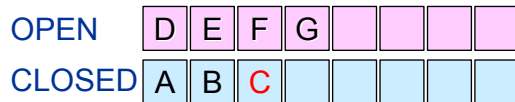
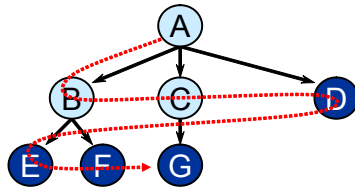
(예)



- OPEN, CLOSED : 초기에 CLOSED는 비어 있고, OPEN에는 출발노드를 넣은 상태에서 시작
- 노드를 확장한 결과 생성된 후계노드는 OPEN의 끝에 추가
- 후계노드 생성 후 : OPEN에 있는 노드 중 깊이방향의 노드를 선택
  - ⇒ OPEN의 끝에서 1개의 노드를 꺼냄
    - 꺼내는 노드는 마지막으로 OPEN에 넣은 노드임
  - ⇒ OPEN은 스택구조
- 계속해서 깊이방향으로 탐색해 들어갈 경우, 만약 이 경로에 해가 존재하지 않으면 탐색이 성공할 수 없음
  - ⇒ **깊이제한** 설정
  - ⇒ 깊이제한에 도달하면 탐색하지 않은 최근의 후계노드가 존재하는 상태로 복귀하여 탐색을 계속
  - ※ 백트래킹 : 다른 방향으로의 탐색이 가능한 최근의 상태로 복귀하는 것

● 너비우선 탐색

(예)



- OPEN, CLOSED : 초기에는 CLOSED는 비어 있고, OPEN에는 출발노드를 넣은 상태에서 시작
- 노드를 확장한 결과 생성된 후계노드는 OPEN의 끝에 추가
- 다음 확장할 노드의 선택 : 같은 레벨의 노드를 먼저 선택
  - ⇒ OPEN의 앞에서 1개의 노드를 꺼냄
    - 꺼내는 노드는 가장 먼저 OPEN에 넣은 노드임
  - ⇒ OPEN은 큐 구조
  - ⇒ 해가 발견된다면 그 해는 출발노드로부터 가장 가까운 경로임.

※ REVIEW

● 깊이우선 탐색

- 너비우선 탐색에 비해 적은 메모리 공간을 소비
- 유연히 아주 빠르게 풀이경로를 찾을 수 있다.

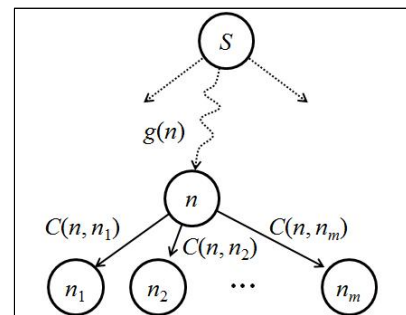
● 너비우선 탐색

- 해가 없는 경로를 불필요하게 깊이 탐색할 필요가 없다.
- 해가 존재한다면 반드시 찾을 수 있고, 그 풀이경로는 최단경로이다.

3. 균일비용 탐색 (uniform-cost search)

● 경로비용의 계산

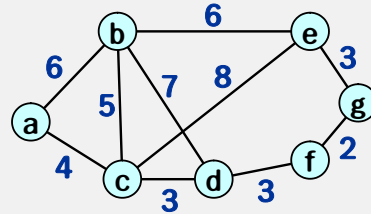
- $g(n)$  : 출발노드  $S$ 로부터 노드  $n$ 까지의 최소 경로비용
- $C(n, n_i)$  : 노드  $n$ 과 후계노드  $n_i$  사이의 경로비용
- 후계노드  $n_i$ 의 비용은 노드  $n$ 의 최소 경로비용에 노드  $n$ 과 후계노드  $n_i$  사이의 경로비용을 합한 것
  - ⇒  $g(n_i) = g(n) + C(n, n_i)$
- ※ 최소비용 경로의 탐색 :  $g(n)$ 이 최소인 노드  $n$ 을 우선 선택



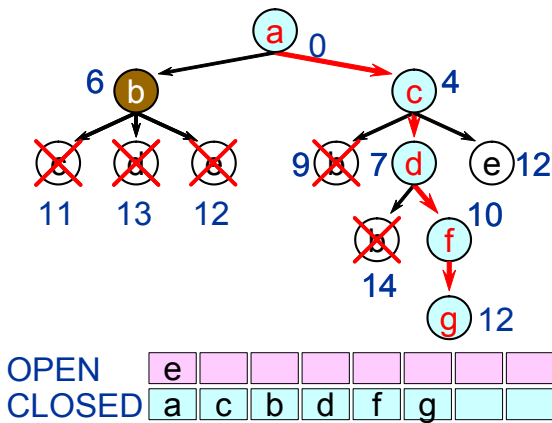
● 문제풀이 과정의 예

**최단경로 탐색문제**

a~g의 7개의 도시가 있다. 도시 사이에는 도로가 건설되어 있고, 각 도로의 거리는 그림과 같다. 도시 a에서 g까지 가는 최단 경로를 찾아라.



- 상태 : 현재 위치한 도시
- 연산자 : 각 도시로 이동하라는 지시
- 탐색과정



**※ REVIEW - 균일비용 탐색**

- 경로비용이 최소인 노드를 선택하여 확장하므로 선택된 노드가 목표노드이면 그 경로는 최소비용 경로
- 8-퍼즐 문제에서는 조각의 이동횟수가 비용이며, 따라서 모든 연산자의 적용비용이 동일하므로 균일비용 탐색은 너비우선 탐색과 동일한 탐색을 하게 된다.

**4. 경험적 탐색방법**

※ 경험적 지식 : 해에 대한 완벽한 정보는 아니지만, 많은 경우 유용하게 사용할 수 있는 경험적으로 얻은 지식

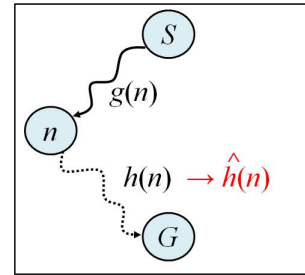
● 경험적 탐색

- 목표상태를 보다 신속하게 탐색하기 위해 경험적 지식을 활용하는 탐색방법
- 평가함수 사용
  - 어떤 상태가 주어졌을 때 그 상태를 거쳐 가는 것이 목표상태로 가는 데 얼마나 바람직한가를 나타내는 함수
  - 해를 향해 가는 데 필요한 비용, 해로 향하는 경로상에 존재할 가능성 등

● 평가함수

- $g(n)$  : 출발노드 S로부터 현재 상태를 나타내는 노드 n까지의 경로비용

- $h(n)$  : 노드  $n$ 으로부터 목표노드  $G$ 까지의 경로비용
  - 아직 탐색하지 않은 경로이므로 정확한  $h(n)$  값을 알 수 없음
  - ⇒ 예측비용  $\hat{h}(n)$  사용



- ※ 탐색 알고리즘에 따라 사용하는 요소에 차이가 있다.
  - ▷ 언덕오르기 탐색 :  $\hat{h}(n)$  만 사용
  - ▷ A\* 알고리즘 :  $g(n) + \hat{h}(n)$ 을 사용

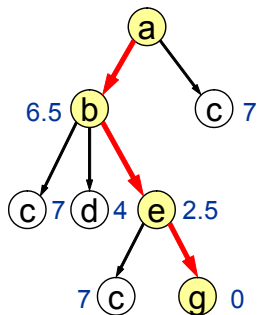
**5. 언덕오르기 탐색**

- 언덕오르기 탐색 알고리즘
  - 깊이우선 탐색과 유사한 순서로 탐색
  - 후계노드의 평가함수를 계산하여, 가장 비용이 적은 노드를 다음 확장할 노드로 선택
  - 평가함수는 후계노드에서부터 목표노드까지의 예측 경로비용을 사용하며, 후계노드까지 도달하는 데 사용된 비용은 고려하지 않음

- 경로탐색 문제의 예
  - 도시 a에서부터 도시 g까지 가는 경로를 탐색하는 문제
  - 평가함수 : 목표도시까지의 직선거리

도시 간 거리	목표도시까지의 직선거리

▷ 탐색과정

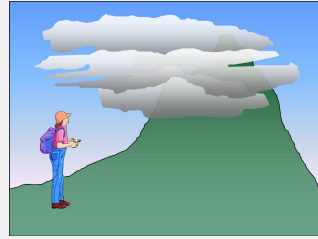


※ 탐색결과가 최소비용 경로는 아님에 주의

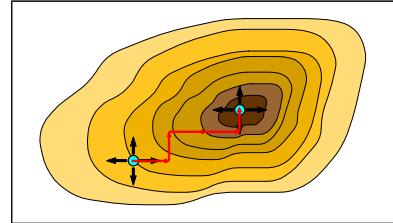
- 계수최적화 문제 : 최적의 결과를 낼 수 있도록 시스템의 계수를 조정하는 문제

**등산가 문제**

어떤 사람이 초행길의 산을 등산하는 도중 짙은 안개를 만났다. 지도도 없고, 사람이 다니는 길도 없다. 다만, 나침반을 가지고 있으며, 산의 정상에 오르려고 한다. 어떻게 정상에 도달할 수 있겠는가?



- 상태 : 좌표 및 고도
- 연산자 : 동서남북 방향으로 정해진 거리만큼 이동
- 탐색과정 : 안개 때문에 일정 거리 이상의 높이는 측정할 수 없으므로 정해진 거리 이내로만 이동하는 것으로 제한
- 목표상태 : 어느 방향으로 이동해도 고도가 낮아지는 지점



※ 계수최적화 문제에서 발생할 수 있는 문제점

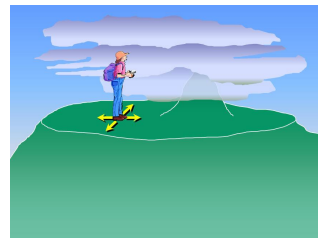
1. 지역최대치 문제

- 정상 주변에 정상보다 낮은 봉우리들이 있을 때, 그 주변 봉우리에 오를 경우 그곳이 정상인 것으로 판단할 수 있음
- ⇒ 대처방안 : 백트래킹을 통해 다른 가능한 경로를 탐색



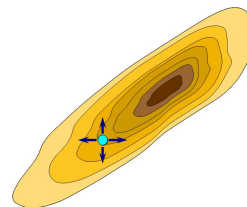
2. 고원문제

- 산중턱에 존재하는 평평한 영역에 도달했을 경우 어느 방향으로 이동해도 현재 상황을 개선할 수 없음
- ⇒ 대처방안 : 동일 연산자를 반복 적용한 결과를 토대로 진행방향을 결정



3. 능선문제

- 날카로운 능선상에 위치해 있을 때 정해진 동서남북 어느 방향으로 이동해도 고도가 낮아져서 정상으로 오관하게 되는 문제(이동방향의 해상도와 관련된 문제)
- ⇒ 대처방안 : 검사방향을 늘리거나 연산자들을 조합하여 적용한 결과를 사용하는 방안



## 6. 모의 담금질

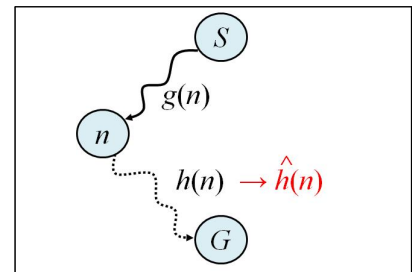
- 모의 담금질 : 방대한 탐색공간에서 주어진 평가함수의 전역최대치를 구하기 위한 확률적인 경험적 접근방법
- 알고리즘 개요
  - 처리과정은 전반적으로 언덕오르기와 흡사하나 후계노드 중 평가함수가 최대인 노드를 선택하는 대신에 임의의 후계노드를 선택
  - 선택된 후계노드가 개선된 상태를 나타낸다면 무조건 그 후계노드를 차기상태로 선택
  - 선택된 후계노드가 상태를 개선하지 못한다면 정해진 확률에 따라 그 후계노드를 선택함. 이 확률은 후계노드의 바람직한 정도에 따라 정해지며, 시간에 따라 감소하는 온도에 따라 감소함
- 장점 및 단점
  - 장점 : 폭넓은 응용 가능성과 최상에 가까운 해답을 얻을 수 있다는 점
  - 단점 : 좋은 해답을 얻는 데 까지 많은 시간이 소비된다는 점

## 7. A\* 알고리즘

※ 최적경로를 탐색하기 위한 방법

### ● 평가함수

- $\hat{f}(n) = g(n) + \hat{h}(n)$ 
  - $g(n)$  : 출발노드로부터 노드  $n$ 까지의 경로비용
  - $\hat{h}(n)$  : 노드  $n$ 으로부터 목표노드까지의 예측 경로비용



### ● A\* 탐색 알고리즘

- 기본골격은 타 알고리즘과 동일
- 특징적 내용
  - 비용 : 목표노드로부터 현재노드를 거쳐 목표노드로 가는 경로의 예측비용
  - OPEN이나 CLOSED에 동일한 노드가 존재할 경우의 처리방법
- 알고리즘

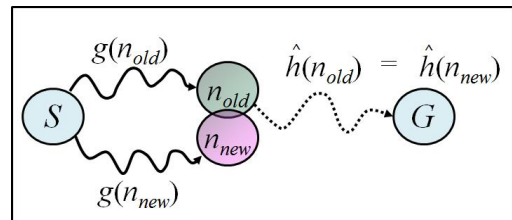


1. 출발노드를 OPEN에 삽입 ( $\hat{f}$ 을 계산하여 첨부)
2. OPEN에 남은 노드가 있으면 다음을 반복
  - 1) OPEN에서  $\hat{f}$ 이 최소인 노드를 꺼내 CLOSED에 넣는다(노드  $n$ ).
  - 2) 노드  $n$ 이 목표노드라면 탐색성공
  - 3) 노드  $n$ 을 확장하여 후계노드  $n_1, n_2, \dots, n_i$ 를 생성
  - 4) 후계노드의 평가함수  $\hat{f}(n_1), \hat{f}(n_2), \dots, \hat{f}(n_i)$ 를 계산
  - 5) 후계노드  $n_k, k=1, \dots, i$ 를 OPEN에 삽입(중복된 노드 제거)
3. 탐색 실패

□ 중복된 노드의 처리 (동일한 상태(노드)가 OPEN에 존재하는 경우)

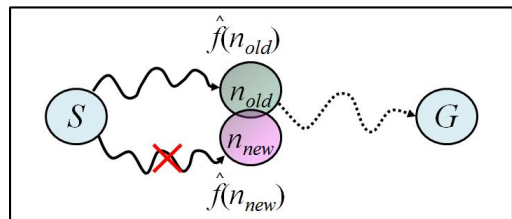
※  $n_{old}$ 를 기존 노드,  $n_{new}$ 를 새로 생성된 노드라고 가정

- 아직 어느 노드도 확장되지 않은 상태이므로 평가함수  $\hat{f}$ 이 큰 노드를 제거하면 됨
- $\hat{h}(n_{old})$ 와  $\hat{h}(n_{new})$ 는 동일
- $g$  값이 큰 노드를 제거



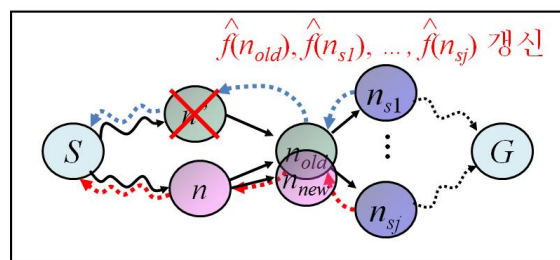
□ 중복된 노드의 처리 (동일한 상태(노드)가 CLOSED에 존재하는 경우)

- ①  $\hat{f}(n_{old}) \leq \hat{f}(n_{new})$  인 경우
  - $n_{new}$ 를 제거하면 됨



②  $\hat{f}(n_{old}) > \hat{f}(n_{new})$ 인 경우

- $n_{old}$ 를 제거해야 하는데, 이미 확장한 노드이기 때문에 후계노드들이 존재하므로 주의가 필요
- $n_{old}$ 는 자신의 부모노드를 가리키도록 포인터가 첨부되어 있고,  $n_{old}$ 의 후계노드들이 OPEN이나 CLOSED에 들어 있음



- $n_{old}$ 를 제거하는 대신  $n_{new}$ 를 제거
- $n_{old}$ 가 가리키던 부모노드 포인터를  $n_{new}$ 의 부모노드를 가리키도록 함
- $n_{old}$  및  $n_{old}$ 의 후계노드들의 평가함수값을 재조정 ( $g(n_{old})$ 가 바뀌었음)

● 최소비용 경로탐색을 위한 조건

만일 어떤 노드로부터 목표노드까지의 경로비용을 예측한 값이 항상 실제 비용보다 작다면, 즉  $\hat{h}(n) \leq h(n)$ 이 항상 만족된다면 A\* 알고리즘은 최소비용 경로를 탐색하는 것을 보장한다.

**증명**

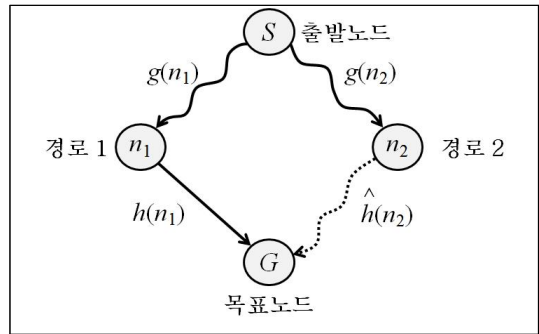
탐색과정 중 노드  $n_1$ 을 거쳐 목표노드  $G$ 가 생성되었고, OPEN의 노드 중에서 이 목표노드  $G$ 의 평가함수값이 최소여서 선택되었다고 가정

- OPEN에 있는 임의의 노드  $n_2$ 의 평가함수는  $G$ 의 평가함수보다 큼

$$g(n_1) + h(n_1) \leq g(n_2) + \hat{h}(n_2)$$

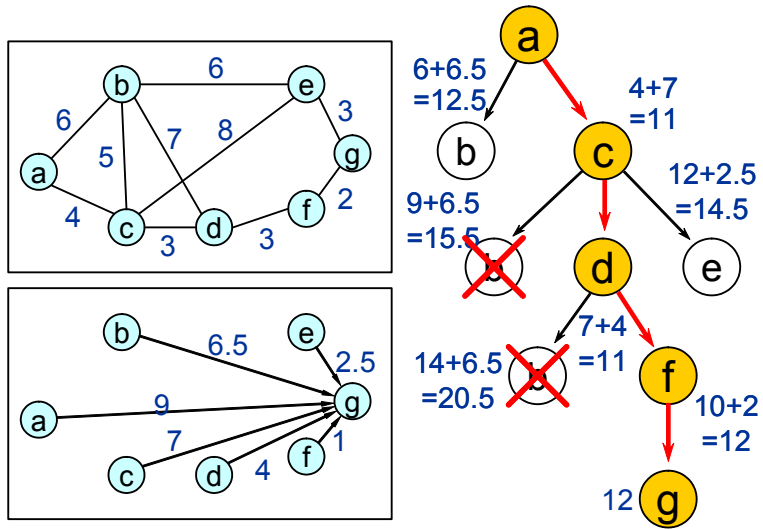
- 앞의 조건에 따라 노드  $n_2$ 로부터 목표노드까지의 실제 비용인  $h(n_2)$ 는 경로비용 예측치인  $\hat{h}(n_2)$ 보다 크거나 같음

$$\begin{aligned} g(n_1) + h(n_1) &\leq g(n_2) + \hat{h}(n_2) \\ &\leq g(n_2) + h(n_2) \end{aligned}$$



- ⇒ OPEN에 있는 어느 노드도  $n_1$ 을 거쳐  $G$ 에 도달한 경로에 비해 비용이 적을 수 없음
- ⇒ 최소비용 경로탐색을 보장

(예) 최단경로 탐색문제



- 직선거리는 실제 거리보다 길지 않으므로 항상  $\hat{h}(n) \leq h(n)$ 임
- ⇒ 최단거리 경로를 탐색하는 것을 보장함

**8. 게임트리와 최대최소 탐색**

- 상대가 있는 게임 : 장기·바둑 등
- 나와 상대방이 번갈아 수를 둠

- 현재의 판의 상태에서 내가 둘 수 있는 수와 상대방이 둘 수 있는 수가 교대로 나타나는 트리가 구성

→ 최대최소 탐색트리

● 최대최소 탐색트리

- 루트노드는 현재의 판의 상태이며, 이후 내가 둘 수 있는 수와 그 수에 대해 상대방이 둘 수 있는 수가 트리의 레벨에 따라 교대로 나타남
- 이러한 수의 예측은 시스템의 자원이나 시간적 제약이 허용하는 수준까지 이루어질 수 있으며, 이후는 각각의 판의 형태가 얼마나 내게 유리한가를 평가함수에 의해 예측
- 나는 내가 둘 수 있는 여러 수들 중에서 내게 가장 유리한 수를 두는 것이 좋으므로, 가장 큰 평가값을 선택 : 최대화
- 상대방은 내게 가장 불리한 수를 두려고 할 것이라는 가정하에 가장 작은 평가값을 선택 : 최소화
- 최대최소 탐색트리의 예 : 교재 <그림 3-16> 참조

● 예 : 삼목게임

- 두 사람이 가로세로 3×3 크기의 판에 수를 두어 한 행, 열, 또는 대각선을 모두 점유하면 이기는 게임

○		×
×	×	×
○		○

- 평가함수  $F$ 
  - 승리한 상태 :  $F = \infty$
  - 패배한 상태 :  $F = -\infty$
  - 그 외의 경우 :  $F = W - L$ 
    - $W$  : 이길 가능성이 있는 행, 열, 대각선의 수
    - $L$  : 질 가능성이 있는 행, 열, 대각선의 수

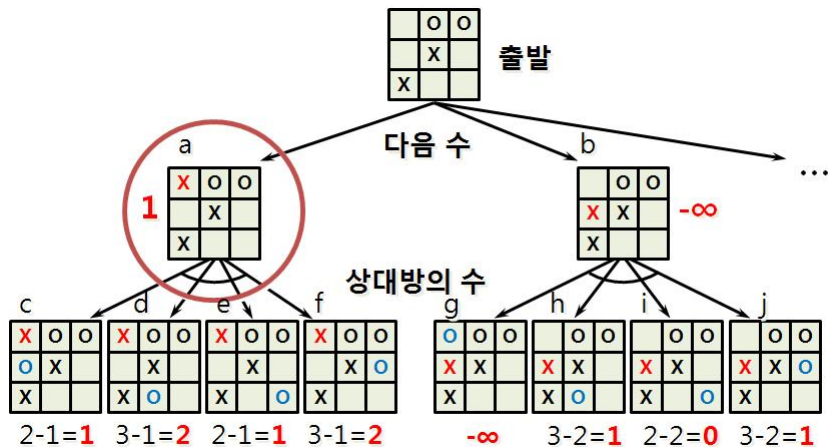
×	○	○
○	×	
×		

→  $F = 2 - 1 = 1$

×	○	○
	×	
×	○	

→  $F = 3 - 1 = 2$

- 다음 수의 결정



●  $\alpha - \beta$  가지치기

- 최대최소 탐색트리의 불필요한 가지를 잘라냄으로써 탐색의 성능을 높이기 위한 알고리즘
- $\alpha$  : 최대화 노드가 갖게 될 최솟값
- $\beta$  : 최소화 노드가 갖게 될 최댓값
- 노드  $i$ 가 노드  $j$ 의 부모노드라면 노드의 가지치기는 다음의 규칙을 따름

- ① 노드  $j$ 가 최소화 노드인 경우  $\alpha_i \geq \beta_j$ 인 것으로 확인되면 노드  $j$ 는 가지치기 됨
- ② 노드  $j$ 가 최대화 노드인 경우  $\beta_i \leq \alpha_j$ 인 것으로 확인되면 노드  $j$ 는 가지치기 됨

## 제4장 지식과 인공지능

### ▶ 주요 학습내용

- 지식기반 시스템의 개념
- 지식표현 방법이 갖추어야 할 요건
- 지식표현 방법 : 논리, 규칙, 시맨틱 네트, 프레임 등
- 전문가 시스템

### ▶ 학습지침

지식의 사용은 인공지능 시스템에서 핵심적인 사항이다. 제4장에서는 지식표현의 전반에 대해 이해를 한다. 사람의 두뇌에 형성된 지식을 어떻게 정형화하여 표현하려 하는가의 관점에서 여러 가지 표현방법을 학습한다.

### 1. 지식기반 시스템

#### ● 지식기반 시스템(knowledge based system, KBS)

- 지능적 문제의 풀이는 궁극적으로 그 문제와 관련된 지식들을 어떻게 저장하고 이용할 것인가에 달려 있음 : 지식공학(knowledge engineering)
- 데이터 처리 시스템에서 데이터베이스를 사용하는 것처럼 지식 베이스를 사용
- 지식 베이스 : 특정한 문제 분야의 지식을 쉽게 접근할 수 있는 형태로 컴퓨터 내에 체계적으로 축적한 것
- 지식기반 시스템 : 방대한 양의 특정 분야 지식을 축적하고 있는 지식 베이스를 이용하여 문제를 풀이하거나 판단작업을 자동화하는 시스템

#### ● 지식기반 시스템의 구성

- 지식 베이스 : 적용하고자 하는 분야의 지식을 포함한 모든 문제풀이에 필요한 지식을 저장
  - 지식을 어떻게 이용할 것인가에 대한 내용은 포함되어 있지 않음
- 추론기관 : 주어진 문제의 상황에 대하여 지식 베이스 내의 지식을 이용해 추론함으로써 결론을 제시하는 역할을 함
- 사용자 인터페이스 : 사용자와 KBS의 연결을 담당
  - 사용자가 지식 베이스에 저장된 지식에 접근하고 수정하는 데 도움을 주는 역할
  - 추론기관의 추론과정에 대한 설명 제공

### 2. 지식의 표현과 그 구조

- 지식의 표현 : 실세계의 지식을 컴퓨터 내부의 표현방법으로, 또는 그 역으로 변환하는 것
  - 정방향 사상 : 실세계의 사실을 내부 지식표현으로 사상하는 것

- 역방향 사상 : 내부 지식표현을 실세계의 표현으로 사상하는 것
- ※ 내부 표현방법이 실세계의 사실을 얼마나 근사하게 묘사하는가, 타당한 추론결과를 이끌어 내는 능력이 얼마나 우수한가에 따라 성능이 좌우됨

(예) 실세계의 지식과 내부 지식표현

실세계의 지식	내부 지식표현
철수는 사람이다.	$man(\text{철수})$
모든 사람은 생각한다.	$\forall x (man(x) \rightarrow think(x))$
철수는 생각한다.	$think(\text{철수})$

- 실세계의 사실과 내부 지식표현 사이의 정방향, 역방향 사상이 올바르게 이루어지고, 실세계의 추론과정이 내부 추론과정으로 바르게 모델링되고 있는가가 중요

● 지식의 분류

- ① 문제에 관한 지식 : 해결하고자 하는 문제영역에서 문제와 관련된 사실로부터 얻어지는 지식
  - 구체적·단편적 지식들
- ② 대상세계에서 성립하는 규칙과 경험법칙 : 문제해결 과정에서 지식 베이스의 지식을 사용하여 탐색하게 되는데, 효율적 탐색을 위해 대상세계에서 성립하는 규칙과 경험법칙이 사용될 수 있음
- ③ 메타지식(meta knowledge) : 지식의 사용에 관한 지식
  - 어떤 규칙이 다른 규칙보다 우선적으로 적용해야 할 조건을 제시
  - 특수한 문제풀이에 유용한 지식의 적용순서를 제시
  - 추론에 대한 설명을 통해 결과에 대한 신뢰감 제공

● 지식표현 방법이 갖추어야 할 요건

- 표현방법의 적합성 : 표현하고자 하는 지식의 실세계의 의미를 최대한 수용할 수 있는 능력
- 추론의 적합성 : 표현된 지식을 이용하여 추론을 할 수 있는 메커니즘이 존재해야 함
- 추론의 효율성 : 추론과정이 효율적으로 진행될 수 있어야 함
- 지식획득 능력 : 새로운 지식을 쉽게 습득할 수 있는 능력
  - 수동적 추가 ↔ 자동적인 학습능력

● 지식의 형태

1. 절차적 지식

- ▷ 어떠한 경우에 무엇을 어떻게 할 것인가에 대한 지식
- ▷ LISP, PROLOG 등과 같은 프로그래밍 언어로 작성된 명령어의 집합 : 지식의 사용은 프로그램 실행하는 것
  - ↳ 지식 사용에 대한 제어정보는 지식 자체에 내포되어 있음
- ▷ 추론의 적합성 면이나 지식획득의 효율성 면에서 낮은 평가

2. 선언적 지식

- ▷ 상호 독립적이고 단편적인 지식을 나열해 놓은 형태 : 정적인 지식
- ▷ 추론기관이라는 프로그램이 별도로 존재하며, 이 프로그램에 의해 지식이 추론에 사용됨.
- ▷ 각각의 지식마다 독립성이 강하므로 개별적으로 지식을 편집, 획득, 검색하는 것이 절차적 지식에 비해 용이함

**3. 논리를 이용한 지식표현**

● 술어논리(predicate logic)

- 명제 : 참·거짓이 명백한 문장
  - 어떠한 객체와 그 객체를 수식하는 술어로 구성 : 술어와 객체를 분리하여 표현
- 논리연산자 : AND, OR, NOT, 조건명제
- 기본적인 추론방식 : 연역법칙 사용
  - X와 X→Y가 참이면 Y가 참이라는 사실을 추론할 수 있다.

(예) 철수는 사람이다.            ⇨ *man(철수)*  
           ↓                    ↓  
       객체                    술어

모든 사람은 생각한다.    ⇨  $\forall x (man(x) \rightarrow think(x))$

**4. 규칙을 이용한 지식표현**

● 규칙

- 주어진 상황을 위한 권고·지시·전략을 나타내는 정형화된 표현방법
- ‘IF ① THEN ②’ 형태의 표현
  - ① : 가정, 전제조건, LHS
  - ② : 결론, 실행할 동작, RHS
- 지식 베이스 : 현재의 상황을 나타내는 사실(또는 데이터 항목)과 이에 대해 적용할 규칙으로 구성
- 추론기관이 현재상태에 의해 만족되는 규칙을 선택하여 실행

● 전방향 추론

- 주어진 사실로부터 만족되는 규칙을 규칙의 조건부와 정합에 의해 선택
- 선택된 규칙의 결론부의 내용을 실행하거나 사실에 추가

● 후방향 추론

- 목표로 하는 결론이 현재상태 또는 알려진 사실로부터 유도해 낼 수 있는가를 알아내고자 함
- 결론부로부터 가정부 방향으로 진행되는 추론

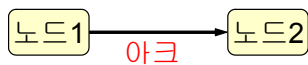
**5. 시맨틱 네트를 이용한 지식표현**

※ 규칙을 이용한 지식표현 방법은 간단하고 이해하기 쉬우나 다음과 같은 문제점이 있음

- ① 구조화되어 있지 않음
- ② 규칙의 개수가 증가하면 처리나 수정이 어려움
  - 규칙 상호 간 종속관계, 규칙 사이의 상충 등이 발생

● 시맨틱(어의에 관한, 의미론적인) 네트

□ 네트워크를 기초로 한 지식표현 방법

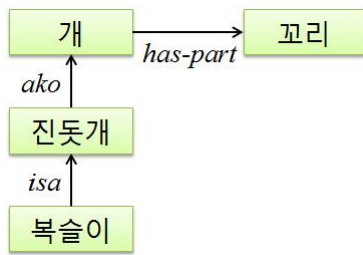


- 노드 : 객체, 개념, 사건 등을 표현
- 아크 : 노드 사이의 관계를 표현

● 아크 속성의 대표적인 예

- *ako*(a kind of) 아크 : 상위 개념의 하부 클래스
    - 즉, 부분집합을 나타내는 속성
  - *isa*(is a) 아크 : 어떤 클래스에 해당되는 개별적인 하나의 사례(instance)
    - 즉, 원소관계
  - *has-part* : 어떤 객체의 부속품을 나타내는 속성
- ※ 책에 따라 *ako* 아크를 *isa* 아크로, *isa* 아크를 *instance* 아크로 표기하기도 함

(예)



- 개 : 여러 종류의 개를 통틀어 칭하는 포괄적인 개념
- 진돗개 : 여러 종류의 개 중에서 한 부류에 속하는 것들 - 보다 포괄적인 개념인 '개'의 부분집합
- 복슬이 : 진돗개 중 특별한 한 마리의 개를 칭하는 개념 - 진돗개 집합의 한 원소

● 특성상속(property inheritance)

- 상위 클래스의 속성과 값을 하위 클래스 또는 사례가 이어받도록 하는 추론형태
- 상위 개념의 지식을 하위 개념이 공유하는 중앙집중적 방법의 지식표현
- 특성상속이 이루어지는 아크 : *isa* 아크와 *ako* 아크

※ 상위 개념은 일반적인 속성을, 하위 개념은 일반적 속성과 다르거나 고유한 속성을 연결

● 특성상속 알고리즘 : 교재 105쪽



● 특성상속의 장점

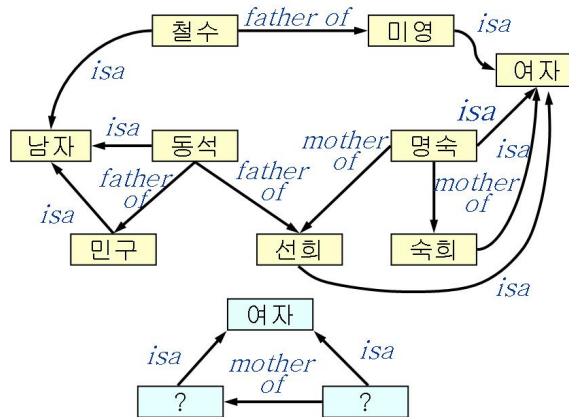
- 지식을 구성하기 쉬움 : 동일한 유형에 속하는 여러 개의 객체에 공통적인 속성을 개별적으로 저장할 필요가 없음
- 표현된 지식의 오류를 쉽게 수정할 수 있음
- 최신의 지식을 유지하기 쉬움
- 지식의 분배가 자동적으로 이루어짐

● 시맨틱 네트에서의 지식 검색

1. 활성영역 전개

- 두 노드 사이의 연결관계를 파악하기 위한 추론방법음는다.

2. 정합



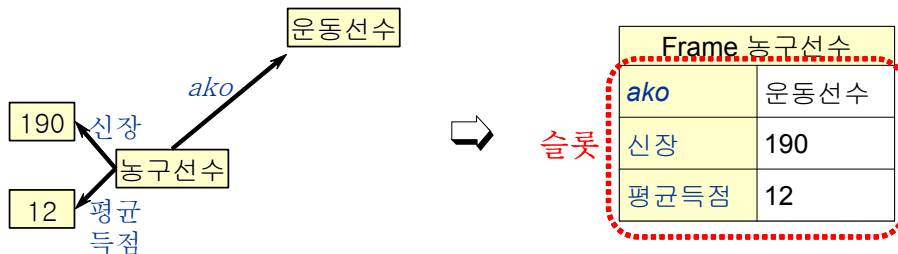
- 어떠한 부분 네트워크가 주어졌을 때, 이 네트워크의 변수 노드에 대한 값을 결정하기 위해 사용

6. 프레임을 이용한 지식표현

※ 시맨틱 네트 : 객체를 표현하는 노드들을 속성이 부착된 아크로 연결한 표현형태

⇒ 표현해야 할 지식이 다양하고 방대해짐에 따라 보다 구조적인 표현방법이 필요

⇒ 프레임 구조



● 프레임

- 어떠한 객체를 표현하기 위한 속성 및 속성값의 집합

- 관련된 프레임들이 상위 개념, 하위 개념으로 분류되어 연결
- 클래스 프레임 : 유사한 성격을 갖는 객체의 집합을 표현하는 프레임
- 부클래스 프레임 : 어떠한 클래스에 속하는 일부 객체를 표현하는 프레임
- 인스턴스(instance) 프레임 : 어떤 클래스에 속한 특정객체를 표현하는 프레임
- 특성상속

(예)

Frame 인간	
<i>ako</i>	동물
이동	직립보행
지능	(내정값=100)

Frame 성인남자	
<i>ako</i>	인간
연령	
키	(내정값=170)
체중	(내정값=65)
결혼관계	
배우자	

Frame 홍길동	
<i>instance</i>	성인남자
연령	35
키	175
체중	70
결혼관계	기혼
배우자	이영숙

● 부가 프로시저

- 어떠한 슬롯에 값을 넣거나 읽거나 지우는 등 해당 슬롯을 사용했을 때, 그 사용과 관련하여 수행해야 할 동작을 지시
- 평소에는 동작하지 않고 있다가, 그 슬롯에 지정된 동작이 행해지면 자동적으로 실행
- 부가 프로시저의 종류 : 필요 프로시저, 판독 프로시저, 기록 프로시저, 제거 프로시저 등

※ 부가 프로시저를 첨부함으로써 절차적 지식을 포함시킬 수 있음

※ 프레임의 특성

- 각각의 객체의 고유한 구조 및 행동을 캡슐화시킬 수 있음
- 상속을 통해 일반적 개념을 수용할 수 있음
- 복잡한 영역에서도 지식의 모듈 특성을 유지할 수 있고, 문제영역의 깊이 있는 지식을 표현할 수 있음

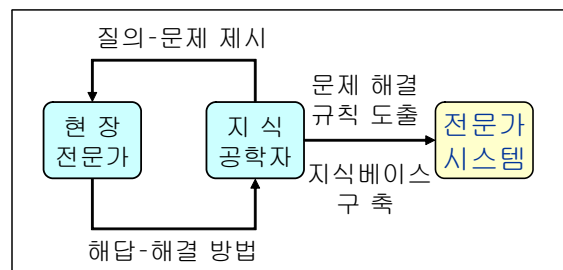
**7. 전문가 시스템**

● 전문가 시스템의 개요

- 한정된 문제 분야에 적용하기 위해 그 분야의 전문지식을 축적한 지식 베이스를 구축하여 만든 지식기반 시스템
- ※ 지식기반 시스템 : 문제영역 지식이 그 외의 다른 지식과 분리되어 있는 시스템
  - 지식 베이스 + 추론기관

● 전문가 시스템의 구축

- 현장전문가(domain expert) : 문제 분야의 전문지식을 가지고 있는 전문가
- 지식공학자(knowledge engineer) : 현장전문가가 가지고 있는 전문지식을 체계적으로 지식 베이스화하는 역할을 담당



● 전문가 시스템의 장점

현장전문가	전문가 시스템
한시적	영구적 ↳ 인간과는 달리 영구적으로 사용할 수 있음
지식의 전달이 어려움	지식의 전달이 쉬움 ↳ 한 사람의 전문가를 양성하려면 교육을 위해 많은 시간이 소비되나, 전문가 시스템은 프로그램이므로 쉽게 많은 복사본을 만들 수 있음
문서화하기 어려움	지식을 문서화하기 쉬움
일관적이지 않음.	문제풀이 과정이 일관적임 ↳ 신체적 상황 등에 따라 다른 결과를 내지 않음
많은 비용 소비	저렴함

● 전문가 시스템의 단점

현장전문가	전문가 시스템
창조적	영감이 없음 ↳ 창조적 능력이 부족함
적응적	새로운 상황에 적응하기 어려움 ↳ 사람은 새로운 상황이 발생했을 때, 이에 대처함은 물론 그 경험을 바탕으로 새로운 지식을 축적할 수 있는 능력이 있음
복잡하고 다양한 입력형태 사용	정해진 기호로 입력되어야 함 ↳ 사람은 다양한 형태의 입력을 받아들일 수 있으나, 전문가 시스템은 이를 기호화된 형태로 변환하여 입력해야 함
거시적 판단기준	미시적 판단 위주 ↳ 거시적 판단능력의 부족함
상식적 지식 포함	전문적 지식만 포함 ↳ 상식적 지식의 활용능력이 부족함

**8. 전문가 시스템의 구조**

● 지식 베이스

- 주로 규칙이나 프레임을 이용하여 지식표현
- 사실 및 규칙을 저장
- 완전하지 않거나 확실적인 정보도 포함할 수 있음
  - 경험적 지식

● 추론기관

- 지식 베이스를 이용하여 문제를 풀이기 위한 일반적인 제어지식을 포함하고 있는 부분

□ 추론기관의 구성

- 규칙해석기 : 지식 베이스의 지식을 어떻게 적용할 것인가를 결정하는 부분
- 스케줄러 : 지식을 적용하여 문제를 풀이하는 순서를 결정하는 부분

※ 문제 분야의 전문지식과 일반적인 문제풀이 지식이 분리되어 있어 지식의 추가, 수정이 용이함

● 전문가 시스템의 개발도구

□ 일반적인 프로그래밍 언어

- 범용 프로그래밍 언어 : PASCAL, C 등 일반적으로 사용하는 프로그래밍 언어를 이용한 구현
- 기호처리언어(인공지능용 언어) : LISP, PROLOG 등 기호처리에 적합한 언어

□ 지식 공학 언어 : 지식기반 시스템 구성을 위해 제공되는 언어 - 전문가 시스템 셸(shell)

- 골격 시스템 : 기존의 전문가 시스템에서 지식 베이스 부분만 제거하고 난 추론기관과 그 외의 지원도구로 구성된 도구
- 범용목적 시스템 : 전문가 시스템 구축을 위해 만들어진 지식공학용 언어

## 제5장 논리에 의한 지식표현

### ▣ 주요 학습내용

- 이진논리의 개념
- 명제논리식의 표현 및 이용
- 명제논리의 한계 및 술어논리의 필요성
- 술어논리식의 표현
- 술어논리식의 도출연역

### ▣ 학습지침

명제논리의 정형식 및 표준형, 특히 연언표준형은 도출연역을 위해 숙지해 둔다. 술어, 객체, 객체변수, 함수 등의 기호를 활용하는 술어논리 정형식 표현방법을 학습한다. 술어논리식의 절분리 및 도출연역 과정은 강의내용을 참조하여 직접 손으로 따라 해 보면 쉽게 이해할 수 있다.

## 1. 명제논리

### ● 명제

- 기호논리학에서 어떠한 문장의 진위값을 참과 거짓이라는 2개의 이산적 값으로 표현
- 참-거짓을 명백하게 판별할 수 있는 문장을 명제라고 함

### ● 기본명제

- 더 이상 분해할 수 없는 최소단위의 명제

(예)  $p$  = 영수는 한국인이다.  
 $q$  = 존은 미국인이다.

- ☞ 명제에 기호를 부여하여 명제 자체의 내용보다는 기호를 대상으로 한 처리
- ☞ 원소식(atomic formula)

### ● 합성명제(복합명제)

- 2개 이상의 명제를 논리연산자로 결합한 것

(예)	$p$ = 영수는 한국인이다. $q$ = 존은 미국인이다.	→	$p \wedge q$ =영수는 한국인이고, 존은 미국인이다.
-----	-------------------------------------	---	--

### ● 논리연산자

- $\vee$ (OR),  $\wedge$ (AND),  $\sim$ (NOT)
  - ☞ 완전집합
- $\rightarrow$ (조건명제),  $\leftrightarrow$ (동치)
  - ☞  $\vee, \wedge, \sim$  으로 표현할 수도 있지만 특히 의미있는 연산자이므로 별도로 고려

● 조건명제

- p가 참인 경우에만 q에 따라 조건명제의 값이 변화하고, p가 거짓인 경우에는 q의 값에 관계없이 참
- $\sim p \vee q$ 와 동일한 진리표를 가짐

● 명제논리식

1. 기본명제  $p, q, r, \dots$ 은 논리식이다.
2.  $p$ 가 논리식이라면  $p$ 의 부정  $\sim p$ 도 논리식이다.
3.  $p, q$ 가 논리식이라면  $p \rightarrow q$ 도 논리식이다.
4. 1~3에 의해 얻어지는 식만이 논리식이다.

※ 논리연산자  $\sim$ 과  $\rightarrow$ 만 언급했지만  $\wedge$ 와  $\vee$ 도  $\sim$ 과  $\rightarrow$ 의 조합으로 구현할 수 있기 때문에 포함된 것과 같음

● 논리식의 표준형

- 명제기호와 논리 연산자를 조합하여 다양한 형태의 논리식이 결합될 수 있으나, 이를 표준화된 방법으로 표기할 필요가 있음

연언표준형	선언표준형
$F_1 \wedge F_2 \wedge \dots \wedge F_i \wedge \dots \wedge F_n, n \geq 1$ $F_i = P_{i1} \vee P_{i2} \vee \dots \vee P_{im}$ $P_{ij} : \text{리터럴}$	$G_1 \vee G_2 \vee \dots \vee G_i \vee \dots \vee G_n, n \geq 1$ $G_i = Q_{i1} \wedge Q_{i2} \wedge \dots \wedge Q_{im}$ $Q_{ij} : \text{리터럴}$

● 항진식 : 논리식을 구성하는 논리변수의 값에 관계없이 항상 참인 논리식

(예) 삼단논법 :  $\{(p \rightarrow q) \wedge (q \rightarrow r)\} \rightarrow (p \rightarrow r)$

● 추론방법

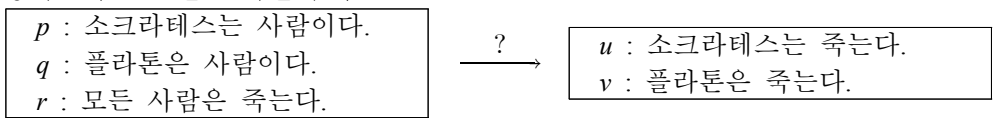
- 긍정논법(modus ponens) :  $p$ 와  $p \rightarrow q$ 가 모두 참이면  $q$ 도 참이
- 부정논법(modus tolens) :  $p \rightarrow q$ 와  $\sim q$ 가 사실이면  $\sim p$

**2. 술어논리의 필요성**

※ 명제논리에서 사실의 표현 : 명제기호

- ↳ 표현된 결과는 문장의 의미를 표현하는 것이 아니라 단지 기호일 뿐
- ⇨ 기호로부터 의미를 구하여 추론할 수 없음

(예) 명제논리로 표현된 사실의 추론



- 문장의 의미로 추론한다면 ‘소크라테스는 죽는다.’, ‘플라톤은 죽는다.’라는 사실을 얻을 수

있지만,  $p, q, r$ 이라는 기호로부터는 이러한 추론을 할 수 없음( $\because p, q, r$ 은 단지 기호일 뿐)

● 술어논리를 이용한 표현

- 명제를 구성하는 요소를 객체와 술어로 분리
- ‘술어(객체)’의 형태로 표현

(예) 소크라테스는 사람이다.  $\rightarrow Man(SOCRATES)$

플라톤은 사람이다.  $\rightarrow Man(PLATO)$

⇨ ‘소크라테스(SOCRATES)’와 ‘플라톤(PLATO)’이 모두 ‘사람이다(Man)’라는 공통적인 술어를 가짐

⇨ ‘모든 사람은 죽는다.( $\forall x\{Man(x) \rightarrow Mortal(x)\}$ )’라는 형태로 표현된 명제에 의해 ‘소크라테스는 죽는다.( $Mortal(SOCRATES)$ )’와 ‘플라톤은 죽는다.( $Mortal(PLATO)$ )’를 추론할 수 있음

**3. 술어논리식의 표현**

● 1차 논리(1st order logic)

- ‘술어(객체 리스트)’ 형태의 표현 : 1개 이상의 객체가 포함될 수 있음

(예) 소크라테스는 사람이다.  $\Leftrightarrow Man(SOCRATES)$

$X$ 가  $Y$  위에 있다.  $\Leftrightarrow On(X, Y)$

- 객체의 표현 : 객체 상수, 변수, 함수가 사용될 수 있음

(예) 변수의 사용

▪  $x$ 가 새라면  $x$ 는 날개가 있다.  $\Leftrightarrow \forall x\{Bird(x) \rightarrow Has\ Wings(x)\}$

- 한정기호(양화사, quantifier) : 변수가 사용될 경우 그 변수의 대상이 되는 범위를 정의하기 위해 사용함

▪ 전칭기호(universal quantifier,  $\forall$ ) : 모든 원소에 대하여 성립함

(예) 깃털을 가지고 있는 것은 모두 새이다.

$\Leftrightarrow \forall x\{Feather(x) \rightarrow Bird(x)\}$

▪ 존재기호(existential quantifier,  $\exists$ ) : 조건을 만족하는 원소가 적어도 1개 존재함

(예) 농구선수 중에는 키가 작은 사람도 있다.

$\Leftrightarrow \exists x\{BasketballPlayer(x) \wedge Short(x)\}$

※ 한정기호에 대한 등식

- ①  $\sim(\exists x) P(x) \equiv (\forall x) \sim P(x)$
- ②  $\sim(\forall x) P(x) \equiv (\exists x) \sim P(x)$
- ③  $(\forall x) P(x) \wedge Q(x) \equiv (\forall x) P(x) \wedge (\forall y) Q(y)$   
 $\Rightarrow P(x) \wedge Q(x)$ 는  $P(x)$ 와  $Q(x)$ 가 모두 참이어야 참이 되므로
- ④  $(\exists x) P(x) \vee Q(x) \equiv (\exists x) P(x) \vee (\exists y) Q(y)$   
 $\Rightarrow P(x) \vee Q(x)$ 를 만족하는  $x$ 가 존재한다는 것은  $P$ 를 만족하는 객체가 존재하거나  $Q$ 를 만족하는 객체가 존재함을 의미한다.
- ⑤  $(\forall x) P(x) \equiv (\forall y) P(y)$
- ⑥  $(\exists x) P(x) \equiv (\exists y) P(y)$

□ 함수기호 : 객체 사이의 관계를 나타내는 기호

(예)  $Korean(father(\text{철수}))$

$\Rightarrow father$  : 인수로 전달된 객체의 아버지에 해당되는 객체

● 술어논리식의 정형식

□ 항(term)

- ① 객체상수, 객체변수
- ②  $f(\text{항}_1, \text{항}_2, \dots, \text{항}_n)$ ,  $f$  : 함수기호
- ③ ①, ②에 의해 정의되는 것만이 항이다.

□ 술어논리 정형식(well-formed formulas, wff)의 순환적 정의

- ① 참( $T$ ), 거짓( $F$ )
- ②  $P(\text{항}_1, \text{항}_2, \dots, \text{항}_n)$   $P$  : 술어 기호
- ③  $P, Q$ 가 wff일 때  $\sim P, P \rightarrow Q$
- ④  $P$ 가 wff이고,  $x$ 가 객체변수일 때  $\forall xP, \exists xP$
- ⑤ ① ~ ④에 의해 정의되는 것만이 wff

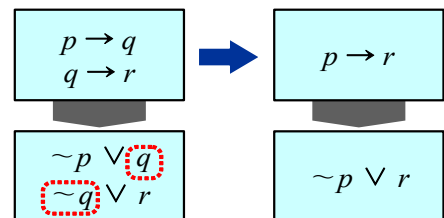
**4. 도출연역에 의한 정리증명**

※ 논리로 표현된 지식을 활용하는 대표적인 분야는 정리 증명

$\Rightarrow$  도출연역을 사용

● 도출

- $p \rightarrow q$ 이고  $q \rightarrow r$ 이면  $p \rightarrow r$ 이임
- 조건명제  $p \rightarrow q$ 와 동치인 논리식은  $\sim p \vee q$ ,  $q \rightarrow r$ 와 동치인 논리식은  $\sim q \vee r$
- $q$ 와  $\sim q$  쌍을 제거한 나머지를 논리함으로 묶은 결





론을 도출

● 기본 도출식

부모절	도출절
$p, \sim p \vee q$	$q$
$p \vee q, \sim p \vee q$	$q$
$\sim p, p$	$false$
$\sim p \vee q, \sim q \vee r$	$\sim p \vee r$

● 단일화

- 술어논리식의 도출을 위해서 객체변수를 객체상수 또는 다른 객체변수와 동일하게 만듦으로써 서로 달라 보이는 절을 동일한 절로 통일하는 과정
- 예 :  $\sim Father(x, y) \vee Male(x)$ 와  $Father(A, B)$ 에서  $x$ 와  $y$ 를 각각  $A$ 와  $B$ 로 단일화함으로써  $Male(A)$ 를 도출함

- 도출연역에 의한 정리 증명 과정 : 증명하고자 하는 정리를 부정한 후, 모순을 이끌어 냄으로써 그 정리가 참임을 보인다.

● 도출연역을 이용한 정리증명 알고리즘

1. 증명하고자 하는 정리를 부정하여 공리의 리스트에 첨가
2. 공리 리스트를 연언표준형으로 표현한 후 절분리
3. 도출 가능한 쌍이 없을 때까지 다음을 반복
  - 1) 도출 가능한 절의 쌍을 찾아 도출절을 구한다.
  - 2) 도출절을 공리 리스트에 추가한다.
  - 3)  $false$ 가 얻어지면 정리가 참임이 증명된다.
4. 정리가 거짓임을 알리고 끝낸다.

(예) 주어진 공리 :  $A, (A \wedge F) \rightarrow C, (D \vee E) \rightarrow B, (B \wedge G) \rightarrow F, E, G$

증명하고자 하는 정리 :  $C$

- 연언표준형으로 변환한 후 절(clause)을 분리
- 증명하고자 하는 정리를 부정하여 공리 리스트에 첨가

공리	절
$A$	$A$
$(A \wedge F) \rightarrow C$	$\sim A \vee \sim F \vee C$
$(D \vee E) \rightarrow B$	$\sim D \vee B, \sim E \vee B$
$(B \wedge G) \rightarrow F$	$\sim B \vee \sim G \vee F$
$E$	$E$
$G$	$G$
증명할 정리의 부정	$\sim C$

- 도출연역 과정 : 교재 <그림 5-2> 참조
  - 증명하고자 하는 정리를 부정함으로써 *false*가 도출되며, 참인 공리로부터 유도된 결과가 거짓이 되므로 모순
  - 따라서 증명하고자 하는 정리인 *C*는 참임이 증명됨

## 제6장 생성 시스템

### ▣ 주요 학습내용

- 생성 시스템의 구조
- 추론방법

### ▣ 학습지침

생성 시스템의 구성요소 및 추론과정의 이해가 중요하다. 문제풀이의 목적에 따라 어떠한 추론 방식을 사용하는 것이 바람직한지, 각 추론과정에서 추론사슬이 어떻게 형성되는지 강의 및 교재에 제시된 예를 통해 학습한다.

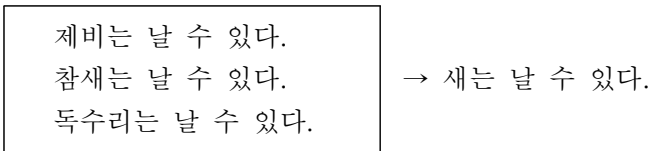
### 1. 생성 시스템의 구조

- 생성규칙(production rule) : 주어진 패턴에 따른 응답절차
  - 'IF 조건부 THEN 결론부' 형태의 규칙
    - 조건부 : 규칙이 선택되는 조건
    - 결론부 : 규칙이 선택되어 실행한 결과 제시하게 되는 결론 또는 행동
- 규칙 기반 시스템(rule-based system) : 문제 분야에서 적용될 수 있는 많은 규칙을 저장하여 활용 → 생성 시스템(production system)
- 생성 시스템의 구성
  - ① 생성 메모리 : 생성규칙을 모아 저장하고 있는 부분
  - ② 작업 메모리 : 현재의 상태를 표현하는 사실의 집합
    - 외부로부터 입력되거나 추론에 의해 얻은 사실이 저장
    - 작업 메모리의 내용은 상황에 따라 변화될 수 있으므로 단기 메모리(short-term memory, STM)라고도 함
  - ③ 인터프리터 : 현재의 작업 메모리의 내용에 따라 규칙을 선택하고, 이를 실행하는 역할을 함
- ※ 생성 메모리는 프로그램을 저장하고 있는 부분, 작업 메모리는 데이터를 저장하는 부분, 인터프리터는 프로그램을 실행시키는 것으로 생각할 수 있음
  - 프로그램의 실행은 미리 주어진 알고리즘에 따라 이루어지지만, 생성 시스템의 동작은 현재의 상태에 따라 규칙이 선택되어 실행됨
- ※ 인터프리터의 실행주기
  - ① 정합(matching) : 생성 메모리 내의 많은 규칙 중 작업 메모리의 내용에 의해 만족되는 것을 찾음
  - ② 충돌해결 : 현재상태에 의해 정합이 이루어지는 규칙이 여러 개 있을 수 있음. 이때 그중 실행할 규칙을 선택하는 과정
  - ③ 선택된 규칙의 실행

**2. 추론**

● 추론방법

- 연역법(deduction)
  - ‘A’와 ‘IF A THEN B’로부터 ‘B’를 추론
  - 정확한 전제로부터 정확한 결론을 추론할 수 있음
- 유도법(abduction)
  - ‘IF A THEN B’와 ‘B’로부터 ‘A’를 추론
  - 추론의 결과가 반드시 참임을 보장할 수 없다.
  - (예) ‘IF x가 어류 THEN x는 물에서 산다’, ‘x가 물에서 산다’→ ‘x는 어류’  
 ⇨ 많은 경우 참이지만 그렇지 않은 경우도 있음(예 : 고래).
- 귀납법(induction)
  - 공통적인 내용을 표현하는 사실들로부터 일반적 규칙을 찾아내는 추론방법
  - 항상 참인 규칙만을 만들어 내지는 못함
  - 학습과 관련
  - (예) 여러 가지 새에 대한 정보를 제공

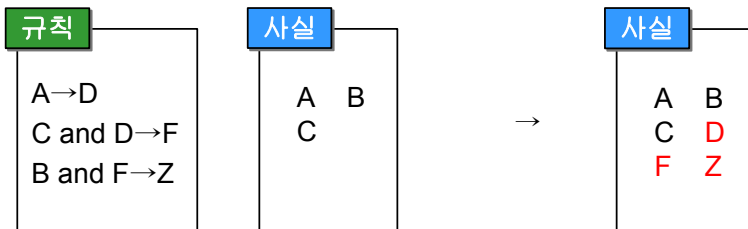


※ 규칙기반 시스템에서 주로 사용하는 추론방법은 연역법

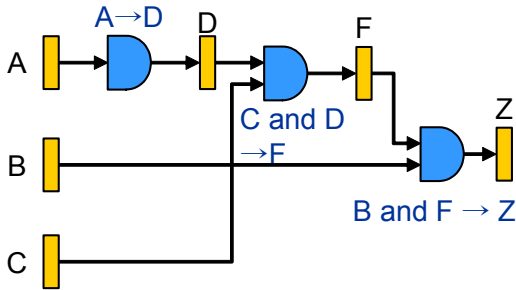
● 추론의 방향

- 전방향 추론(forward chaining) : 현재의 상태에 만족되는 조건부를 가지고 있는 규칙을 찾아 결론부에 해당되는 사실을 얻는 것
- 후방향 추론(backward chaining) : 얻고자 하는 결론(목표)을 미리 설정한 다음, 그 목표가 참인가를 알아내기 위해 결론부가 목표와 부합하는 규칙을 찾아, 그 규칙의 조건부가 만족되는가 판단하는 추론방법  
 - 만일 그 규칙의 조건부가 참이면 결론은 참임

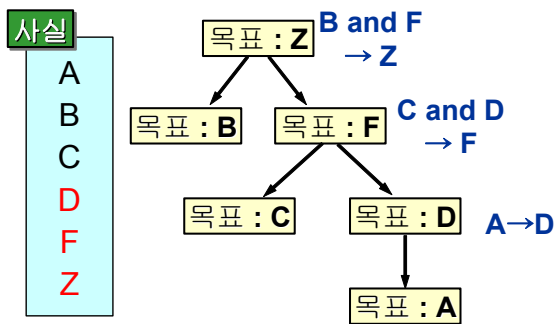
(예) 전방향 추론



⇒ 추론사슬



(예) 후방향 추론 : Z가 참인가?



● 추론방향의 결정요인

- 출발상태와 목표상태의 수
- 어느 방향으로의 분기율이 낮은가?
  - 분기율이 낮은 방향으로 진행하는 것이 유리
- 문제풀이를 시작하게 한 사건
  - 새로운 사실의 발견 : 전방향 추론
  - 어떤 사실에 대한 질문 : 후방향 추론

※ 생성 시스템을 사용할 때 고려할 사항

(1) 규칙의 정합

- 전체 규칙을 순차적으로 정합 : 규칙의 수가 많은 경우 시간 소비량이 큼
- 색인화
  - 규칙의 전제조건에 포함된 사실을 이용해 색인을 구성
  - 규칙의 전제조건이 특정상황의 명확한 기술인 경우 적합
- 필터의 사용
  - 규칙의 술어에 해당되는 특징을 정의
  - 작업 메모리의 변화에 의해 영향을 받는 규칙을 필터링

(2) 메타지식의 사용

- ※ 메타지식(meta-knowledge) : 지식의 사용에 대한 지식
- 어떠한 규칙의 집합을 선호하게 되는 조건을 제시 : 규칙 베이스의 규칙을 순차적으로 비교

하게 하지 않고, 정해진 조건에서는 특정한 규칙집합을 탐색하도록 함

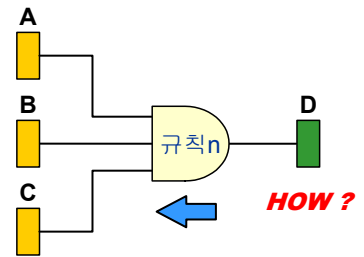
- 개별적인 문제풀이에 적합한 규칙의 적용순서 제시 : 규칙을 적용함으로써 얻게 되는 예상 효과, 다른 규칙과의 상호작용 등에 대한 지식의 활용
- 어떠한 목표(또는 하부목표)를 풀이하는 순서를 제시 : 주어진 문제에 대한 하부목표가 있을 때 어느 것을 먼저 풀이해야 하는가에 대한 순서제시

(3) 설명기능

- 사용자가 시스템에게 추론과정에 대한 질문을 했을 때 응답을 할 수 있게 하는 기능
- 추론사슬을 보여 줌으로써 응답
- 시스템이 내린 결론에 대한 신뢰감을 제공함
- 추론과정을 보여 줌으로써 규칙이 올바르게 활용되고 있는지 확인하는 데 도움이 됨

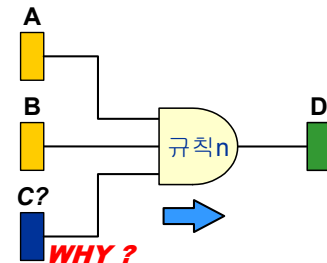
□ ‘어떻게?’에 대한 설명

- 어떻게 그러한 결론을 내렸는가?
  - ⇒ 적용한 규칙을 거슬러 올라가서 조건인 A, B, C가 만족되었고, 이를 전제조건으로 하는 규칙에 의해 얻었다는 응답



□ ‘왜?’에 대한 설명

- 시스템이 사용자에게 어떠한 사항을 질문했을 때, 왜 그러한 질문을 하는가를 되묻는 경우
- 어떠한 목표가 참임을 보이기 위한 전제조건을 사용자로부터 구하는 과정에서 나타날 수 있는 질문
  - ⇒ 해당되는 규칙을 보여 줌



## 제7장 퍼지이론

### ▶ 주요 학습내용

- 퍼지이론의 기본 개념
- 퍼지집합의 개념 및 연산
- 퍼지논리
- 퍼지이론의 활용

### ▶ 학습지침

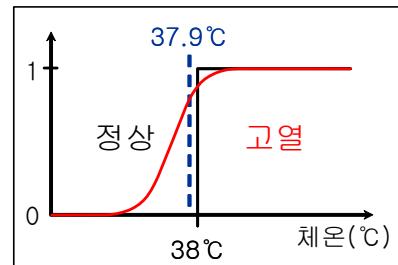
퍼지집합은 소속함수로 집합의 소속관계를 표현하며, 0과 1로 소속함수값이 한정된 고전집합의 일반화로 생각하면 이해가 쉽다. 퍼지집합 및 퍼지논리의 각 연산에서 소속함수가 어떻게 계산되는가를 예를 통해 계산해 보면 이해가 쉽다. 고전집합 및 고전논리의 연산 특성이 유지되는 점을 확인해 보되, 예외사항에 주의해야 한다. 퍼지추론의 경우 방대한 내용을 간략히 다루고 있어 이해가 쉽지 않을 수 있다. 전반적인 처리과정 및 개념 중심으로 학습한다.

### 1. 퍼지이론의 기본 개념

※ 컴퓨터는 0과 1로 표현되는 값을 다루는 이진논리를 사용하나, 이진논리로 접근하기 어려운 문제가 많음

(예) 정상체온과 고열의 판단

- 어떠한 판단작업을 이진논리로 처리하는 일반적인 방법은 적절한 임계치를 주어 그 값을 기준으로 판단함
- 체온의 판단 : 38℃를 기준으로 이를 넘으면 고열, 그렇지 않으면 정상으로 판단
- 체온이 37.9℃이면?



※ 퍼지이론 : 0과 1의 경직된 논리추론을 탈피하여 보다 유연한 판단과정을 구현하기 위한 이론

※ 1960년대 초 미국 캘리포니아 대학(UC 버클리)의 자데(Zadeh) 교수가 퍼지집합의 개념을 제시

※ 1965년 *Information and Control*에 <Fuzzy Sets>라는 논문을 발표

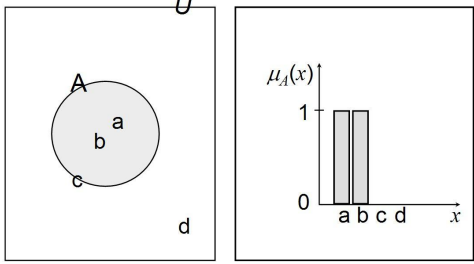
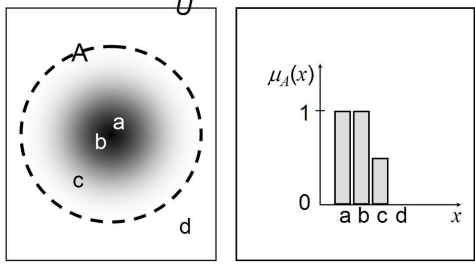
※ 제어, 예측, 의사결정, 전문가 시스템 등 많은 분야에서 활용

### 2. 퍼지집합(fuzzy sets)

※ 퍼지집합과 구분하기 위해 기존의 집합이론을 따르는 집합을 고전집합이라 칭함

● 고전집합과 퍼지집합

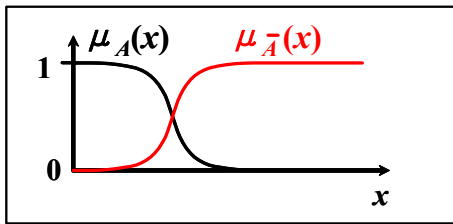
□ 소속함수  $\mu_A(x)$  :  $x$ 가 집합  $A$ 의 원소일 가능성을 나타내는 함수

고전집합	퍼지집합
<p>□ <math>x</math>가 집합 <math>A</math>의 원소이면 1 그렇지 않으면 0</p> $\mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$ <p>□ 집합의 경계가 명백하여, 포함 여부가 참거짓으로 명백하게 결정됨</p>  <ul style="list-style-type: none"> <li>▪ <math>a, b</math> : 집합 <math>A</math>의 원소</li> <li>▪ <math>c, d</math> : 집합 <math>A</math>의 원소가 아님</li> <li>▪ <math>a</math>와 <math>b</math>의 소속함수값은 1, <math>c</math>와 <math>d</math>의 소속함수 값은 0</li> </ul>	<p>□ <math>x</math>가 집합 <math>A</math>의 원소일 가능성을 0과 1사이의 값으로 표현</p> $\mu_A(x) \in [0, 1]$ <p>□ 집합의 경계가 확실히 구분되지 않으며, 포함관계가 모호함</p>  <ul style="list-style-type: none"> <li>▪ <math>a, b</math> : 집합 <math>A</math>의 원소가 확실</li> <li>▪ <math>c</math> : 집합 <math>A</math>의 원소일 가능성 0.5</li> <li>▪ <math>d</math> : 집합 <math>A</math>의 원소일 가능성 없음</li> </ul>

※ 퍼지집합에 대한 연산 : 고전집합에서 사용되는 연산에 해당되는 집합연산을 퍼지집합에 대하여 정의

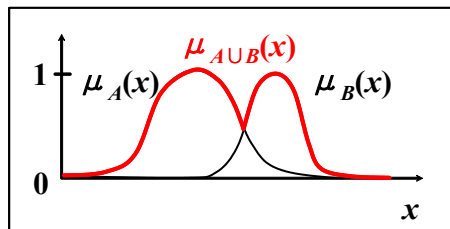
● 집합의 연산

□ 여집합



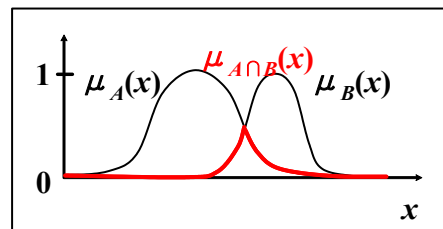
※  $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$

□ 합집합



※  $\mu_{A \cup B}(x) = \text{Max}[\mu_A(x), \mu_B(x)], \forall x \in U$

□ 교집합



※  $\mu_{A \cap B}(x) = \text{Min}[\mu_A(x), \mu_B(x)], \forall x \in U$



(예) 퍼지집합의 연산

$x$	$\mu_A(x)$	$\mu_B(x)$	$\mu_{A \cup B}(x)$	$\mu_{A \cap B}(x)$	$\mu_{\bar{A}}(x)$
$a$	1	0	1	0	0
$b$	0.7	0.2	0.7	0.2	0.3
$c$	0.4	0.5	0.5	0.4	0.6
$d$	0	1	1	0	1

**3. 퍼지논리(fuzzy logic)**

<p>● 고전논리 : 어떠한 명제의 참·거짓을 0과 1로 표현</p>	<p>● 퍼지논리 : 논리 변수의 값을 0과 1만으로 제한하지 않고, 0과 1 사이의 값으로 표현</p>

● 퍼지논리의 연산자

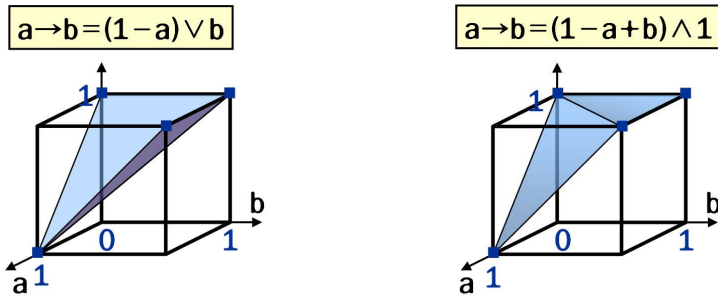
연산자	계산식
부정	$\bar{a} = 1 - a$
논리곱	$a \wedge b = \min(a, b)$
논리합	$a \vee b = \max(a, b)$
조건명제	$a \rightarrow b = \min(1, 1 - a + b)$

□ 퍼지논리 연산의 정의는 여러 가지 유형이 있으며, 위의 연산자 정의는 그중 대표적인 한 예

※ 조건명제가  $a \rightarrow b$ 가  $\sim a \vee b$ 으로 정의되지 않은 점에 주의

▪ 고전논리의 조건명제  $a \rightarrow b$ 의 4개의 진리값을 보간(interpolation)하는 평면의 비교를 통한 퍼지 조건명제 정의방법의 비교

-  $a \rightarrow b \equiv \sim a \vee b$ 인 관계를 이용할 경우  $a \rightarrow b = (1 - a) \vee b$ 가 되며, 이에 의해 형성되는 평면과 앞에서 정의한 수식에 의해 형성되는 평면을 비교하면 다음과 같음

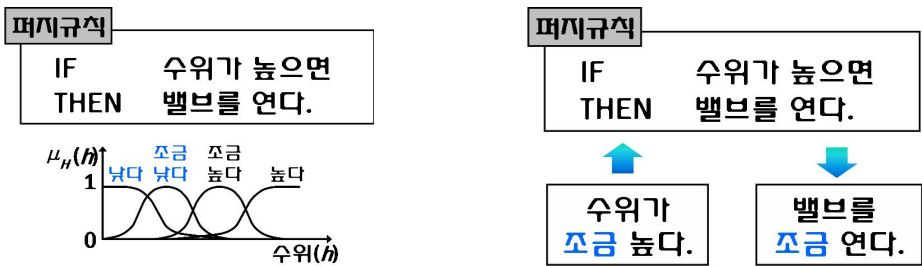


※ 차량제어 시스템 :  $a \rightarrow b$ 를  $a \wedge b$ 로 구현함으로써 보다 좋은 결과를 얻을 수 있음

#### 4. 퍼지이론의 활용

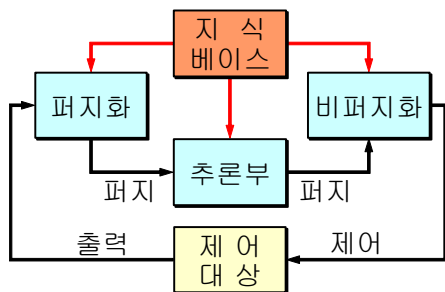
● 퍼지추론(근사추론, approximate reasoning)

- 퍼지이론을 이용한 추론
- 언어적 형태의 규칙을 사용
  - 조건의 정도를 나타내는 수식어를 통해 융통성 있게 규칙을 적용할 수 있도록 함
- 물탱크의 수위조절을 위한 규칙의 예



- 수위가 높은 정도에 따라 밸브를 여는 정도가 다른데, 이를 일일이 규칙을 정의하여 해결하려면 매우 많은 수의 규칙이 필요
- 또한 입력 파라미터의 값을 정확히 측정할 수 없다면 조건 자체가 모호한 값임  
→ 퍼지규칙을 활용

● 퍼지제어기의 구조

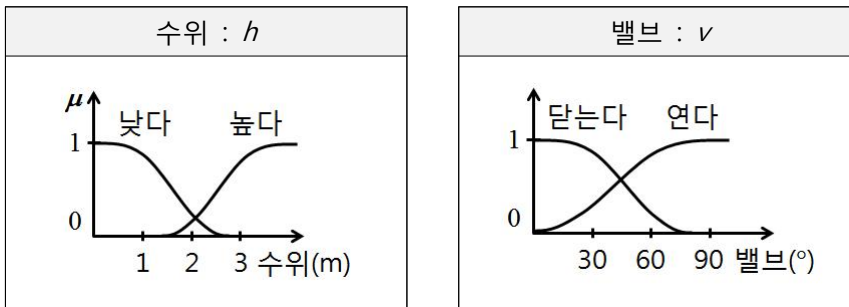


● 수위조절 제어 시스템의 예

- 규칙의 예

규칙 1	IF	수위가 높으면	IF	수위가 낮으면
	THEN	밸브를 연다.	THEN	밸브를 닫는다.

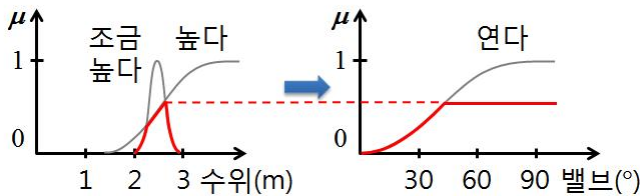
- 규칙의 조건부 및 결론부에 언어적 변수를 포함
  - 일반적 규칙과는 추론절차가 다름
- 수위에 따라 밸브를 여는 정도를 제어
- 수위, 밸브 개폐에 대한 퍼지 소속함수를 정의함



- 수위가 '조금 높을 때' 밸브를 얼마나 열어야 하는가?

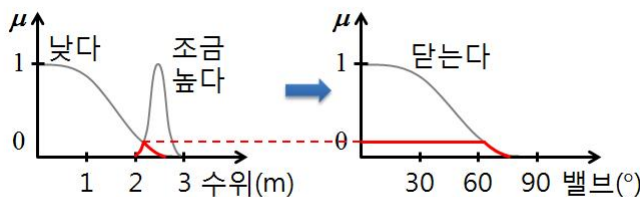
⇒ 추론과정

- 규칙 1의 적용



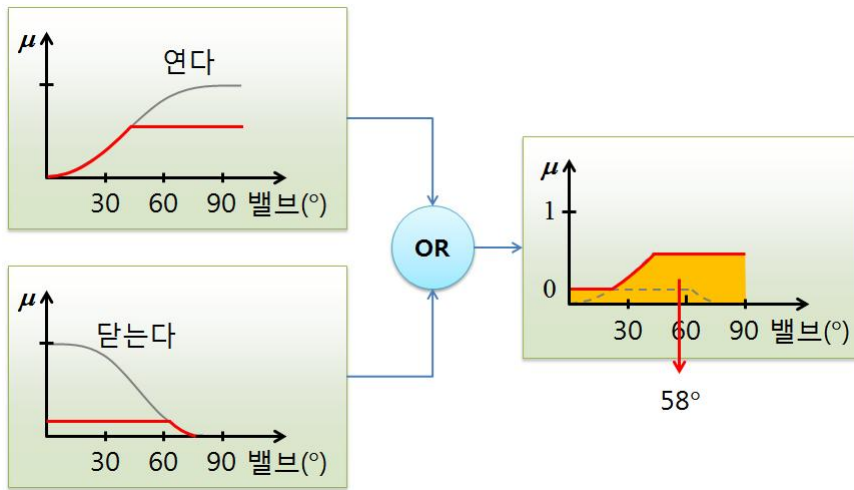
규칙1 : IF 수위가 높으면 THEN 밸브를 연다.

- 규칙 2의 적용



규칙2 : IF 수위가 낮으면 THEN 밸브를 닫는다.

- 결과의 종합



## 제8장 컴퓨터 시각

### ▶ 주요 학습내용

- 컴퓨터 시각 시스템의 전반적 처리 흐름
- 디지털 영상
- 영상의 필터링
- 영상분할
- 영상의 표현

### ▶ 학습지침

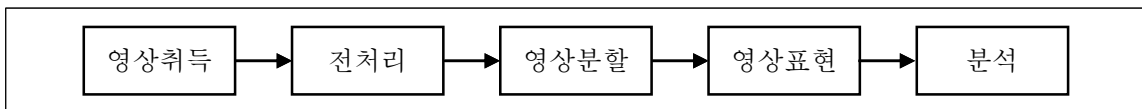
컴퓨터 시각 시스템은 컴퓨터를 이용하여 시각정보를 처리하고 정보를 추출하는 제반과정을 구현하고 있는 시스템이다. 디지털 영상의 개념과 이를 가공하는 여러 가지 처리과정에 대하여 이해하도록 한다.

### 1. 컴퓨터 시각 시스템의 개요

#### ● 컴퓨터 시각이란 무엇인가?

- 컴퓨터에게 인간의 시각체계에 해당되는 기능을 심어 넣기 위해 연구하는 인공지능의 한 분야
- 2차원 평면에 투영된 화상으로부터 유용한 정보를 획득하기 위한 여러 가지 처리 기능이 포함되어야 함

#### ● 컴퓨터 시각 시스템의 일반적 구조



- 영상취득 : 센서(카메라, 스캐너 등) 및 A/D 변환기를 통해 시각적 정보를 컴퓨터에 입력
  - 정보원의 종류 : 사진 촬영, 레인지 영상, X-선 단층촬영, 적외선촬영
- 전처리 : 취득한 영상을 보다 처리하기 좋은 형태로 가공
- 영상분할 : 영상을 몇 개의 의미 있는 영역으로 나누는 작업
- 영상표현 : 분할된 영역으로 구성되는 객체를 해석하기 위해 각각의 영역을 적절한 방법으로 컴퓨터 내에 묘사
- 분석 : 표현된 대상으로부터 필요한 정보를 구하는 단계

### 2. 디지털 영상

#### ● 픽셀

- 2차원 평면상의 하나하나의 점
- 일반적으로 사각형 격자 형태의 픽셀구조를 사용

□ 연결성 : 어떠한 점에 직접 연결된 주변 점에 대한 조건을 정한 것

4-이웃 연결성	8-이웃 연결성																		
<ul style="list-style-type: none"> <li>픽셀 p로부터 거리가 가장 가까운 4개의 픽셀을 이웃으로 정하는 것</li> </ul> <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr><td></td><td>n<sub>1</sub></td><td></td></tr> <tr><td>n<sub>4</sub></td><td>P</td><td>n<sub>2</sub></td></tr> <tr><td></td><td>n<sub>3</sub></td><td></td></tr> </table> </div>		n <sub>1</sub>		n <sub>4</sub>	P	n <sub>2</sub>		n <sub>3</sub>		<ul style="list-style-type: none"> <li>픽셀 p를 둘러싼 8개의 픽셀을 모두 이웃으로 정하는 것</li> </ul> <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr><td>n<sub>1</sub></td><td>n<sub>2</sub></td><td>n<sub>3</sub></td></tr> <tr><td>n<sub>8</sub></td><td>P</td><td>n<sub>4</sub></td></tr> <tr><td>n<sub>7</sub></td><td>n<sub>6</sub></td><td>n<sub>5</sub></td></tr> </table> </div>	n <sub>1</sub>	n <sub>2</sub>	n <sub>3</sub>	n <sub>8</sub>	P	n <sub>4</sub>	n <sub>7</sub>	n <sub>6</sub>	n <sub>5</sub>
	n <sub>1</sub>																		
n <sub>4</sub>	P	n <sub>2</sub>																	
	n <sub>3</sub>																		
n <sub>1</sub>	n <sub>2</sub>	n <sub>3</sub>																	
n <sub>8</sub>	P	n <sub>4</sub>																	
n <sub>7</sub>	n <sub>6</sub>	n <sub>5</sub>																	

● 디지털 영상의 취득

- 표본화 : 시간축에서 연속적으로 변화하는 신호를 이산(discrete)신호로 변환하는 것
  - 나이퀴스트 간격보다 더 조밀하게 표본화할 경우 원래의 신호를 완전히 복원해 낼 수 있음
- 양자화 : 값의 구간을 정해진 방법에 따라 분할하여 해당구간에 대한 대표값을 부여해 주는 과정
- 256단계의 그레이 스케일 영상
  - $I = \{g(x, y) \mid g(x, y) \in \{0, 1, \dots, 255\}, x \in \{0, 1, \dots, M-1\}, y \in \{0, 1, \dots, N-1\}\}$

**3. 전처리**

● 전처리(preprocessing) : 영상을 최대한 실제 신호에 가깝게 만들거나, 보다 처리하기 좋은 형태로 영상의 특성을 개선하는 과정

● 영상 필터링

- 필터 : 입력신호를 필터의 특성에 따라 변환하여 새로운 신호를 만들어 출력함
- 필터를 적용하는 기본적인 방법 :  $n \times n$  크기의 마스크를 이용하여 일정영역의 픽셀의 가중치 합을 구함

● 평활화(smoothing)

- 영상에 포함된 고주파 잡음을 제거하기 위해 한 픽셀의 값을 그 픽셀을 중심으로 한 이웃 픽셀과의 가중치 평균을 구하는 것
- 영상신호에 대해 저역통과 필터를 취하는 의미
- 예 : 가우시안 필터를 근사화한 필터 마스크를 적용함

$$M_{LP1} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \qquad M_{LP2} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- 잡음성분은 줄어드는 대신 전반적으로 영상이 흐려지는 현상이 일어난다.
- 예 : 중간값 필터
  - 원 영상의 각각의 픽셀 위치에 대하여 그 위치를 중심으로  $n \times n$  영역에 속한 픽셀의 그레이 레벨 값을 크기순으로 나열한 후, 그중 중간값을 그 픽셀위치에 대한 결과영상의 픽셀값으로

취하는 것

- 점잡음(salt and pepper noise)에 효과적임

## 4. 영상분할

### ● 영상분할

- 주어진 영상을 유사한 속성을 갖는 부분으로 분할하는 작업
- 좋은 영상분할 알고리즘의 조건
  - ① 영상 내의 각각의 영역은 정해진 기준에 대해 속성이 균일해야 함
  - ② 영역 내부는 단순해야 하며 미세한 작은 구멍이 적어야 함
  - ③ 인접한 영역은 서로 상당히 다른 속성을 가져야 함
  - ④ 영역의 경계는 단순하고 부드러워야 하며, 공간적으로 정확한 위치에 존재해야 함

### ● 이진화

- 영상의 밝기 등에 의해 ‘0’ 또는 ‘1’의 두 영역으로 구분하는 것
- 영상의 히스토그램으로부터 임계치를 결정하여 이를 기준으로 영역을 구분
- 임계치 결정방법의 예
  - 선형적 지식에 따라 두 영역에 속하는 픽셀수의 비율을 정하고, 히스토그램에서 이 비율에 따라 임계치를 정하는 방법
  - 영상밝기 히스토그램이 2개의 피크를 갖는 2정점(bi-modal) 형태인 경우 두 피크 사이에 존재하는 최솟값을 임계치로 선택하는 방법

### ● 영역분할

- 영상의 공간적 특성을 이용한 방법
- 분할과 합병 알고리즘
  - ① 만일 하나의 정사각형 영역이 균일한 영상이 아니라면 이를 네 개의 정사각형 영역으로 분할
  - ② 4개의 정사각형 영역을 합병한 정사각형 영역이 균일하다면 이들을 하나의 영역으로 합병
  - ③ 위의 두 과정의 반복을 마치면 모든 영역에 대해 인접한 영역과 균일성을 검사하여, 만일 균일하다면 이들을 하나의 영역으로 합병

### ● 경계검출

- 에지(edge) : 밝기가 다른 두 영역 사이의 픽셀의 그레이 레벨이 크게 변하는 부분
- 1차 미분을 이용하는 방법
  - 예 : 소벨 연산자
    - 2개의 필터를 이용하여 에지의 강도와 방향을 구함

$$M_{SobelH} = \frac{1}{4} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad M_{SobelV} = \frac{1}{4} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- 2차 미분을 이용하는 방법
  - 예 : 라플라스 연산자
    - 필터링 결과의 0 교차점을 찾아 경계를 구함

$$M_{Lap} = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

## 5. 영상의 표현

- 영상의 표현 : 물체를 인지하거나 정보를 구하기 위하여 대상영역을 적절한 자료 및 구조로 표현
- 영상표현 방법의 예
  - 이진영상 표현 : 문자를 인식하는 경우와 같이 2진패턴을 인식하고자 하는 경우 사용
    - 그물망(mesh) 표현 : 영역의 크기가 클 경우 데이터의 수를 줄이기 위해 전체 영역을 정해진 크기의 소영역으로 나눈 후, 각각의 소영역에 속한 '1'의 수를 저장
  - 경계표현 : 물체를 구성하는 경계선으로 물체의 외양을 표현
    - 체인코드 : 물체 경계선을 추적하여 이웃하는 점이 위치한 방향코드를 나열하여 경계표현
    - 정규화된 체인코드 : 시작점에 따라 다른 코드가 나오지 않도록 하기 위해 나열된 코드의 값이 가장 작은 값이 되도록 시작점을 정해 만든 체인코드
    - 차분방향 코드 : 체인코드에서 이전 코드와 현재 코드의 차를 이용하여 표현함으로써 회전된 물체를 일관적으로 표현할 수 있게 함
    - $\psi-s$  곡선 : 체인코드를 일반화한 곡선



## 제9장 패턴인식

### ▶ 주요 학습내용

- 패턴인식 개요
- 패턴인식 시스템
- 결정이론적 패턴인식 방법
- 언어이론적 패턴인식 방법

### ▶ 학습지침

패턴인식은 인간의 인식능력을 컴퓨터에 구현하는 실용적인 연구 분야이다. 제9장에서는 패턴인식을 수행하기 위한 처리단계를 학습한다. 이 처리단계의 세부사항은 응용 분야에 상당히 의존적이지만, 각 단계마다 어떠한 처리가 이루어지는지 개괄적으로 이해한다. 결정이론적 패턴인식 방법에 대해서는 특징공간과 특징공간에서의 결정경계를 어떻게 구하는가에 관심을 가지고 학습한다.

### 1. 패턴인식이란 무엇인가?

#### ● 패턴인식

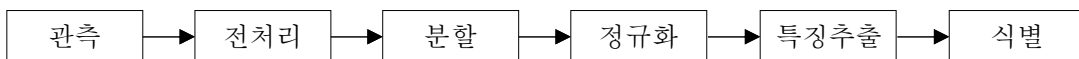
- 인간의 지각능력을 본떠서 만든 프로그램에 의해 컴퓨터가 문자, 물체, 음성 등을 식별하도록 하는 것
- 인간의 패턴인식 능력을 구현할 수는 없지만, 제한된 응용 분야에 맞는 인식기능을 구현하는 것은 충분히 가능성이 있음

### 2. 패턴인식 시스템의 구조

#### ● 패턴인식 시스템의 구성

- 입력장치 : 외부의 인식대상을 전기적 신호로 변환하기 위한 기구
  - 비디오 카메라, 마이크, 스캐너, 각종 센서 등
- A/D 변환기 : 아날로그 전기신호를 컴퓨터가 다룰 수 있는 디지털값으로 변환하는 장치
- 입력된 패턴을 가공하고 식별하기 위한 소프트웨어가 탑재된 컴퓨터 시스템

#### ● 패턴인식의 처리단계



- 관측 : 대상패턴을 입력장치를 통하여 취득하는 단계
- 전처리 : 입력된 데이터를 가공하여 보다 처리하기 적합하게 만드는 작업
- 분할 : 입력된 신호로부터 인식대상 패턴만 분리하는 과정
- 정규화 : 표준화된 형태로 패턴을 변환
- 특징추출 : 실제 인식에 사용될 데이터를 뽑아내는 단계

□ 식별

※ 처리 순서는 응용대상에 따라 차이가 있을 수 있음

● 패턴의 관측

- 카메라, 마이크, 센서 등 대상에 맞는 입력장치를 통해 대상물을 전기적 신호로 변환
- ⇒ 아날로그 신호를 디지털 신호로 변환(A/D 변환)
  - ↳ 표본화 및 양자화

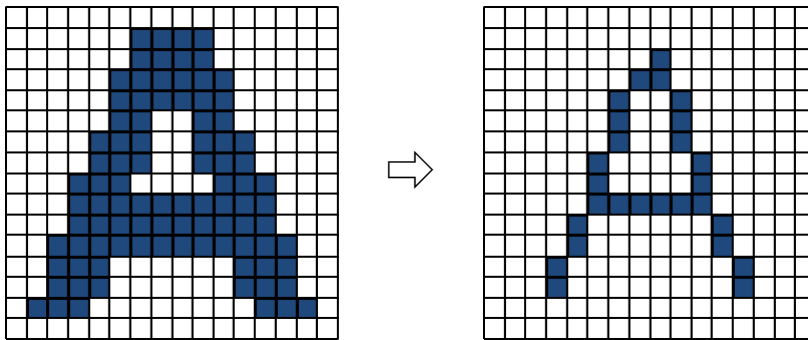
● 전처리(preprocessing) : 잡음제거, 신호의 개선 등

(예) 필터를 사용한 잡음 제거

- 중간값(median) 필터, 평활화(smoothing) 필터 등
- 잡음을 제거함으로써 인식하고자 하는 패턴을 정확하게 추출할 수 있음

(예) 세선화

- 패턴 구성요소의 연결성을 잃지 않는 범위에서 불필요한 픽셀을 제거
- 원래의 패턴으로부터 추출하기 어려운 구조적 정보 추출 용이



● 분할

- 인식하고자 하는 대상패턴이 다른 패턴이나 배경과 함께 있을 때, 인식대상 패턴을 분리해 내는 작업

● 정규화

- 패턴인식을 하려면 학습과정을 통해 대상패턴에 대한 정보를 취득
- 학습할 때와 동일한 조건에서 판단할 수 있는 정보를 취득하기 위해 정규화
  - 크기, 진폭, 위치 등

● 특징추출

- 특징
  - 인식대상의 패턴 중에서 어떠한 패턴을 다른 패턴과 구분하기 위해 사용될 수 있는 정보
  - 기호형태 또는 수치 데이터

- 특징추출의 목적

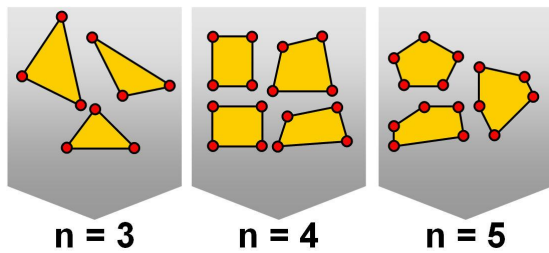
- 패턴공간의 차수를 낮춤
- 판별작업에 더욱 적합하도록 의미 있는 값을 얻음

□ 특징선택의 조건

- 계산에 의해 얻을 수 있어야 함
- 다른 패턴과의 구분에 도움이 되는 것이어야 함
- 유용한 정보를 잃지 않으면서 데이터를 처리할 수 있는 양으로 줄일 수 있어야 함

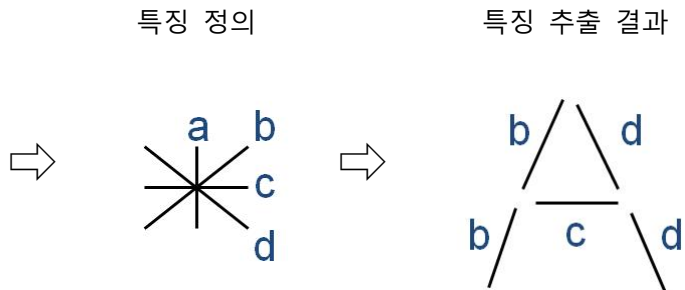
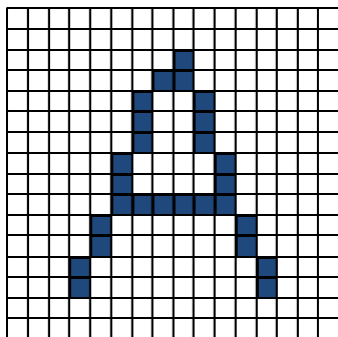
(예) 다각형의 식별

- 대상패턴이 단순히 다각형만을 대상으로 식별하는 경우에는 불필요한 많은 영상 데이터를 사용할 필요 없이, 꼭짓점수를 검출하여 이를 특징으로 삼으면 쉽게 식별할 수 있음



(예) 세션화된 문자패턴에서 획 특징의 추출

- 획의 방향성에 따라 네 가지로 구분하고, 각각에 a~d의 기호를 부여



● 식별

□ 입력된 패턴이 학습된 패턴 중 어느 것에 속하는가를 판별하는 단계

□ 식별방법의 종류

- 결정이론적 방법 : 각각의 패턴에 대한 특징값의 확률분포 형태를 분석하여 만든 식별함수를 통해 구분
- 언어이론적 방법
  - 원시패턴을 정의
  - 원시패턴이 어떠한 규칙에 따라 나열되어 하나의 패턴을 구성한다고 가정
  - 원시패턴의 나열규칙을 문법으로 정의하여 그에 따라 구문분석을 함으로써 패턴을 인식
- 신경회로망, 퍼지 이론 등을 통한 방법

**3. 결정이론적 패턴인식 방법**

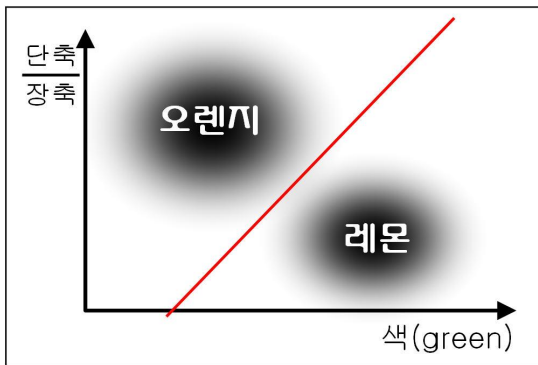
(1) 개요

※ 어떠한 패턴은 특징추출 과정에서 구해진 특징으로 표현

● 특징공간 : 패턴을 구성하는 특징요소를 좌표축으로 하여 형성된 공간

(예) 오렌지와 레몬의 식별

- 특징 : 단축과 장축의 비, 색
- 특징공간



- 패턴의 인식 : 특징공간의 두 그룹을 분리하는 함수를 정의하여 인식  
↳ 장·단축 비와 색의 함수

● 결정이론적 패턴인식 방법

- 패턴을 구성하는 특징값의 통계적 분포를 이용하여 인식하는 확률통계적 방법
- 특징값의 형태 : 수치값
- 각각의 패턴은 각각의 특징을 한 축으로 하는 특징공간상의 한 점으로 표현  
↳  $(x_1, x_2, \dots, x_d)$ 로 표현되는 특징 벡터
- 특징공간상에 분포하는 각 패턴그룹의 경계를 찾아, 그 경계를 기준으로 패턴을 식별함

(2) 거리측정자

● 거리측정자  $J$ 가 타당한 의미를 갖기 위해 만족해야 할 4가지 공리

- ①  $J(\mathbf{x}, \mathbf{y}) = 0$  iff  $\mathbf{x} = \mathbf{y}$
- ②  $J(\mathbf{x}, \mathbf{y}) \geq 0$
- ③  $J(\mathbf{x}, \mathbf{y}) = J(\mathbf{y}, \mathbf{x})$
- ④  $J(\mathbf{x}, \mathbf{y}) + J(\mathbf{y}, \mathbf{z}) \geq J(\mathbf{x}, \mathbf{z})$

● 유클리드 거리 : 두 점 사이의 직선거리

$$J_e(\mathbf{x}_k, \mathbf{x}_l) = \left[ \sum_{i=1}^d (x_{ik} - x_{il})^2 \right]^{1/2}$$

● 해밍거리 : ‘예, 아니오’, 또는 ‘있다, 없다’ 등의 불값(Boolean value)을 다루는 경우 적합

$$J_H(\mathbf{x}_k, \mathbf{x}_l) = \sum_{i=1}^d (x_{ik} \oplus x_{il})$$

- 도시블록 거리(맨해튼 거리)

$$J_{cb}(\mathbf{x}_k, \mathbf{x}_l) = \sum_{i=1}^d |x_{ik} - x_{il}|$$

- 마할라노비스(Mahalanobis) 거리

- 특징 간의 상관관계를 고려함으로써 비율의 변화에 영향을 받지 않도록 정의한 거리
- 어떠한 군집의 평균벡터가  $\mu$ 이고 공분산행렬(covariance matrix)이  $\Sigma$ 인 경우, 어떠한 특징 벡터  $\mathbf{x}_1$ 과 평균벡터  $\mu$  사이의 마할라노비스 거리의 제곱  $D^2$ 은 다음과 같다.

$$D^2 = (\mathbf{x}_1 - \mu)^T \Sigma^{-1} (\mathbf{x}_1 - \mu), \Sigma = E(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T$$

(3) 베이즈 결정이론

- 베이즈 분류기의 패턴 식별기준

- 특징을 추출한 결과가  $\mathbf{x}$ 일 때 두 클래스  $C_1$ 과  $C_2$  중 하나로 식별하는 기준

$$p(C_1 | \mathbf{x}) > p(C_2 | \mathbf{x}) \text{이면 } \mathbf{x} \text{는 } C_1$$

$$p(C_1 | \mathbf{x}) < p(C_2 | \mathbf{x}) \text{이면 } \mathbf{x} \text{는 } C_2$$

- 베이즈 규칙에 따라 다음과 같이 표현할 수 있음

$$p(\mathbf{x} | C_1) \cdot p(C_1) > p(\mathbf{x} | C_2) \cdot p(C_2) \text{이면 } \mathbf{x} \text{는 } C_1$$

$$p(\mathbf{x} | C_1) \cdot p(C_1) < p(\mathbf{x} | C_2) \cdot p(C_2) \text{이면 } \mathbf{x} \text{는 } C_2$$

- 식별 대상 클래스가  $N$ 개인 경우로의 일반화

$$p(\mathbf{x} | C_i) \cdot p(C_i) = \max_{k=1}^N p(\mathbf{x} | C_k) \cdot p(C_k) \text{이면 } \mathbf{x} \text{는 } C_i$$

(4) 인식 방법의 예 :  $k$ -NN

- 학습

- 학습표본들을 모두 저장함
- 각각의 학습표본에는 특징벡터의 값과 함께 그 표본이 어느 클래스에 속하는가의 정보가 지정됨 : 지도학습

- 식별

- 미지의 패턴  $\mathbf{x}$ 가 주어졌을 때 전체 학습표본 패턴과  $\mathbf{x}$ 와의 거리를 구하여 그 거리가 가장 가까운  $k$ 개의 학습표본을 선택함
- 선택된 표본수가 가장 많은 클래스에 속하는 것으로 식별함

(5) 선형 판별식에 의한 식별

- $n$ 차원 공간의 초평면(hyperplane)의 방정식

$$w_1x_1 + w_2x_2 + \dots + w_nx_n + w_0 = 0$$

⇒ 판별식  $D_i(x_1, x_2, \dots, x_n)$ 의 부호에 따라 패턴을 식별

$$D_l(x_1, x_2, \dots, x_n) = w_1x_1 + w_2x_2 + \dots + w_nx_n + w_0$$

#### 4. 언어이론적 패턴인식 방법

※ 언어이론적 패턴인식 방법 : 패턴의 구조적 특성을 파악하여 인식하기 위한 방법

- 사람이 사용하는 언어 : 단어들이 정해진 문법규칙에 따라 나열되어 문장을 구성
- 몇 개의 기본 패턴이 정해진 결합규칙에 따라 연결되어 패턴을 형성
- 원시패턴(primitives) : 패턴을 구성하는 기본 요소

● 언어이론적 패턴인식 시스템 (교재 250쪽, <그림 9-15> 참조)

- 패턴의 분석 : 표본패턴으로부터 원시패턴 및 추출방법을 정의하고, 이를 분석하여 문법을 정의 → 학습과정
- 인식과정
  - 입력패턴으로부터 원시패턴을 추출
  - 정의된 문법에 따라 구문분석을 하여 받아들여지는 패턴으로 인식

● 패턴의 정의 : 문법

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

여기서  $V_N$  : 비단말기호

$V_T$  : 단말기호 - 원시패턴

$P$  : 생성규칙

$S$  : 시작기호 - 패턴

● 언어이론적 패턴인식 방법의 특징

- 패턴의 구조적 해석에 의한 인식
- 패턴의 구조체계를 유지하는 범위 내에서 패턴의 변형 수용
- 원시패턴 추출이 어려움
- 생성규칙 자동생성이 어려움

## 제10장 학습과 신경회로망

### ▶ 주요 학습내용

- 학습의 개념
- 기계적 학습
- 결정트리의 학습
- 군집화
- 신경회로망의 학습

### ▶ 학습지침

단순한 기억방법부터 보다 개념화된 학습방법에 이르는 여러 가지 학습방법을 학습한다. 신경회로망의 정보처리 방식이 일반적인 컴퓨터의 처리방식과 어떻게 다른지 유념하여 학습한다. 뉴론의 동작원리, 활성화함수, 연결형태 등을 이해한다. 퍼셉트론을 비롯한 여러 가지 신경회로망 모델의 특성 및 학습방법에 대하여 이해한다.

### 1. 학습이란 무엇인가?

#### ● 학습이란

- 허버트 사이먼 : 이전에 수행한 경험이 있는 동일한 모집단에 대한 작업을 다음에는 보다 효율적이고 효과적으로 수행할 수 있게 하는 시스템의 적응적 변화

#### ● 학습형태

- ① 지도학습(supervised learning)
  - 입력과 그 입력을 가했을 경우 기대되는 출력의 쌍을 학습 데이터로 사용
  - 입력에 대한 출력이 기대되는 출력과 다르면 그 차이를 해소할 수 있도록 학습
- ② 자율학습(unsupervised learning, 비지도학습)
  - 학습 데이터는 입력 데이터로만 구성되며, 기대되는 출력은 제시하지 않음
  - 유사한 입력에 대해서는 동일한 출력을 낼 수 있도록 학습이 이루어짐
    - 예) 군집화
- ③ 강화학습(reinforcement learning)
  - 교사에 의해 무엇을 할 것인가를 지시받는 것이라기보다는 보상을 통해 학습

### 2. 기계적 학습

#### ● 기계적 학습

- 가장 초보적인 학습방법 : 문제와 풀이결과를 그대로 저장하여 동일한 상황에서 다시 사용할 수 있도록 하는 것
- 입력 :  $(x_1, x_2, \dots, x_m)$
- 처리  $f$ 의 결과 :  $(y_1, y_2, \dots, y_n)$

- ⇒  $[(x_1, x_2, \dots, x_m), (y_1, y_2, \dots, y_n)]$  쌍을 지식으로 저장
- ⇒ 이후  $(x_1, x_2, \dots, x_m)$ 이라는 문제가 나타나면 다시 풀이하지 않고 이를 검색하여  $(y_1, y_2, \dots, y_n)$ 이라는 해를 제시

※ 기계적 학습 : 계산하는 과정 보다 저장 및 검색 비용이 유리할 경우 적합

(예) 새뮤얼의 서양장기 프로그램

- 장기나 바둑 : 몇 수 앞을 머릿속에 두어 본 후 유리한 수를 둬. 이때 수를 더 깊이 내다 볼수록 좋은 수를 둘 수 있음
- 어떠한 상태에 대해 계산했던 평가함수값을 기억해 두면 다음번 동일한 상황이 발생했을 때 이를 이용하여 수를 더 깊이 내다본 효과를 낼 수 있음

● 기계적 학습 시 필요한 능력

- 정보를 체계적으로 저장하여 효율적으로 검색할 수 있도록 함
  - 새뮤얼의 서양장기 프로그램 : 말의 수와 같은 몇 개의 중요한 특성에 대하여 인덱싱하여 판의 형태를 저장
- 일반화를 통하여 저장능력을 향상시키는 기능이 필요
  - 새뮤얼의 서양장기 프로그램 : 백을 기준으로 하여 저장, 대각선을 중심으로 회전시킨 것을 포함

### 3. 귀납적 학습

● 귀납적 추론

- 반복적인 관찰이나 경험을 기반으로 하여 논리를 이끌어 내는 추론 방법

● 귀납적 추론에 의한 학습

- 학습 예들을 그대로 암기하는 것이 아니라 이를 **일반화**하여 표현함
- 학습에 사용된 표본이 모든 가능한 데이터를 포함하지 못하므로 학습에 사용된 예가 아닌 대상에 적용하게 되면 잘못된 결론을 내릴 수 있음
  - 거짓 양성(false positive) : ‘참’이 아닌데 ‘참’으로 잘못 판단하는 경우
  - 거짓 음성(false negative) : ‘참’인데 ‘거짓’으로 잘못 판단하는 경우

● 학습에 오류를 유발하는 학습 예의 결함

- ① 부정확한 입력 속성값이 존재하는 경우
- ② 학습 예를 잘못 분류한 경우
- ③ 고려해야 할 주요 속성이 빠진 경우

### 4. 결정트리의 학습

● 결정트리



- 분할정복 방식으로 입력공간을 분할하여 입력된 대상을 분류하거나 회귀분석을 하기 위한 트리

● 결정트리의 학습

- 제시된 학습표본들을 이용하여 지도학습 방식으로 학습
- 각각의 노드에서는 그 노드에 분류된 학습표본 부분집합에 대해 불순도검사 시행
  - 불순도가 높으면 적절한 검사를 통해 이를 분할할 수 있는 비단말노드를 생성
  - 불순도가 0(또는 정해진 임계치 미만)이면 해당 클래스(또는 다수의 표본이 속하는 클래스)를 출력하는 단말노드를 생성

## 5. 선형 식별함수의 학습

● 선형 식별함수

- 어떠한 패턴  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ 을 식별하기 위한 판별식  $D(\mathbf{x})$ 는 각각의 특징값에 가중치  $w_1, w_2, \dots, w_n$ 을 적용한 가중 합으로 계산함

$$D(\mathbf{x}) = w_1x_1 + w_2x_2 + \dots + w_nx_n + w_0$$

● 선형 식별함수의 학습

- 학습표본 집합을 이용하여 가중치  $w_0, w_1, w_2, \dots, w_n$ 의 값을 조정함으로써 적절한 결정경계를 형성하도록 하는 것
- 선형함수로 식별하기 어려운 경우는 여러 개의 선형 판별식을 결합하여 근사화할 수 있음

## 6. 군집화

● 군집화

- 패턴분포가 주어졌을 때, 같은 종류라고 생각할 수 있는 몇 개의 서브클래스로 분할하는 것
- 자율학습

●  $k$ -평균 군집화

- 학습표본 집합을 대표하는  $k$ 개의 평균벡터를 찾음
- 평균벡터의 초기 추정치로 임의의 값을 부여한 다음 반복적으로 이를 수정하여 적절한 위치로 수렴하도록 하는 방법을 사용
  - 학습표본을 가장 가까운 평균벡터로 분류
  - 각각의 평균벡터에 대해 그 군집으로 분류된 표본의 평균을 구하여 평균벡터를 수정

## 7. 신경회로망의 기본구조

● 두뇌 속의 신경구조

- 약 10억~100억 개의 신경세포(뉴론)
- 각 뉴론은 약 1천~10만 개의 서로 다른 신경세포와 신경연접을 통해 연결
- 신경연접의 특성에 따라 신호가 전달이 증폭되거나 감쇄되어 전달

※ 컴퓨터와의 비교

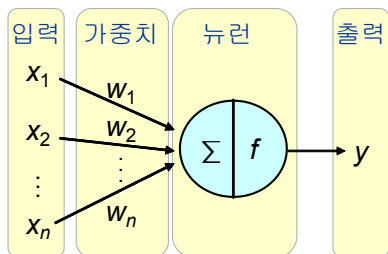
	CPU	뉴론
처리속도	수 나노초( $10^{-9}$ 초) 단위	수 밀리초( $10^{-3}$ 초) 단위
처리방법	나열된 명령어를 순차적으로 실행	대단위 병렬처리

※ 신경구조를 모델링하여 지능적 처리에 응용하고자 하는 시도 : 인공신경회로망

● 신경회로망의 특성

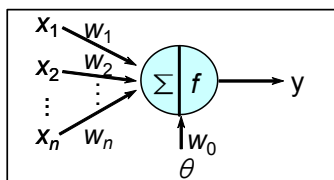
- 복잡한 기능을 수행하기 위해 많은 명령어를 포함하고 있는 CPU와는 달리 입력된 내용을 합하고, 간단한 변환을 하는 기능을 포함하는 많은 수의 뉴론으로 구성
- 각각의 뉴론은 다른 뉴론과 방대한 연결을 유지
- 수많은 뉴론이 동시에 동작하는 병렬처리 기능 : 신경망칩의 개발
- 정보의 저장 : 신경연접의 연결강도를 통해 저장
- 신경연접은 수많은 뉴론 간에 존재하며, 따라서 정보의 저장은 분산됨
- 학습능력 : 학습 데이터에 따라 자동적으로 연결강도를 조정할 수 있는 알고리즘 존재
- 결함내성 : 몇 개의 뉴론에 고장이 발생해도 전체 시스템의 성능이 크게 저하되지 않음

● 뉴론의 기본 구조



- 입력 : 외부 또는 이전 단으로부터 정보를 전달 받음
- 연결가중치 : 학습에 따라 조정
- 뉴론 : 입력을 합한 결과에 변환( $f$ )을 하여 출력

● 뉴론의 연산 기능

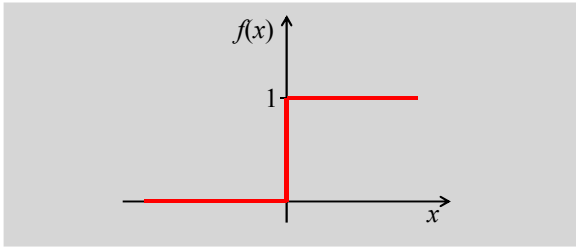


$$y = f\left(\sum_{i=1}^n x_i w_i - \theta w_0\right)$$

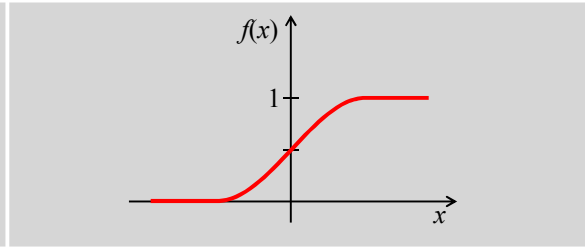
여기서  $\theta$  : 임계치,  $f$  : 활성화함수

- 입력에 가중치를 곱한 값들의 합에 임계치를 적용한 후 활성화함수(전달함수)를 통과한 결과를 출력
  - 활성화함수 : 비선형함수 사용

● 활성화함수의 종류



□ 하드 리미트(hard limit) : 임계값을 취한 결과가 0보다 크면 1, 그렇지 않으면 0



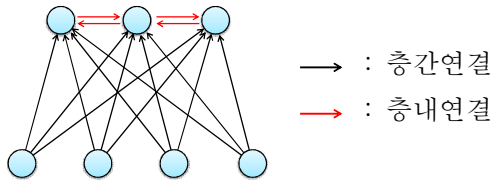
□ 시그모이드(sigmoid)  
 ▪ 미분 가능한 함수  
 ▪ 자동 이득조절 특성 : 작은 입력에 대해서는 이득이 크고, 큰 입력에 대해서는 이득이 작음

● 연결형태의 종류

① 흥분성 연결 및 금지연결

흥분성 연결	금지연결
<p>흥분성 연결 : 상대방을 강화</p>	<p>금지연결 : 상대방의 출력을 약화시키는 연결</p>

② 층내연결, 층간연결



□ 층내연결 : 동일 층 내 뉴론과의 연결  
 □ 층간연결 : 서로 다른 층간의 연결

③ 피드포워드 연결 및 순환연결

피드포워드 연결	순환연결
<p>출력층</p> <p>입력층</p> <p>피드포워드(feedforward) 연결 : 입력단에서 출력단 방향으로 데이터가 이동되는 연결</p>	<p>출력</p> <p>입력</p> <p>순환 연결</p> <p>순환연결 : 어떠한 뉴론의 출력이 자기 자신에게 입력되는 연결</p>

● 학습방법

① 지도학습(supervised learning)

- 입력벡터와 그 입력을 가했을 경우 기대되는 출력의 쌍을 학습 데이터로 사용
- 입력된 벡터에 대한 출력이 기대되는 출력과 다르면 그 차이를 해소할 수 있도록 연결가중치를 조정

② 자율학습(unsupervised learning)

- 입력벡터만을 학습 데이터로 제공하고, 기대되는 출력은 제시하지 않음
- 입력벡터의 집합을 벡터공간상에서 군집화하여 유사한 벡터는 동일한 출력을 낼 수 있도록 연결가중치를 조정

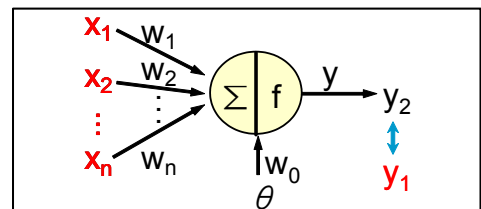
**8. 퍼셉트론**

● 퍼셉트론 모델

- 1962년 로젠블라트(Rosenblatt)가 퍼셉트론의 학습이론을 제시하고, 선형분리가 가능한 입력벡터의 집합들에 대한 선형분리 경계면을 학습할 수 있음을 입증
- 뉴런의 기능
  - 입력벡터가 연결가중치를 통해 입력되면 이를 합하여 활성화함수를 거쳐 출력
  - 활성화함수는 하드 리미트
- 지도학습 : 델타규칙을 사용

● 퍼셉트론의 학습

- 학습 데이터 :  $[(x_1, x_2, \dots, x_n), y_1]$
- 입력벡터에 대한 출력 결과 =  $y_2$



- 실제 출력과 원하는 출력의 차  $\delta$

$$\delta = y_1 - y_2$$

$$\Delta_i = \eta \delta x_i,$$

여기서  $\eta$  : 학습률,  $0 < \eta < 1$

- 모든 가중치를  $\delta$ 와 입력에 따라 조정

$$w_i(n+1) = w_i(n) + \Delta_i$$

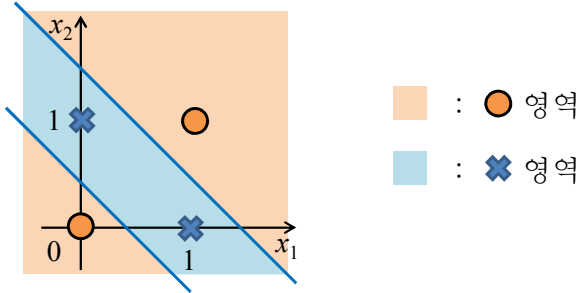
● 퍼셉트론 학습모델의 특징

- 학습 데이터를 이용하여 반복 학습하면 이를 분리할 수 있는 선형 결정경계를 형성
  - 선형분리가 되지 않는 경우에는 해결 불가
- ⇒ 민스키(Minsky) : 단층 퍼셉트론은 XOR와 같은 단순한 함수도 생성할 수 없다고 비난

**9. 오차역전파 모델**

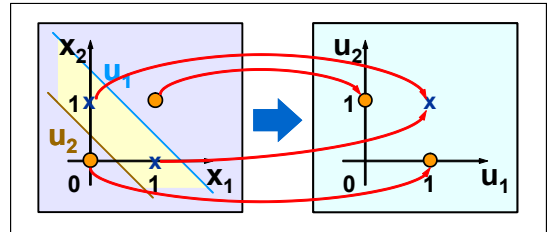
● XOR 문제

□ 1개의 선형함수로는 ○와 ×를 분리해 낼 수 없음

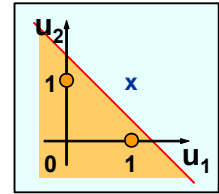


⇒ 2단계로 구현하면 분리가능

① 하나의 분류기의 출력  $u_1$ 은 그 분류기가 정의하는 선형 결정경계의 아래에 존재하면 1, 그렇지 않으면 0이 되도록 하고, 다른 하나의 분류기의 출력  $u_2$ 는 그 분류기가 정의하는 선형 결정경계의 위에 존재하면 1, 그렇지 않으면 0이 되도록 분류함. 이와 같은 방법으로  $x_1$ - $x_2$  공간상의 점들을  $u_1$ - $u_2$  좌표 공간으로 변환

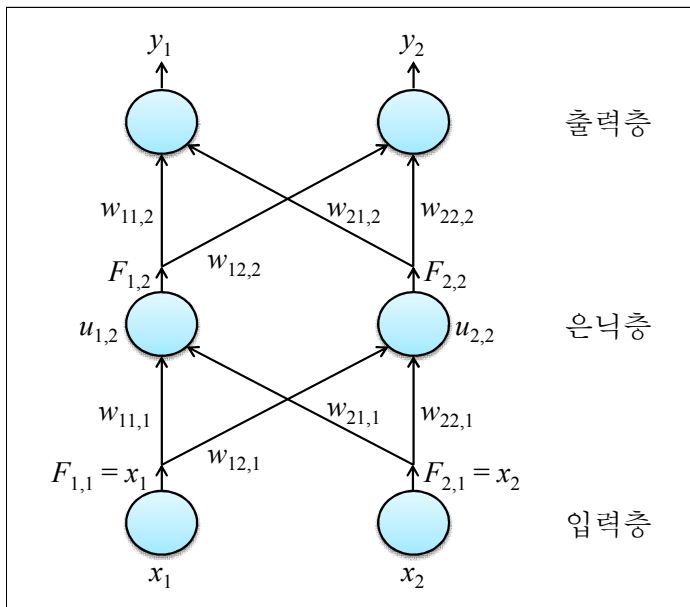


②  $u_1$ - $u_2$  공간에서는 선형함수로 ○와 ×를 분리할 수 있게 되었으므로 선형 결정경계를 구함



● 다층 퍼셉트론 구조

- 여러 단계의 층을 두어 선형분리가 불가능한 문제에 대한 해결책을 모색하고자 하는 구조
- 입력층과 출력층 사이에 은닉층 존재



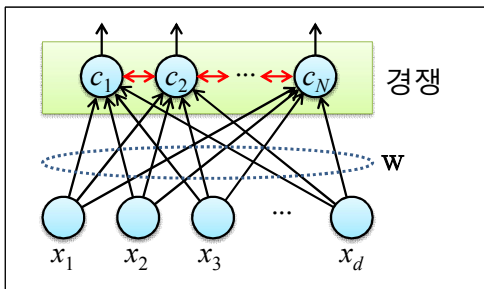
□ 은닉층에 대한 출력을 자동적으로 지정해 줄 수 없어서 은닉층에 연결되는 가중치의 학습을 할 수 있는 모델을 만들어 내는 데 어려움이 있었음

⇒ 오차역전파(error back propagation, BP) 모델이 제안되어 은닉층을 포함하는 다층 신경망 구조의 학습이 가능해짐

**10. 자기조직화 지도 및 LVQ**

● 자기조직화 지도(SOM ; self-organizing map)

- 경쟁학습 모델 : 입력 데이터에 대해 반응을 하기 위한 권한을 다른 노드와 경쟁을 통해 얻어내는 방식으로 학습



- 입력벡터  $\mathbf{x}_j$ 에 따른 연결가중치 갱신 : 승자뉴론  $c_i$ 와 함께  $c_i$ 를 중심으로 정의된 이웃 집합  $N_{c_i}$ 에 대해 다음과 같이 갱신

$$\mathbf{w}_m(n+1) = \begin{cases} \mathbf{w}_m(n) + \alpha(n)\{\mathbf{x}_j - \mathbf{w}_m(n)\} & \text{if } m \in N_{c_i}(n) \\ \mathbf{w}_m(n) & \text{if } m \notin N_{c_i}(n) \end{cases}$$

- 연결가중치  $\mathbf{w}_m(n)$ 을 갱신한  $\mathbf{w}_m(n+1)$ 은  $\mathbf{x}_j$ 에 가까워지는 방향으로  $\alpha(n)$ 의 비율로 이동함
- 이웃 집합  $N(n)$ 의 크기는  $n$ 에 따라 변화하는데, 초기에는 크게 하고, 점차 작아지게 함

● LVQ(learning vector quantization)

- SOM과 유사하나, 지도학습 방식에 따라 대표벡터를 학습
- 학습표본  $\mathbf{x}_j = (x_{j1}, x_{j2}, \dots, x_{jd})$ 가 입력되면 출력뉴론은 경쟁을 통해  $\mathbf{x}_j$ 와 가장 유사한 연결가중치 벡터  $\mathbf{w}_i$ 를 통해 연결된 출력노드  $c_i$ 가 승자노드가 됨
- $c_i$ 가  $\mathbf{x}_j$ 와 같은 클래스에 속하는지 여부에 따라 다음과 같이  $\mathbf{w}_i$ 를 갱신함

$$\mathbf{w}_i(n+1) = \begin{cases} \mathbf{w}_i(n) + \alpha(n)\{\mathbf{x}_j - \mathbf{w}_i(n)\} & \text{if } \mathbf{x}_j \text{의 클래스} = c_i \text{의 클래스} \\ \mathbf{w}_i(n) - \alpha(n)\{\mathbf{x}_j - \mathbf{w}_i(n)\} & \text{if } \mathbf{x}_j \text{의 클래스} \neq c_i \text{의 클래스} \end{cases}$$

- 승자노드가 학습표본과 같은 클래스라면 승자노드에 연결되는 연결가중치를 학습표본에 가까워지도록 갱신하고, 그렇지 않으면 그 반대로 밀어내도록 학습이 이루어짐

## 중간시험 대비문제와 해설

1. ‘지능적 시스템’을 설명하는 다음 문장의 공란에 넣을 적절한 단어는?

지식을 사용하며, (㉠)을(를) 통해 새로운 지식을 습득하거나 알고 있는 지식을 보강한다. 외부의 환경을 인식함으로써 문제를 (㉡)할 수 있고, 지식을 기반으로 (㉢)을(를) 함으로써 대상문제의 해를 찾아낸다.

[풀이] (㉠) 학습, (㉡) 이해, (㉢) 추론

2. 물리적 기호 시스템 가설이란 무엇인가?

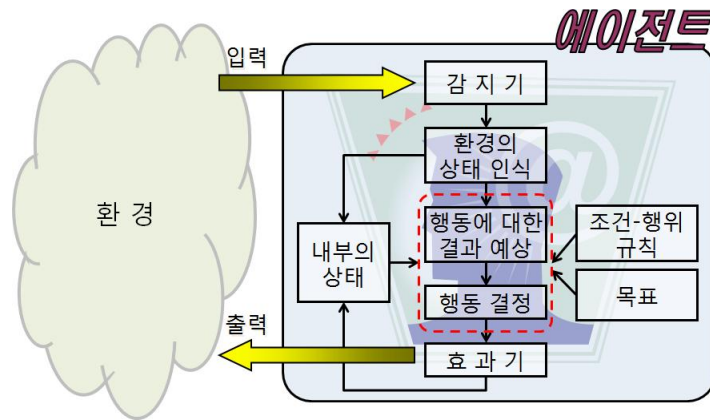
[풀이] 인간의 지능은 두뇌 속에 형성된 개념을 기본으로 하는데, 이 개념을 기호(심벌)로 표현할 수 있으며, 이렇게 표현된 기호를 처리하는 기호처리 시스템이 일반적인 지적 행동을 하는 데 필요충분한 수단이라는 가설이다. 이러한 가설은 본질적으로 기호를 처리하는 장치인 컴퓨터를 통해 프로그램을 작성함으로써 인간이 행하는 지능적 작업을 수행할 수 있다는 가능성을 제시함으로써, 인공지능 연구의 기본적인 근거가 된 가설이다.

3. 지능형 에이전트의 일반적인 특성을 3개 이상 나열하고 설명하라.

[풀이] 지능형 에이전트가 갖는 공통적인 속성으로는 자율성, 반응성, 능동성, 지속성 등이 있다. 자율성은 에이전트가 사용자를 위해 자율적으로 행동할 수 있음을 나타낸다. 즉, 사용자에게 일일이 허가받지 않고 사용자를 위해 스스로 행동할 수 있다는 의미이다. 반응성은 에이전트가 환경의 변화에 대해 시간적으로 적절하게, 즉, 시스템이 요구하는 시간 내에 응답한다는 의미이다. 능동성은 에이전트가 단순히 주어진 환경에 반응하여 행동하는 것이 아니라 주도권을 가지고 목표지향적으로 행동한다는 의미이다. 지속성은 에이전트가 주어진 작업을 1회성으로 실행한 후 종료하는 것이 아니라, 데몬과 같이 지속적으로 실행하면서 임무에 맞는 상황에서 자동적으로 동작하는 프로세스라는 의미이다.

4. 목표기반 에이전트 시스템이란 무엇인가?

[풀이] 목표기반 에이전트에서는 에이전트의 동작이 주어진 환경에서 감지된 입력값에만 의존하여 반응하는 것이 아니라, 부여된 목표를 달성하기 위한 행동을 선택하여 동작한다. 이를 위해서 에이전트는 내부에 자신이 놓여 있는 환경과 추구하는 목표, 가능한 행동들에 대한 명시적인 기호 모델을 가지고 있으며, 이 모델을 바탕으로 논리적 추론을 전개함으로써 가능한 행동을 찾아낸다. 그리고 그러한 행동을 했을 때의 결과를 내부 모델을 바탕으로 시뮬레이션을 통해 예상하고, 그중 목표를 달성하는 데 적합한 것으로 판단되는 행동을 결정하여 동작한다.



5. 경험적 문제풀이 방식이란 무엇인가?

[풀이] 문제의 풀이방법이 정립되지 못한 경우 주어진 문제를 풀이기 위한 직접적인 알고리즘을 구현할 수 없다. 이러한 경우 시행착오나 경험 등을 통해 얻은 지식을 사용하여 문제를 풀이할 수 있는데, 이러한 지식은 완전하지는 않지만, 많은 경우 문제풀이에 적용될 수 있는 지식이다. 이러한 지식을 경험적 지식이라 하며, 이러한 경험적 지식에 의존한 문제풀이 방식을 경험적 문제풀이 방식이라고 한다.

6. 문제축소에 의한 문제풀이 방식이란 무엇인가?

[풀이] 주어진 문제를 직접 풀이기 어려울 경우, 이를 몇 개의 부분문제로 분해한다. 이러한 문제의 분해과정은 각각의 부분문제에 대해 반복적으로 실시된다. 이를 문제축소(problem reduction)라고 한다. 이때 분해된 부분문제가 아주 쉽게 풀이될 수 있는(해가 알려진) 문제가 되었다면 그 부분문제는 풀이된 것이다. 만일 분해한 부분문제가 모두 풀이되면 그 해들의 집합으로 문제 전체의 해를 구할 수 있다. 이와 같은 문제풀이 과정을 문제축소에 의한 문제풀이 방식이라고 한다.

7. 상태, 연산자, 상태공간에 대하여 설명하라.

[풀이] 상태(state)란 문제풀이 과정 중 어느 한 지점에서의 문제의 형태를 나타낸다. 풀이해야 할 문제의 상태는 초기상태라고 하며, 풀이된 결과에 해당되는 상태를 목표상태라고 한다. 문제의 풀이과정은 초기상태를 정해진 방법에 따라 변화시켜 목표상태가 되도록 하는 변화수순을 찾는 것이라 할 수 있는데, 이처럼 주어진 상태를 변화시키는 방법의 집합을 연산자라고 한다. 초기상태에 대해 적용할 수 있는 모든 연산자를 적용함으로써 새로운 상태들의 집합이 만들어지며, 이들 상태에 대해 또다시 적용가능한 연산자를 모두 적용하면 이에 해당되는 후계상태가 만들어진다. 이와 같이 연산자들을 반복 적용하면 상태들이 서로 그래프 형태로 연결된 모든 상태의 집합이 만들어지는데, 이를 상태공간이라고 한다.



8. 상태묘사란 무엇인가?

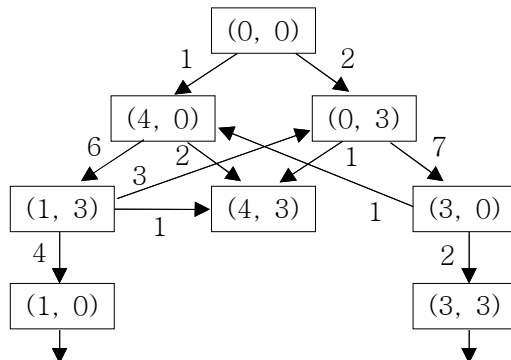
[풀이] 상태공간에서 문제를 풀이하고자 한다면 문제의 상태를 컴퓨터상에 표현할 수 있어야 한다. 문제풀이 과정 중 어느 한 시점에서 문제의 형태를 나타내는 상태는 대상문제를 적절히 표현할 수 있는 자료구조를 통해 컴퓨터 내에 표현될 수 있는데, 이를 상태묘사(state description)라고 한다. 상태묘사는 벡터, 트리, 리스트 등 다양한 자료구조 중에서 해당 문제의 상태를 적절히 표현할 수 있는 방법을 선택하게 된다.

9. 4리터와 3리터들이 물병을 사용하는 물병문제에 대해 다음과 같이 연산자가 정의되어 있을 때 상태공간의 일부를 보여라. 단, 초기상태는 두 물병이 모두 비어 있는 상태이다.

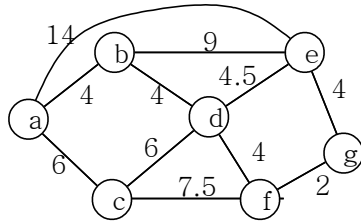
번호	연산자
1	4리터들이 물병을 가득 채운다.
2	3리터들이 물병을 가득 채운다.
3	4리터들이 물병을 비운다.
4	3리터들이 물병을 비운다.
5	4리터들이 물병이 찰 때까지 3리터들이 물병의 물을 옮긴다.
6	3리터들이 물병이 찰 때까지 4리터들이 물병의 물을 옮긴다.
7	3리터들이 물병의 물을 4리터들이 물병으로 모두 옮긴다.
8	4리터들이 물병의 물을 3리터들이 물병으로 모두 옮긴다.

[풀이] 우선 상태묘사 방법을 정한다. 주어진 문제의 경우 4리터들이 물병과 3리터들이 물병에 든 물의 양을 각각  $x$ 와  $y$ 라고 할 때,  $(x, y)$  형태의 벡터로 표현할 수 있다. 이러한 표현방법을 이용할 경우 초기상태는  $(0, 0)$ 으로 표현된다.

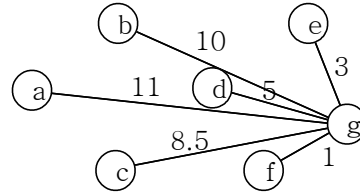
초기상태에  $(0, 0)$ 에 대해 적용할 수 있는 연산자는 1과 2이며, 1을 적용할 경우 후계상태는  $(4, 0)$ , 2를 적용할 경우 후계상태는  $(0, 3)$ 이 된다. 상태  $(4, 0)$ 에 대해 적용할 수 있는 연산자는 2와 3이며, 각각의 경우  $(4, 3)$ ,  $(1, 3)$ 이 후계상태가 된다. 연산자 3을 적용할 수도 있으나, 이는 부모상태인  $(0, 0)$ 으로 환원되는 것이므로 제거하도록 한다. 이와 같은 과정을 반복하여 적용한다면 다음과 같은 그래프 형태의 상태공간을 얻을 수 있다.



10. 다음 <그림 1>은 7개의 도시와 이들을 연결하는 도로망을 나타낸 것이다. 간선에 표시된 값은 도시 사이의 도로 거리이다. 또한 <그림 2>는 각 도시로부터 목적지인 도시 g까지의 직선거리를 나타낸다. 도시 a에서 출발하여 도시 g로 가는 경로를 탐색하고자 한다.



<그림 1> 도시 사이의 도로 거리



<그림 2> 도시 g까지의 직선거리

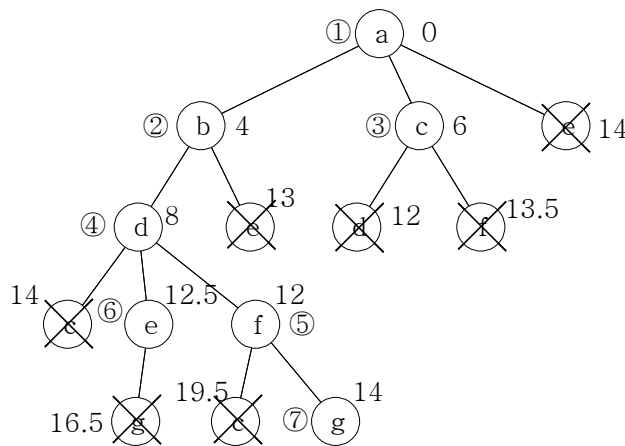
(가) 균일비용 탐색을 적용할 경우와 A\* 알고리즘을 적용할 경우 평가함수는 각각 어떻게 정의되며, 이 문제에서 어떻게 적용될 수 있는가?

(나) 균일비용 탐색과 A\* 알고리즘에 의한 탐색트리를 각각 보여라. 각 노드에는 그 노드의 평가함수 값을 표시하고, 노드가 확장되는 순서를 ①, ②, ...로 표시하라.

[풀이] (가) 출발노드로부터 탐색을 진행하여 특정 노드  $n$ 까지 도달했다고 하자. 이때  $g(n)$ 을 출발 노드로부터 노드  $n$ 까지 도달하는 데 소비된 경로비용이라 한다면, 균일비용 탐색에서는  $g(n)$ 이 노드  $n$ 의 평가함수가 된다. 반면 A\* 알고리즘에서는 목표노드까지 도달하기 위해 소비할 것으로 예상되는 비용을 포함한다. 즉, 이 예상비용을  $\hat{h}(n)$ 이라고 한다면 A\* 알고리즘의 평가함수  $\hat{f}(n)$ 은  $g(n) + \hat{h}(n)$ 이다.

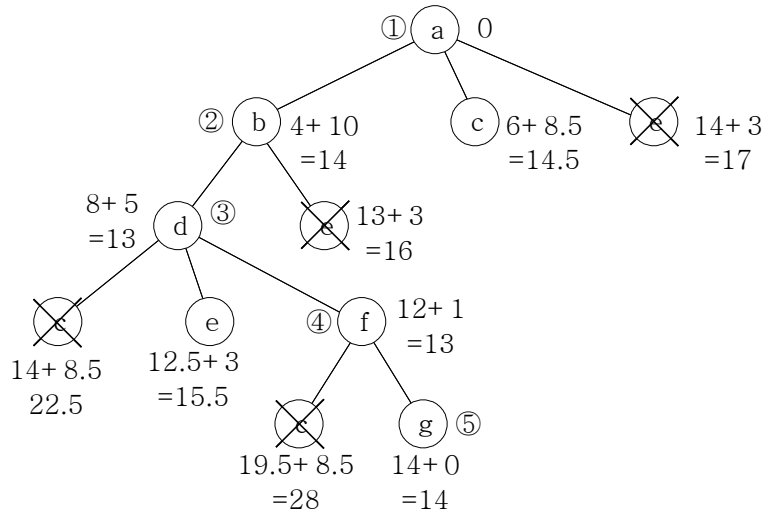
이 문제에서  $g(n)$ 은 출발 도시 a로부터 도시  $n$ 까지 도달하는 동안 거친 도로의 거리를 합산한 것으로 정의할 수 있으며,  $\hat{h}(n)$ 은 도시  $n$ 으로부터 목적지인 도시 g까지의 직선거리로 정의할 수 있다.

(나) 균일비용 탐색에서는  $g(n)$ 이 최소인 노드를 선택하여 확장한다. 다음 그림은 균일비용 탐색의 탐색트리이다. 예를 들어 ④번째로 확장되는 노드 d의 경우 경로비용  $g(d)$ 는  $4+4=8$ 이며, 이 노드의 후계노드인 e의 경로비용은  $g(d)$ 에 d와 e 사이의 거리 4.5를 더한 12.5가 된다.



A\* 알고리즘에서는  $\hat{f}(n)$ 이 최소인 노드를 선택하여 확장한다. 다음 그림은 A\* 알고리즘의 탐색트리이다. 예를 들어 ③번째로 확장되는 노드 d의 경우 노드 d까지 도달하는 경로비용  $g(d)$ 와 노드

$d$ 로부터 목표노드까지의 예측 경로비용  $\hat{h}(d)$ 의 합이 평가함수 값이다.  $g(d)$ 는 균일비용 탐색에서와 같이 8이며,  $\hat{h}(d)$ 는 도시  $d$ 로부터  $g$ 까지의 직선거리인 5이다. 그러므로  $\hat{f}(d)=8+5=13$ 이다.

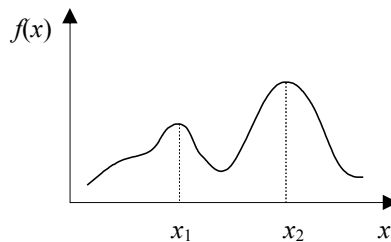


11. 지역최대치 문제란 무엇인가?

[풀이] 탐색의 목표가 정해진 평가값(수익, 성능 등)을 최대화하기 위해 시스템의 계수를 조정하는 문제에서 풀이에 실패하게 되는 상황 중 한 가지이다. 이러한 계수최적화 문제의 일반적인 풀이방법은 현재 상태에서 시스템의 평가값이 가장 많이 개선되는 인접한 값으로 정해진 범위 내에서 시스템의 계수를 조정하는 것이다. 그런데 이러한 문제에서 어떠한 평가값이 목표값이라고 절대적으로 정하기 어려운 경우가 많다. 대신 다음과 같이 목표상태를 정의하게 된다.

만일 조정된 계수에 의해 목표상태에 도달했다면 그 상태에서는 시스템의 평가값이 최대일 것이며, 따라서 시스템의 계수를 어떻게 조정하더라도 평가값은 개선되지 않을 것이다. 그러므로 이러한 시스템에서는 시스템의 계수를 조정하여 개선된 평가값을 얻지 못하는 상태를 목표상태로 정의할 수 있다.

그러나 탐색을 진행하는 과정에서 목표상태는 아니지만 아래 그림에서와 같이 지역적 극댓값( $x_1$  위치)에 도달하게 되면 마치 목표상태인 것으로 판단하여 탐색을 멈추게 되어 실제 목표상태( $x_2$  위치)를 놓치게 된다. 이러한 문제를 지역최대치 문제라고 한다.



12. 모의 담금질의 개념을 설명하라.

[풀이] 모의 담금질은 경사하강법으로 최솟값을 찾고자 할 때 지역최소치에 빠지는 문제(역으로는 지역최대치에 빠지는 문제)를 해결하기 위한 기법 중 하나이다. 금속이나 유리를 높은 온도로 가열하면 원자들이 현재위치(내부 에너지의 지역최소치)에서 벗어나 무작위로 이리 저리 움직이며, 이를 천천히 식히면 초기위치에 비해 더 낮은 내부 에너지 상태를 찾아 위치할 가능성을 높임으로써 결정의 크기를 크게 하고 결함을 줄이는 기술을 담금질이라고 한다. 모의 담금질은 이러한 담금질의 개념을 시뮬레이션하여 지역최소치 또는 지역최대치에서 빠져나와 원하는 최솟값 또는 최댓값을 얻는 알고리즘이다.

후계상태로 이동하는 각 단계에서 임의 후계노드를 선택하여 그 노드가 개선된 상태를 나타내는 경우 그 후계노드를 차기상태로 선택한다. 만일 선택된 후계노드가 상태를 개선하지 못한다면 그 후계노드를 선택할 수도 있고 선택하지 않을 수도 있는데, 이 선택은  $e^{\Delta E/T}$ 를 확률로 하여 이루어진다.  $\Delta E$ 는 차기상태 후보의 에너지와 현재상태의 에너지의 차이이며,  $T$ 는 시간에 따라 서서히 감소하는 온도를 나타낸다. 동일한  $\Delta E$ 에 대해서 처음에는 확률값이 커서 더 안 좋은 상태로 움직일 가능성도 높다. 이것은 지역최소치에 빠져 있는 상태로부터 무작위로 빠져 나올 수 있는 불안정한 형국이다. 시간이 흘러  $T$ 의 값이 작아지면 확률은 0의 값으로 접근하여 상태를 개선하지 않는 경우 차기상태로 선택될 가능성이 점점 작아져 안정된다.

13. A\* 알고리즘의 평가함수에 대하여 설명하고, 평가함수에 따른 최적경로 탐색 가능 여부에 대하여 설명하라.

[풀이] 탐색과정 중 임의의 노드  $n$ 을 거쳐 목표노드에 도달하는 데 필요한 비용은 출발노드로부터 노드  $n$ 까지 도달하는 데 소비한 경로비용  $g(n)$ 과 노드  $n$ 으로부터 목표노드까지 도달하는 데 필요한 비용  $h(n)$ 의 합이 된다. 이때 탐색과정 중 노드  $n$ 에 도달했을 때  $g(n)$ 은 이미 거쳐 온 경로의 비용을 합산하면 계산할 수 있으나,  $h(n)$ 은 아직 탐색하지 않은 경로의 비용이므로 실제 비용을 알 수 없다. 경험적 탐색 알고리즘인 A\* 알고리즘에서는 경험적 지식을 반영하여  $h(n)$ 의 예측치인  $\hat{h}(n)$ 을 대신 사용한다. 즉, A\* 알고리즘에서 평가함수  $\hat{f}(n)$ 은 다음과 같다.

$$\hat{f}(n) = g(n) + \hat{h}(n)$$

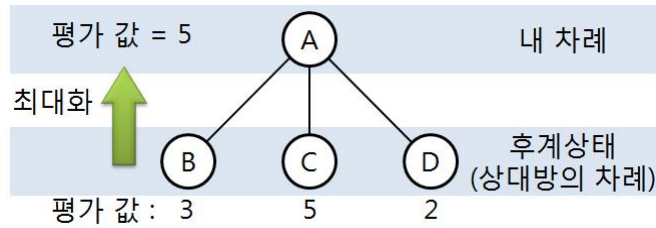
이때 어떤 노드로부터 목표노드까지의 경로비용을 예측한 값이 항상 실제 비용보다 작다면, 즉  $\hat{h}(n) \leq h(n)$ 이 항상 만족된다면 A\* 알고리즘은 최소비용 경로를 탐색하는 것을 보장한다.

14. 바둑이나 서양장기 등과 같이 상대방이 있는 게임을 진행할 때 다음 둘 수를 결정하기 위한 탐색트리에 대하여 설명하라.

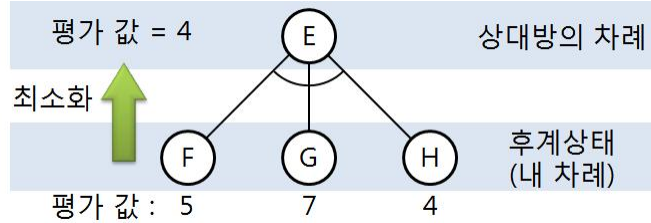
[풀이] 상대방과 대결하는 게임의 경우 나는 내가 둘 수 있는 다음 수 중 내게 가장 유리한 수를 선택하려 할 것이고, 상대방은 내게 가장 불리한 수를 선택하려고 할 것이다. 이러한 관계는 최대최소 트리로 표현되고, 수의 선택은 최대최소 트리를 탐색하여 정할 수 있다.

우선 내게 가장 유리한 수를 선택하는 것은 후계상태 중에서 내게 가장 유리한 상태를 만드는 상태를 선택하는 것이다. 이것은 후계상태 중 노드의 바람직한 정도('나'를 기준으로)를 나타내는

평가값이 가장 큰 것을 선택하는 것이며 이는 최대화 단계에 해당된다. 예를 들어 아래의 그림에서 'A'에서는 평가값이 가장 큰 'C'를 선택한다.

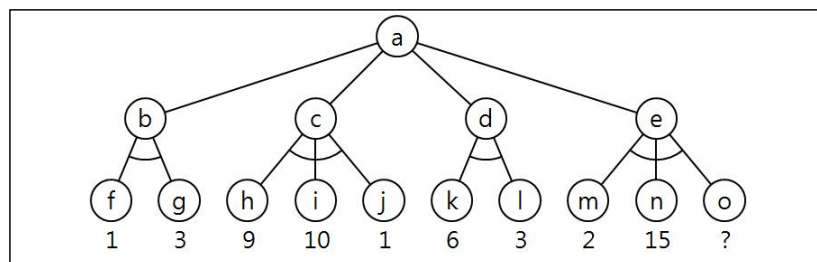


상대방이 둘 차례를 나타내는 상태에서 상대방이 내게 가장 불리한 수를 선택하는 것은 후계상태 중에서 내게 가장 불리한 상태를 선택하는 것이다. 이것은 후계상태 중 노드의 바람직한 정도 ('나'를 기준으로)를 나타내는 평가값이 가장 작은 것을 선택하는 것이며 이는 최소화 단계에 해당된다. 예를 들어 아래의 그림에서 'E'에서는 평가값이 가장 작은 'H'가 선택된다.



탐색트리는 최대화 단계와 최소화 단계가 교대로 일어나는 형태로 구성되며, 이를 최대최소 탐색트리라고 한다.

15. 게임을 위한 최대최소 탐색트리가 다음과 같다고 하자. 리프노드의 평가값은 클수록 유리하다. a 상태에서 다음 수를 선택하는 과정에 대하여 설명하라.



[풀이] 최소화 단계의 노드인 b, c, d, e의 평가값은 자식노드 평가값의 최솟값을 택한다. 따라서 노드 b와 c의 평가값은 1, 노드 d의 평가값은 3이다. e의 평가값은 m, n, o의 평가값의 최솟값이다. 아직 노드 o의 값은 결정하지 않았지만, e의 평가값은 m의 평가값인 2보다 클 수 없다. 노드 a에서는 평가값이 최대인 자식노드를 선택하여 수를 둔다. 따라서 노드 d가 선택된다.

## 기말시험 대비문제와 해설

1. 지식기반 시스템에서 주어진 문제의 상황에 따라 지식을 이용하여 결론을 이끌어 내는 역할을 담당하는 것은?

- ① 지식 베이스
- ② 사용자 인터페이스
- ③ 지식표현
- ④ 추론기관

[정답] ④

[해설] 지식기반 시스템에서 지식 베이스는 적용하고자 하는 분야의 지식을 포함한 모든 문제 풀이에 필요한 지식을 저장하고 있으며, 추론기관은 지식 베이스의 지식을 이용하여 주어진 문제의 상황에 맞는 결론을 추론하는 역할을 한다.

2. 지식기반 시스템의 주요 구성요소에 해당되는 것은?

- ① 지식 베이스와 추론기관
- ② 지식공학자와 DB
- ③ 추론기관과 입력장치
- ④ 입력장치와 출력장치

[정답] ①

[해설] 지식기반 시스템은 적용하고자 하는 분야의 지식을 포함한 모든 문제풀이에 필요한 지식을 저장하는 지식 베이스, 주어진 문제의 상황에 대하여 지식 베이스 내의 지식을 이용한 추론을 함으로써 결론을 제시하는 추론기관의 두 가지 핵심요소와 함께 사용자와 지식기반 시스템 사이의 연결을 담당하는 사용자 인터페이스 등으로 구성된다.

3. 다음 중 지식의 사용에 관한 지식을 나타내는 것은?

- ① 문제영역 지식
- ② 사실
- ③ 메타지식
- ④ 경험적 지식

[정답] ③

[해설] 어떤 규칙이 다른 규칙보다 우선적으로 적용해야 할 조건을 제시하거나, 특수한 문제풀이에 유용한 지식의 적용순서를 제시하거나, 추론에 대한 설명을 통해 결과에 대한 신뢰감을 제공하는 등 지식의 사용에 관한 지식을 메타지식(meta-knowledge)이라고 한다.

4. 선언적 지식표현 방법은 무엇인가?

- ① 상호 독립적이고 단편적인 지식을 모아놓는 방식이다.
- ② 사용에 대한 제어정보가 지식 자체에 내포된 지식이다.
- ③ 지식의 사용에 관한 지식이다.
- ④ 문제 분야 이외의 지식을 집약하는 표현방법이다.

[정답] ①



- ① (㉠), (㉡)
- ② (㉢), (㉣)
- ③ (㉠), (㉢)
- ④ (㉡), (㉣)

[정답] ④

[해설] 선언적 지식은 상호 독립적이고 단편적인 지식을 나열해 놓은 형태로, 지식의 적용을 위한 제어정보는 지식 자체에 표현되지 않으며, 별도의 추론기관에 의해 추론에 활용된다.

8. 조건과 그 조건을 만족할 경우 내리게 되는 결론, 취할 행동 등을 기술하는 형태의 지식표현 방법은?
- ① 규칙
  - ② 시맨틱 네트
  - ③ 프레임
  - ④ LISP 프로그램

[정답] ①

[해설] 규칙은 'IF 조건부 THEN 결론부' 형식의 지식표현 방법으로, 현재의 상태에 의하여 조건부가 만족되는 규칙을 찾아 결론부의 행동을 실행한다.

9. 프레임을 이용한 지식표현에 대한 올바른 설명은?
- ① 특성상속을 사용한다.
  - ② 절차적 지식만을 포함한다.
  - ③ 절차적 지식을 포함시킬 수 없다.
  - ④ 형식논리를 이용한 지식표현 방법이다.

[정답] ①

[해설] 프레임은 isa 슬롯을 이용하여 계층적 개념관계를 표현할 수 있으며, 이를 통해 일반적 개념의 속성을 하위 개념의 프레임이 이어받을 수 있는 특성상속을 사용한다.

10. 다음 중 부가 프로시저를 통해 절차적 지식을 표현할 수 있는 지식표현 방법은?
- ① 프레임
  - ② 술어논리
  - ③ 규칙
  - ④ 시맨틱 네트

[정답] ①

[해설] 부가 프로시저는 프레임에서 어떠한 슬롯에 값을 넣거나 읽거나 지우는 등 해당 슬롯을 사용했을 때, 그 사용과 관련되어 수행해야 할 동작을 지시하는 프로시저로, 프레임에서 절차적 지식을 활용할 수 있게 한다.

11. 전문가 시스템에 대한 올바른 설명은?

- ① 일반적 상황에 대한 지능적 판단과 행동을 구현한 것이다.



- ② 인간 전문가에 비해 창조적인 해법을 제시한다.
- ③ 특정문제 분야에 적용하기 위한 지식기반 시스템이다.
- ④ 상식적 판단이 필요한 분야에 적합하다.

[정답] ③

[해설] 전문가 시스템은 특정응용 분야에 정통한 전문가로부터 유용한 지식, 문제해결의 전략 및 규칙 등을 도출하여 지식 베이스화함으로써 그 분야의 문제해결 및 조언 등을 할 수 있도록 설계된 시스템이다.

12. 다음 중 현장전문가와 비교했을 때 전문가 시스템이 갖는 장점에 해당되는 것은?

- ① 창조적 문제해결 능력이 뛰어나다.
- ② 거시적 판단능력이 뛰어나다.
- ③ 문제풀이 과정이 개인의 상태에 관계없이 일관적이다.
- ④ 상식적인 지식을 잘 활용할 수 있다.

[정답] ③

[해설] 사람은 신체적 상황 등에 따라 업무실행 능력이 다를 수 있으나, 전문가 시스템은 문제풀이 과정이 항상 일관적인 특성을 갖는다.

13. 전문가 시스템을 구축하는 과정에서 지식공학자의 역할은 무엇인가?

- ① 문제 분야의 지식을 제공한다.
- ② 문제풀이에 도움이 되는 경험적 규칙을 제공한다.
- ③ 문제의 특정상황에 해당되는 해결방법을 제시한다.
- ④ 설계된 지식표현 방법에 따라 전문가의 지식을 표현한다.

[정답] ④

[해설] ①~③은 현장전문가의 역할이고, 지식공학자는 이러한 현장전문가의 전문지식과 문제풀이를 위한 경험적 규칙을 체계적으로 지식 베이스화하는 역할을 한다.

14. 전문가 시스템을 구성하는 핵심요소는?

- ① 지식 베이스와 지식편집기
- ② 추론기관과 지식편집기
- ③ 지식편집기와 사용자 인터페이스
- ④ 지식 베이스와 추론기관

[정답] ④

[해설] 한정된 문제 분야에 적용하기 위해 그 분야의 전문지식을 축적한 지식 베이스를 구축하여 만든 지식기반 시스템이다. 지식기반 시스템은 문제영역의 지식과 추론기관이 핵심적 구성요소이

다.

15. 전문가 시스템의 추론기관에 대한 **잘못된** 설명은?

- ① 생성 시스템의 인터프리터에 해당된다.
- ② 규칙해석기는 현재의 사실을 바탕으로 적용할 규칙을 찾는다.
- ③ 스케줄러는 규칙이 적용되는 순서를 결정한다.
- ④ 문제 분야의 전문지식이 추론기관 내에 결합되어 있다.

[정답] ④

[해설] 전문가 시스템은 지식기반 시스템이다. 지식기반 시스템은 문제영역의 지식을 저장하는 지식 베이스와 이를 이용한 추론을 담당하는 추론기관이 분리되어 있다.

16. 전문가 시스템 개발도구에 대한 설명이 올바른 것은?

- ① 문제지향언어 : 추론기관을 내장하고 있다.
- ② 기호처리언어 : 과학 및 상업 분야의 수치계산에 쓰인다.
- ③ 골격 시스템 : 기존 전문가 시스템에서 지식 베이스를 제거한 것이다.
- ④ 범용 지식공학 언어 : C와 같은 프로그래밍 언어이다.

[정답] ③

[해설] 전문가 시스템을 구성하는 요소 중 문제 분야의 지식 베이스를 제외한 나머지 요소로 구성된 전문가 시스템 구성도구를 골격 시스템이라고 한다. 지식 베이스만 구축함으로써 손쉽게 전문가 시스템을 구성할 수 있다는 것이 장점이다.

17. 기호논리에 대한 올바른 설명은?

- ① 인식의 본질이나 과정을 표현하는 것이다.
- ② 불대수라는 수학적 도구를 사용한다.
- ③ 참인 동시에 거짓인 명제가 존재한다.
- ④ 기호가 내포하는 의미를 해석하여 추론을 한다.

[정답] ②

[해설] 기호논리는 기호로 표현된 명제와 AND, OR, NOT 등의 논리연산자를 이용한 불대수를 도구로 사용한 연역을 할 수 있게 한다. 인식론적 논리학이 인식의 본질이나 과정을 연구하는 것과 달리 기호논리학에서는 기호가 내포하는 의미 자체보다는 형식적 논리 연산 및 추론에 관심을 가지고 있다.

18. 다음 중 연언표준형에 해당되는 논리식은?

- ①  $(P \wedge Q) \vee (R \wedge S)$
- ②  $(\sim P \vee Q) \wedge (R \vee S \vee \sim Q)$

- ③  $(\sim P \vee Q) \wedge (R \vee (S \wedge U))$                       ④  $\sim (P \vee Q) \wedge (P \vee \sim R)$

[정답] ②

[해설] 연언표준형은 리터럴의 논리합으로 연결된 절을 논리곱으로 연결한 형식의 표준형 논리식이다. ①은 선언표준형이다.

19. 다음 중 항상 참인 추론은?

- ①  $a$ 가 참이고  $a \rightarrow b$ 이면  $b$ 가 참이다.
- ②  $a$ 가 거짓이고  $a \rightarrow b$ 이면  $b$ 가 거짓이다.
- ③  $b$ 가 참이고  $a \rightarrow b$ 이면  $a$ 가 참이다.
- ④  $b$ 가 거짓이고  $a \rightarrow b$ 이면  $a$ 가 참이다.

[정답] ①

[해설] 조건명제  $a \rightarrow b$ 로부터 항상 참인 추론을 할 수 있는 것은  $a$ 가 참일 때  $b$ 도 참임을 추론하는 긍정식(modus ponens)과  $b$ 가 거짓일 때  $a$ 도 거짓임을 추론하는 부정식(modus tolens)이다.

20.  $P \rightarrow Q$ 가 참일 때, 긍정논법에 의한 올바른 추론은?

- ①  $P$ 가 참이면  $Q$ 도 참이다.
- ②  $P$ 가 거짓이면  $Q$ 도 거짓이다.
- ③  $Q$ 가 참이면  $P$ 도 참이다.
- ④  $P$ 가 거짓이면  $Q$ 는 참이다.

[정답] ①

[해설] 긍정논법은  $P$ 라는 사실과  $P \rightarrow Q$ 라는 명제로부터  $Q$ 라는 사실을 추론한다.

21. 다음 중 부모절과 도출절의 쌍이 올바른 것은?

- ① 부모절 :  $p, \sim p \vee q$                       도출절 :  $q$
- ② 부모절 :  $p \vee q, \sim p \vee q$                       도출절 :  $\sim p$
- ③ 부모절 :  $p, \sim p$                               도출절 :  $true$
- ④ 부모절 :  $p \wedge q, \sim q \wedge r$                       도출절 :  $p \wedge r$

[정답] ④

[해설] 리터럴들의 논리합만으로 표현된 두 절에 동일 리터럴의 긍정과 부정이 동시에 존재할 때, 이들을 제거한 나머지 논리식의 논리합에 해당되는 도출절을 얻는다. ③에서  $p$ 는  $p \vee false$ ,  $\sim p$ 는  $\sim p \vee false$ 와 동치이므로, 이 두 절에서 각각  $p$ 와  $\sim p$ 를 제거하고 난 나머지 절의 논리합  $false \vee false$ , 즉,  $false$ 가 도출된다.





④ 규칙기반 시스템에서는 주로 귀납법에 의한 추론을 한다.

[정답] ①

[해설] 연역법은 'A'와 'if A then B'로부터 'B'를 추론하는 것으로, 이에 의한 추론은 항상 올바른 결론을 이끌어 낸다. 따라서 규칙기반 시스템에서 연역법을 일반적인 추론방법으로 사용한다. 귀납법은 공통적인 내용을 표현하는 사실들로부터 일반적 규칙을 찾아내는 추론방법으로, 항상 참인 규칙만을 만들어 내지는 못한다.

※ (29~30) 다음과 같은 규칙과 사실에 대한 추론과정에 대한 질문에 답하라.

규칙	(㉠) $A \wedge B \rightarrow C$ (㉡) $C \rightarrow E$ (㉢) $D \rightarrow B$ (㉣) $A \wedge E \rightarrow H$ (㉤) $E \wedge F \rightarrow G$	사실	A D F
----	--	----	-------------

29. 전방향 추론을 할 경우 어떠한 규칙이 제일 먼저 실행되는가?

- ① (㉠)
- ② (㉡)
- ③ (㉢)
- ④ (㉣)

[정답] ③

[해설] 전방향 추론은 조건부가 만족된 규칙을 선택하여 결론부에 해당되는 내용을 추론결과로 제시한다. (㉠)~(㉤) 중 현재 사실을 바탕으로 하여 조건이 만족되는 것은 (㉢)이다.

30. G가 사실인지 입증하고자 한다. 이때 효과적인 추론방향과 그 경우 처음 선택되는 규칙은?

- ① 전방향 추론, (㉠)
- ② 전방향 추론, (㉡)
- ③ 후방향 추론, (㉢)
- ④ 후방향 추론, (㉤)

[정답] ④

[해설] 어떠한 사실을 입증하고자 한다면 그 사실을 결론부에서 제시하는 규칙을 찾아, 그 규칙의 조건부가 만족되는지를 검증해 나가는 추론방법을 사용할 수 있다. 이러한 추론방법을 후방향 추론이라고 한다. 현재 입증하고자 하는 사실이 G이므로, 이를 결론부에 가지고 있는 규칙 (㉤)이 처음 선택된다.

31. 다음 중 학습과 관계가 깊은 추론방법은?

- ① 연역법
- ② 유도법
- ③ 귀납법
- ④ 유사추론

[정답] ③







39. 퍼지집합 A와 B의 교집합은?

$$A = \{(a, 1.0), (b, 0.7), (c, 0.2)\}$$

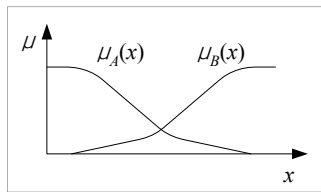
$$B = \{(b, 0.5), (c, 0.4), (d, 0.1)\}$$

- ①  $A \cap B = \{(a, 1.0), (b, 0.7), (c, 0.4), (d, 0.1)\}$
- ②  $A \cap B = \{(a, 1.0), (b, 0.5), (c, 0.2), (d, 0.1)\}$
- ③  $A \cap B = \{(b, 0.5), (c, 0.2)\}$
- ④  $A \cap B = \{(b, 0.7), (c, 0.4)\}$

[정답] ③

[해설] 퍼지집합의 교집합은 소속함수값의 최소값을 취한다. 원소  $a$ 와  $d$ 의 경우는 각각 집합  $B$ 와  $A$ 에 나열되어 있지 않으므로 소속함수값이 0인 것으로 계산해야 한다.

40. 퍼지집합 A와 B의 소속함수가 다음과 같을 때  $A \cap B$ 의 소속함수는?



- ①
- ②
- ③
- ④

[정답] ①

[해설] 두 집합의 소속함수 그래프의 작은 값에 해당되는 것을 구하면 된다.

41. 퍼지논리의 연산자를 적절히 정의한 것은?

- ①  $\bar{a} = 0.5a$
- ②  $a \wedge b = a - b$
- ③  $a \vee b = \max(a, b)$
- ④  $a \rightarrow b = a + b$

[정답] ③

[해설] 퍼지논리에서 논리합은 두 논리값의 최댓값으로, 논리곱은 최솟값으로 정할 수 있다. 부정

의 논리값은  $\bar{a} = 1 - a$ 이다. 조건명제  $a \rightarrow b$ 는  $\min(1, 1 - a + b)$ 로 정의할 수 있다.

42. 퍼지추론 및 제어기에 대한 올바른 설명을 모두 나열한 것은?

- (㉠) 조건 및 결론부에 언어적 변수를 포함한다.
- (㉡) 일반규칙과 동일한 추론절차를 사용한다.
- (㉢) 추론된 결과를 비퍼지화하여 제어대상을 제어한다.
- (㉣) 규칙의 조건에 대한 소속함수값은 0 또는 1이다.

- ① (㉠), (㉡)
- ② (㉠), (㉣)
- ③ (㉡), (㉣)
- ④ (㉡), (㉢)

[정답] ②

[해설] 퍼지추론은 언어적 형태의 규칙을 사용하며, 조건의 정도를 나타내는 수식어를 통해 융통성 있게 규칙을 적용할 수 있도록 한다. 추론의 결과는 퍼지 소속함수의 형태로 제시되며, 이를 통해 실제 제어를 하려면 특정한 값으로 변환하는 비퍼지화를 하여 사용한다.

43. 디지털 영상의 입력에 대한 올바른 설명은?

- ① 표본화는 입력된 데이터를 정수값으로 표현하는 과정이다.
- ② 양자화는 원래의 값을 그대로 유지한다.
- ③ 레인지 영상은 빛의 밝기를 입력한 영상이다.
- ④ 표본화간격을 나이키스트 간격보다 조밀하게 하면 원래의 신호를 복원해 낼 수 있다.

[정답] ④

[풀이] 표본화는 연속적으로 변화하는 아날로그 신호로부터 이산적인 신호로 변환하는 것이다. 이 때 아날로그 신호가 가지고 있는 최대 주파수의 2배 이상의 빠르기로 표본화하여 표본화 간격을 줄이면 이론적으로 원래 신호를 그대로 복원할 수 있다. 이 주파수에 해당되는 표본화 간격을 나이키스트 간격이라고 한다.

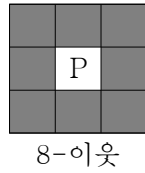
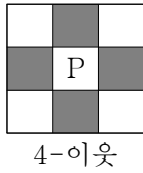
44. 다음 중 4-이웃 연결성을 적용할 때 픽셀 P의 이웃 픽셀들은?

- ① (㉠), (㉡), (㉢), (㉣)
- ② (㉠), (㉡), (㉢), (㉣), (㉤), (㉥), (㉦), (㉧), (㉨)
- ③ (㉡), (㉢), (㉣), (㉦)
- ④ (㉡), (㉢), (㉦), (㉧)

(㉠)	(㉡)	(㉢)
(㉣)	P	(㉤)
(㉥)	(㉦)	(㉧)

[정답] ③

[해설] 4-이웃 연결성 가장 가까운 4개의 픽셀을 이웃으로 정의한 것, 8-이웃 연결성은 가장 가까운 8개의 픽셀을 이웃으로 정의한 것이다.



45. 다음 중 영상의 평활화(smoothing)를 위한 필터 마스크는?

①  $\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

②  $\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$

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

④  $\frac{1}{4} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

[정답] ①

[해설] 평활화 필터는 상에 포함된 잡음을 제거하기 위해 한 픽셀의 값을 그 픽셀을 중심으로 한 이웃 픽셀들과의 가중치 평균을 구하는 것을 의미한다. ②는 라플라스 연산자, ③과 ④는 소벨 연산자이다.

46. 작은 점이 뿌려진 것과 같은 점잡음이 있는 영상에 대해 적용할 경우 효과가 큰 필터는?

① 라플라시안 필터

② 가우시안 평활화 필터

③ 소벨 필터

④ 중간값 필터

[정답] ④

[해설]  $n \times n$  크기의 중간값 필터는 원 영상의 각각의 픽셀 위치에 대하여 그 위치를 중심으로  $n \times n$  영역에 속한 픽셀의 그레이 레벨값을 크기순으로 나열한 후, 그중 중간값을 그 픽셀위치에 대한 결과 영상의 픽셀값으로 취하는 것이다. 그 결과 점 잡음에 해당되는 픽셀이 제거되는 효과를 얻을 수 있다. 그러나 원래 있던 작은 점이나 가느다란 직선 등이 함께 제거되는 문제점이 있을 수 있다. 가우시안 평활화 필터 역시 잡음제거의 역할을 하나 점 잡음에 대해서는 중간값 필터가 더 우수하다.

47. 다음 중 영상에서 에지를 검출하기 위해 사용되는 것은?

① 가우시안 필터

② 중간값 필터

③ 소벨 연산자

④ 저역통과 필터

[정답] ③

[해설] 소벨 연산자는 1차 미분을 계산하여 수평 및 수직 방향의 에지를 구할 수 있으며, 그 결과 에지의 크기와 에지의 방향을 구할 수 있다. 나머지 필터는 영상의 잡음을 제거하기 위한 평활화 필터에 해당된다.



①  $\begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$

②  $\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

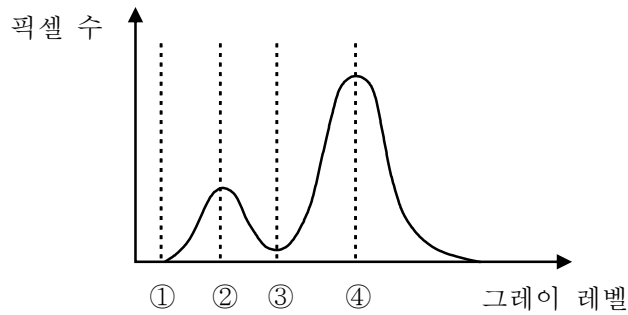
③  $\frac{1}{4} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

④  $\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

[정답] ③

[해설] 중앙 3개의 픽셀을 중심으로 오른쪽 3개의 픽셀 가중 합을 왼쪽 3개의 픽셀 가중 합으로 빼는 형태이므로 수평 방향의 그레이 레벨 변화를 검출할 수 있다.

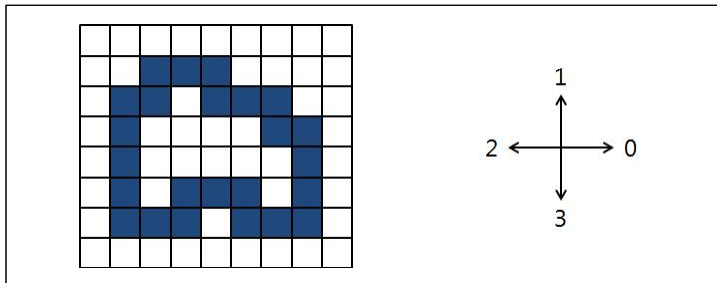
52. 균일한 밝기의 배경 위에 균일한 밝기의 물체가 놓여 있을 때 그레이 레벨 히스토그램이 아래 그림과 같이 나왔다. 배경과 물체를 구분하는 그레이 레벨은 어느 점이 적절한가?



[정답] ③

[해설] 균일한 배경에 해당되는 히스토그램 영역과 균일한 밝기의 물체에 해당되는 히스토그램 영역에 해당되는 2개의 피크를 갖는 히스토그램이다. 이 경우 두 피크치 사이의 최저점에 해당되는 곳을 임계치로 정할 수 있다.

53. 다음과 같이 4-이웃 연결성으로 표현된 경계선을 정규화한 차분 방향코드로 표현한 것은?



- ① 0 0 0 3 1 3 0 3 1 0 3 1 3 0 0 3 0 3 1 0 1 3 0 3
- ② 3 0 3 1 0 3 1 3 0 0 3 0 3 1 0 1 3 0 3 0 0 0 3 1
- ③ 0 1 0 0 3 0 0 3 0 3 3 3 2 2 1 2 2 3 2 2 1 1 1 1





- ③  $J(\mathbf{x}, \mathbf{y}) \neq J(\mathbf{y}, \mathbf{x})$
- ④  $J(\mathbf{x}, \mathbf{y}) + J(\mathbf{y}, \mathbf{z}) \leq J(\mathbf{x}, \mathbf{z})$

[정답] ②

[해설] 자기 자신과의 거리는 0이므로  $J(\mathbf{x}, \mathbf{x}) = 0$ 이어야 하며, 역으로  $J(\mathbf{x}, \mathbf{y}) = 0$ 인 경우 두 점  $\mathbf{x}$ 와  $\mathbf{y}$ 는 같은 위치에 있다. 거리는 음수가 될 수 없으므로  $J(\mathbf{x}, \mathbf{y}) \geq 0$ 여야 한다.  $\mathbf{x}$ 로부터  $\mathbf{y}$ 까지의 거리는  $\mathbf{y}$ 로부터  $\mathbf{x}$ 까지의 거리와 같으므로  $J(\mathbf{x}, \mathbf{y}) = J(\mathbf{y}, \mathbf{x})$ 이고, 두 점 사이의 거리는 제3의 점을 거쳐서 가는 거리보다는 길지 않아야 하므로  $J(\mathbf{x}, \mathbf{y}) + J(\mathbf{y}, \mathbf{z}) \geq J(\mathbf{x}, \mathbf{z})$ 이다.

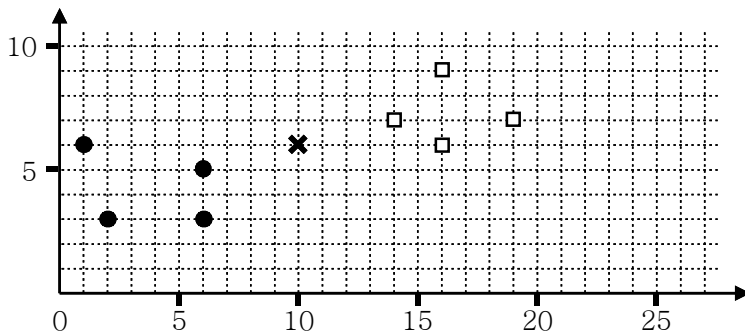
61. 다음 중 군집의 평균벡터와 공분산행렬을 고려한 거리측정자는 무엇인가?

- ① 마할라노비스 거리
- ② 해밍거리
- ③ 맨해튼 거리
- ④ 유클리드 거리

[정답] ①

[해설] 마할라노비스 거리는 특징공간상에서 군집의 분포를 고려하여 거리를 측정하는 통계적 척도이다.

62.  $k$ -NN을 인식방법으로 사용할 때, 거리측정자가 (ㄱ) 도시블록 거리를 사용하는 경우와 (ㄴ) 유클리드 거리를 사용하는 경우  $\times$ 를 어느 패턴으로 인식하게 되는가? 단,  $k=3$ 이다.



- ① (ㄱ) ●, (ㄴ) □
- ② (ㄱ) □, (ㄴ) ●
- ③ (ㄱ) ●, (ㄴ) ●
- ④ (ㄱ) □, (ㄴ) □

[정답] ②

[해설] 도시 블록 거리를 사용할 경우  $\times$ 와  $\bullet$ 들 사이의 거리는 가까운 것부터 5, 7, 9, 11이고,  $\times$ 와  $\square$ 들 사이의 거리는 5, 6, 9, 10이다.  $k=3$ 인 경우  $\bullet$ 가 1개,  $\square$ 가 2개 선택되므로  $\square$ 로 인식한다. 반면 유클리드 거리를 사용하는 경우  $\times$ 와  $\bullet$ 들 사이의 거리는 가까운 것부터 4.12, 5, 8.54, 9이고,  $\times$ 와  $\square$ 들 사이의 거리는 4.12, 6, 6.71, 9.06이다. 이 경우는  $\bullet$ 가 2개,  $\square$ 가 1개 선택되므로  $\bullet$ 로 인식한다.

63. 다음 중 교사에 의해 무엇을 할 것인가를 지시받는 것이라기보다는 보상을 통해 학습을 하는 방식을 의미하는 것은?





있음을 의미하며, 0이면 한 클래스에 속하는 표본만 있다는 의미이다. 따라서 불순도가 높으면 노드를 분할하고, 그렇지 않으면 그 노드는 더 이상 분할하지 않고 단말노드가 된다. 엔트로피는 불순도검사의 척도로 사용될 수 있다. 엔트로피는 데이터 집합이 가지고 있는 불확실성을 의미한다. 엔트로피값이 클수록 불확실성이 높으며, 이는 서로 다른 표본이 뒤섞여 있음을 의미하므로 불순도가 높은 것으로 볼 수 있다.

67. 다음 중 자율학습 방식의 군집화 알고리즘에 속하는 것은?

- ①  $k$ -NN 알고리즘
- ② 결정트리
- ③ 퍼셉트론
- ④  $k$ -평균 알고리즘

[정답] ④

[해설]  $k$ -평균 알고리즘은 표본집합을 대표하는  $k$ 개의 평균벡터를 찾음으로써 특징공간을  $k$ 개의 영역으로 분할하는 군집화 알고리즘이다.

68. 다음은  $k$ -평균 알고리즘이다. 공란 (가)에 넣을 적절한 처리는?

```

function kMeansClustering(x, k)
    // x[1..N] : 학습 표본 데이터 집합
    // m[1..k] : 평균벡터
    // c[1..k] : 가장 가까운 평균벡터가 m[j]인 학습표본의 수
    m[1..k]의 초기위치 선택;
    repeat
        mOld = m;
        // x[1..N]을 k개의 군집에 분류
        for i=1 to N do
            학습표본 x[i]에 대해 가장 가까운 평균벡터 m[j]를 선택;
            x[i]를 군집 j로 분류함;
        end-for;
        // m[1..k]를 새로운 군집의 평균으로 조정
        for j=1 to k do
            _____ (가) _____;
        end-for;
    until m == mOld; // m에 변화가 없으면 마침
    return m;
end-function;
    
```

- ① 군집  $j$ 의 학습 표본들을 제거한다.
- ②  $m[j] =$  군집  $j$ 로 분류된 학습표본들의 평균
- ③  $m[j] = x[j]$
- ④  $m[j] = mOld[j]$

[정답] ②

[해설] k-평균 알고리즘에서는 각각의 학습표본 벡터에 대해 k개의 평균벡터 중 가장 가까운 평균벡터를 구하여 그 평균벡터의 클래스로 분류한 다음 각각의 평균벡터를 그 평균벡터에 분류된 학습표본 벡터의 평균으로 갱신하는 것을 반복한다.

69. 신경회로망에 대한 올바른 설명은?

- ① 하나의 고성능 프로세서가 모든 연산을 담당한다.
- ② 하나의 뉴런이 손상되면 전체 시스템이 마비된다.
- ③ 뉴런 간의 연결강도를 조절함으로써 정보를 저장한다.
- ④ 한 뉴런은 다른 한 뉴런에만 연결된다.

[정답] ③

[해설] 신경회로망은 단순한 기능을 하는 수많은 뉴런이 상호간의 방대한 연결을 유지하며 대단 위 병렬처리를 하는 시스템이다. 정보는 뉴런 사이의 신경연접에 분산 저장되며, 학습기능을 통해 신경연접의 연결강도를 조정할 수 있다.

70. 퍼셉트론 모델에 대한 올바른 설명은?

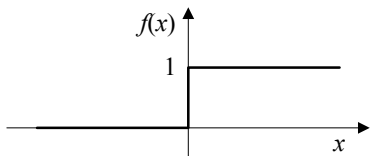
- ① 비선형 결정경계를 형성한다.
- ② XOR 문제를 해결할 수 없다.
- ③ 입력층, 은닉층, 출력층의 3층 구조이다.
- ④ 입력은 2개만 받는다.

[정답] ②

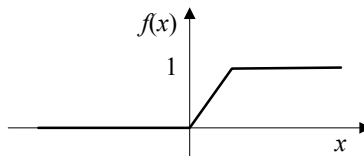
[해설] 퍼셉트론은 선형 결정경계만을 학습할 수 있다. 따라서 특징공간상에서 하나의 직선으로 경계를 정할 수 있는 문제가 아니라면 해결할 수 없으며, XOR 문제는 퍼셉트론이 풀이할 수 없는 문제 중 하나이다.

71. 오차역전파 모델에서 사용하는 활성화함수는?

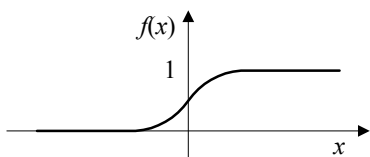
①



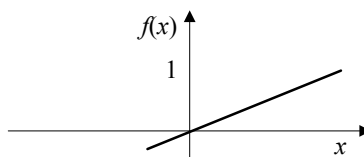
②



③



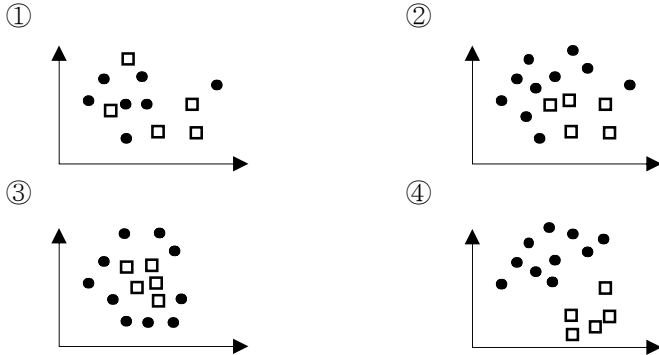
④



[정답] ③

[해설] 오차역전파 모델에서는 시그모이드 함수를 사용한다. 시그모이드 함수는 미분 가능하며, 자동 이득 조절 특성이 있는 함수이다.

72. 다음의 패턴분포 중 단층 퍼셉트론으로 학습할 수 있는 것은?



[정답] ④

[해설] 퍼셉트론은 하나의 직선으로 경계를 구분할 수 있는 경우 학습할 수 있다.

73. 오차역전파 모델에 대한 설명으로 올바른 것은?

- ① 자율학습을 한다.
- ② XOR 문제를 학습할 수 없다.
- ③ 뉴런의 활성화함수는 시그모이드를 사용한다.
- ④ 단층 퍼셉트론에서만 사용될 수 있는 학습모델이다.

[정답] ③

[해설] 오차역전파 모델은 비선형 결정경계를 갖는 경우에도 학습할 수 있는 다층 신경망의 학습 방법이다. 지도학습을 사용하며, 활성화함수는 시그모이드를 사용한다.

74. 신경회로망에서 정보가 저장되는 형태는?

- ① 중앙집중 방식으로 데이터를 저장한다.
- ② 저장기능이 없다.
- ③ 연결가중치의 형태로 분산되어 저장된다.
- ④ 활성화함수에 저장된다.

[정답] ③

[해설] 신경회로망에서 정보는 뉴런 사이의 연결강도를 통해 저장되며, 어느 한 곳에 집중되어 저장되는 것이 아니라 신경망 전반에 분산된다.

75. 다음 중 경쟁학습을 하는 학습모델은?

- ① 자기조직화 지도
- ② 퍼셉트론
- ③ 오차역전파
- ④ 결정트리

[정답] ①

[해설] 자기조직화 지도에서는 학습표본 벡터가 입력되면 출력노드들이 서로 경쟁을 하여 가장 가까운 대표벡터에 해당되는 연결가중치를 갖는 노드만 승자노드가 되고, 이 승자노드의 연결가중치를 학습표본 벡터와 가까운 방향으로 변화시키는 학습을 한다.

76. LVQ에서 어떠한 학습표본  $\mathbf{x}_j$ 와 가장 유사한 연결가중치 벡터  $\mathbf{w}_i$ 를 통해 연결된 출력노드가 그 학습표본과 같은 클래스에 속한다고 할 때 어떠한 처리를 하는가?

- ① 그 학습표본을 학습 표본집합에서 제거한다.
- ②  $\mathbf{w}_i$ 를  $\mathbf{x}_j$ 에 가까워지는 방향으로 조금 이동한다.
- ③  $\mathbf{w}_i$ 를 리셋한다.
- ④  $\mathbf{w}_i$ 를  $\mathbf{x}_j$ 에 멀어지는 방향으로 조금 이동한다.

[정답] ②

[해설] LVQ는 지도학습을 하는데, 어떠한 학습표본  $\mathbf{x}_j$ 와 가장 유사한 연결가중치 벡터  $\mathbf{w}_i$ 를 통해 연결된 출력노드가 그 학습표본과 같은 클래스에 속한다면  $\mathbf{w}_i$ 를  $\mathbf{x}_j$ 에 가까워지는 방향으로 이동하고, 다른 클래스에 속한다면  $\mathbf{w}_i$ 를  $\mathbf{x}_j$ 에서 멀어지는 방향으로 이동한다.