

인공지능 워크북

학습 주안점

인공지능은 컴퓨터에게 지능적인 문제해결 능력을 부여하기 위한 제반 학문 분야를 통칭하는 것이다. 본 과목에서는 문제풀이 방식, 지식의 표현 및 사용, 퍼지 이론, 전문가 시스템, 패턴 인식, 자연어 이해 등 인공지능의 전반적인 분야에 대한 기본적인 이해를 하는 것을 목표로 하고 있다.

이 워크북은 인공지능 과목 강의의 시청을 위한 최소한의 학습내용 요약과 보조 학습내용을 담고 있어 교과목 전체의 내용이 포함되어 있지는 않다. 그러므로 워크북 내용을 중심으로 수강하되, 각 장별 학습지침에 포함된 내용도 교재를 통해 학습하기 바란다.

중간시험은 교재 1~4장(방송강의 1~5강) 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 5~14장(방송강의 6~20강) 및 비출석수업 온라인 강좌 범위에서 출제하며, 전 범위에 걸쳐 고루 출제된다. 방송강의를 중심으로 하여 각 강별 학습지침에 따라 학습하기 바란다.

목 차

제1강	인공지능이란 무엇인가?
제2강	인공지능의 문제
제3강	문제의 표현
제4강	문제풀이방식-탐색 (1)
제5강	문제풀이방식-탐색 (2)
제6강	문제 축소에 의한 풀이 방식 (1)
제7강	문제 축소에 의한 풀이 방식 (2)
제8강	지식의 표현방법(1)
제9강	지식의 표현방법(2)
제10강	논리에 의한 지식표현 (1)
제11강	논리에 의한 지식표현 (2)
제12강	퍼지이론
제13강	생성시스템
제14강	전문가 시스템
제15강	지능형 에이전트
제16강	패턴인식 (1)
제17강	패턴인식 (2)
제18강	자연어 이해
제19강	학습
제20강	신경회로망
중간시험 대비 문제와 해설	
기말시험 대비 문제와 해설	

제1강 인공지능이란 무엇인가?

▶ 주요 학습 내용

- 지능적 시스템의 이해
- 인공지능에서 사용하는 문제풀이 기법
- ☞ 교재 학습 범위 : 1~17쪽

▶ 학습 지침

제1강의 내용은 인공지능의 문제풀이 방식에 대한 기본적인 아이디어를 소개하는 것이 주요 목적이다. 일반적인 정보처리 방식처럼 효율적인 알고리즘으로 정형화하기 어려운 문제에 어떻게 접근하고 있는지 살펴보기 바란다.

1. 지능적 시스템의 이해

● 인공지능(artificial intelligence)이란

- ▷ 지능적인 컴퓨터 프로그램에 의해 움직이는 지능적 기계를 만들기 위한 과학 및 공학

☞ 인공지능 시스템의 사례 - 인간과 컴퓨터의 서양장기 대결

- ▷ 세계 체스 챔피언인 러시아의 게리 카스파로프와 이스라엘의 프로그래머들이 개발한 체스 프로그램 챔피언 ‘딥 주니어’와의 대결
 - 2003년 2월 - 1승 1패 4무승부
- ▷ 1997년에 있었던 게리 카스파로프와 IBM이 개발한 ‘딥 블루’와의 대결에서는 딥 블루가 2승 1패 3무승부로 이긴 바 있음

● 지능(intelligence)의 정의

- ▷ 새로운 사물 현상에 부딪쳐 그 의미를 이해하고 처리방법을 알아내는 지적 활동의 능력 (표준국어대사전)
- ▷ 배우거나 이해하는 능력, 또는 새로운 상황에 대처하는 능력 (Webster)
- ▷ 자동적으로 또는 본능에 의해 일을 처리하는 것이 아니라, 생각하고 추론하며 이해하는 능력 (Collins)

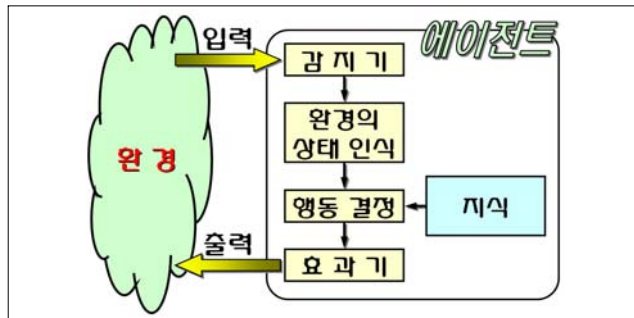
● 지능적 시스템

- ▷ **지식**을 사용한다.
- ▷ **학습**을 통해 새로운 지식을 습득하거나, 이미 학습된 지식을 보강한다.
- ▷ 외부의 환경을 인식함으로써 문제를 **이해**할 수 있다.
- ▷ 지식을 기반으로 **추론**함으로써 대상 문제의 해를 찾아낸다.



● 지능적 에이전트

▷ 지능적 시스템의 개념은 지능적 에이전트 구조의 근간을 이룬다.



2. 인공지능에서 사용하는 문제풀이 기법

● **기호처리 시스템 가설** (Newell과 Simon, 1976) : 인공지능 시스템에서의 기본적인 가설

인간의 지능은 두뇌 속에 형성된 개념이 기본으로, 이를 기호(심벌)로 표현할 수 있다.

⇨ 기호처리 시스템은 일반적인 지적 행동을 위한 필요충분한 수단이다.

※ 기호처리 시스템 가설이 갖는 의미

- ① 인간 지능의 본질에 대한 의미있는 이론으로서, 심리학자들이 큰 관심을 가지고 있다.
- ② 인간이 행하는 지능적 작업을 수행하는 프로그램을 작성할 수 있다는 가능성에 대한 믿음의 근원

● 인공지능 문제풀이의 기본 기법

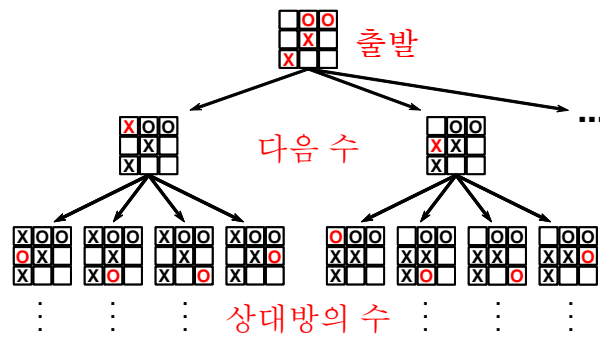
- ▷ 탐색 : 주어진 문제의 초기상태를 문제가 해결된 목표상태로 변화시키기 위한 풀이경로를 탐색하는 방법
- ▷ 지식의 사용 : 문제풀이에 활용될 수 있는 지식을 사용하여 보다 효과적으로 문제를 해결
- ▷ 문제의 분리 및 축소 : 주어진 문제를 보다 풀이하기 쉬운 작은 부분문제들로 분할하여 풀이하는 방법

● 문제 풀이의 예 : 삼목 게임(tic-tac-toe)

- ▷ 수평, 수직, 또는 대각선 중 하나를 먼저 모두 점령하면 이기는 게임

O		X
X	X	X
O		O

- ▷ 현재의 판의 형태에 대해 내가 둘 수 있는 수와, 이에 대해 둘 수 있는 상대방의 응수들을 구해 보면 거대한 그래프가 형성된다.
- ▷ 이러한 그래프 공간에서 나에게 가장 좋은 결과에 해당되는 판의 형태로 향하는 경로를 탐색한다.



- ▷ 방대한 그래프 공간에서 맹목적으로 탐색하는 것은 매우 많은 시간과 자원을 소비한다.
 - 효율적인 탐색이 이루어질 수 있도록 적절한 지식을 사용한다.
- ※ 다른 방법들에 비해 많은 시간을 요함.
- ※ 보다 복잡한 게임에 대한 확장이 쉬우며, 여러 가지 지식을 적용할 수 있다.

제2강 인공지능의 문제

▶ 주요 학습 내용

- 인공지능 문제의 특징
- 문제풀이 방법 - 탐색
- 문제풀이 방법 - 문제축소
- 교재 학습 범위 : 19~33쪽

▶ 학습 지침

제2강에서는 인공지능 문제풀이의 기본적 접근방법인 탐색과 문제축소의 개념을 이해한다. 상태, 연산자, 상태공간 등의 용어를 익혀둔다.

1. 인공지능적 문제의 특징

● 인공지능 문제의 특징

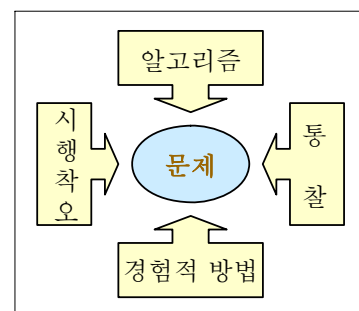
- ▷ 탐색형 추론 문제
 - 알고리즘화하기 어려운 비정형적인 문제에 대하여, 해에 도달하기 위한 수많은 경로를 시행착오적, 경험적 방법을 동원하여 적절한 해를 찾는다.
- ▷ 연역형 추론 문제
 - 정확한 전제로부터 정확한 결론을 얻어내는 형식논리의 추론방식을 사용
- ▷ 초기상태, 목표상태, 연산자로 표현
 - 주어진 문제(초기상태)를 변화시켜(연산자) 원하는 해(목표상태)를 얻음.

● 문제풀이를 위해 필요한 작업

- ▷ 문제를 정확하게 정의
 - 문제의 초기상황과 풀이된 결과로 받아들일 수 있는 상황에 대한 정확한 명세를 제시한다.
- ▷ 문제를 분석
 - 문제를 풀이기 위한 핵심적인 요소를 찾아낸다.
- ▷ 문제를 풀이기 위해 사용할 수 있는 지식을 표현한다. 이때 이 지식을 문제풀이 메커니즘과 분리함으로써 지식의 유지, 보수, 확장이 용이하도록 하는 것이 바람직하다.
- ▷ 적절한 문제 풀이기법을 선택하여 적용

● 문제풀이에 사용될 수 있는 전략

- ▷ 시행착오와 통찰
 - 시행착오 : 가능한 해결책을 한 가지 시도해 보고 나서, 해결되지 않으면 차례로 다른 해결책을 시도해 보는 것
 - 통찰 : 새로운 시점에서 문제의 전체 구조를 파악하는 것
- ▷ 알고리즘과 경험적 방법
 - 알고리즘 : 문제를 해결하기 위해 명확히 정의된 절차
 - 경험적 방법(heuristics) : 이미 정립된 공식에 의한 것이



아니라, 정보가 완전하지 않은 상황에서 시행착오나 경험을 통해 얻은, 완전하지는 않지만 많은 경우 성립하는 문제풀이 방법

2. 문제풀이 방법 - 탐색

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

▷ 초기상태 : 주어진 문제를 나타내는 상태

▷ 목표상태 : 주어진 문제가 풀이된 결과를 나타내는 상태

(예) 8-퍼즐 문제

가로, 세로 각각 3개씩의 격자에 8개의 조각이 놓여 있고, 한 곳은 비어 있다. 빈 격자를 이용하여 조각을 한 개씩 이동시켜 원하는 배치를 만들어 낸다.

▷ 초기상태 및 목표상태

1	2	3
4		5
6	7	8

초기상태



1	2	3
4	5	6
7	8	

목표상태

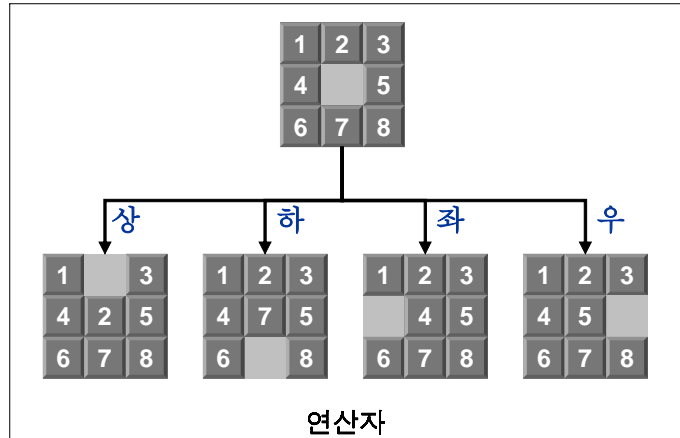
● 상태의 변화

▷ 문제를 풀이하려면 주어진 상태를 변화시켜 목표상태로 움직여 가야 한다.

▷ **연산자(operator)** : 어떠한 상태가 주어졌을 때 이를 새로운 상태로 변화시키는 역할을 하는 것

▷ 8-퍼즐 문제의 연산자 : 실제로는 8개의 조각 중 하나를 움직이는 것이나, 빈 칸을 이동시키는 것으로 볼 수 있다.

▫ 상, 하, 좌, 우의 네 개의 연산자



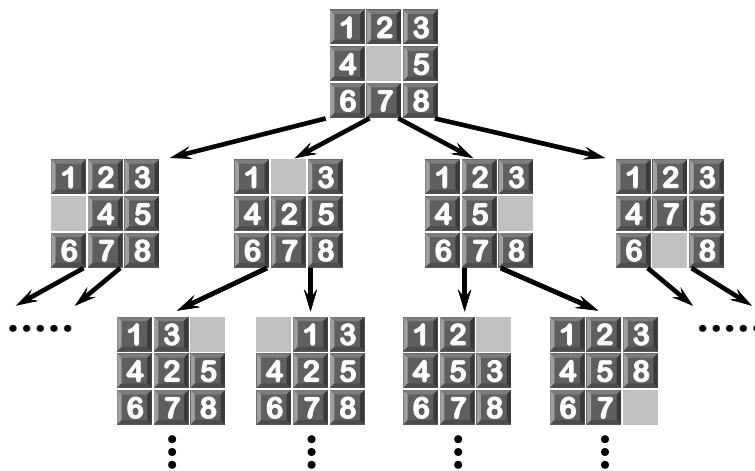
※ 문제의 정의 : (초기상태, 목표상태, 연산자)

▫ 초기상태와 목표상태 및 각각의 상태를 변화시킬 수 있는 정당한 방법들인 연산자를 통해 문제를 정의할 수 있다.

※ 실제로 문제풀이를 하는 과정에서는 초기상태에서부터 다양한 상태로 변화시켜 나갈 수 있다.

⇒ **상태공간(state space)** : 초기상태에서부터 정의된 연산자를 적용하여 만들어 질 수 있는 모든 상태들의 집합을 상태공간이라 한다.

▷ 8-퍼즐 문제의 상태공간



● 탐색에 의한 문제풀이

- ▷ 상태공간을 탐색하여 초기상태에서부터 목표상태로 이르는 경로를 찾음으로써 문제를 풀이
- ▷ 초기상태, 목표상태, 연산자로 문제를 정의
- ▷ 주어진 상태에 적용 가능한 연산자들을 적용하여 차기상태들을 생성한다.
- ▷ 미리 정의된 기준에 의거하여 조사되지 않은 차기상태들의 집합에서 하나를 선택한다.
- ⇒ 정의된 기준을 만족하는 목표노드까지 도달하는 경로가 발견되면 문제가 풀이가 된다.

※ 상태공간 탐색방법의 특성

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

3. 문제풀이 방법 - 문제 축소(Problem reduction)

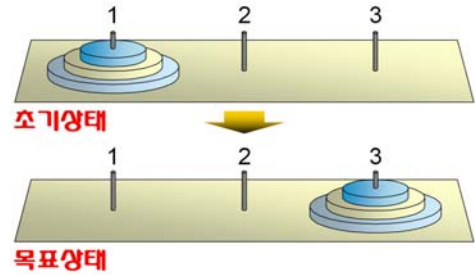
- ※ 주어진 문제를 바로 탐색하여 문제를 풀이하지 않고, 이를 몇 개의 부분 문제로 나누면 쉽게 풀이할 수 있는 경우가 있다. ⇒ 문제 축소

● 문제 축소

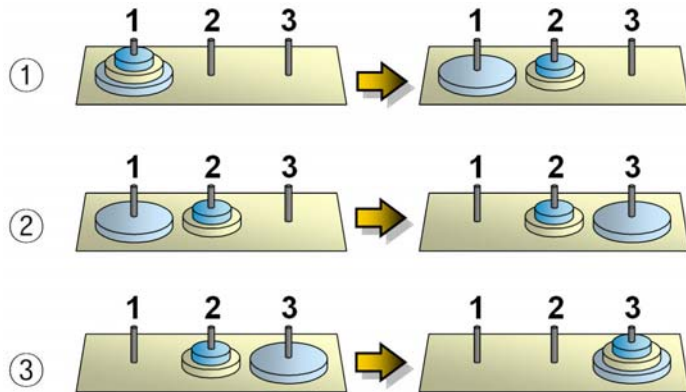
- ▷ 주어진 문제를 분석하여 보다 단순화된 부분 문제로 분해한다.
- ▷ 문제를 분해하는 과정을 반복하여 아주 쉽게 풀이할 수 있는 작은 문제로 분해되면 원래의 문제를 해결할 수 있다.

(예) 하노이 탑 문제

세 개의 말뚝이 있고, 그 중 첫 번째 말뚝에 n 개의 크기가 다른 원판이 꽂혀 있다. 주어진 세 개의 말뚝을 이용하여 첫 번째 말뚝에 있는 원판을 모두 세 번째 말뚝으로 옮겨라. 이때 작은 원판은 언제나 큰 원판의 위에 있어야만 한다.



▷ 문제 분해 : 세 개의 부분 문제로 축소할 수 있다.



▷ 각각의 부분 문제는 또 다시 더욱 작은 문제로 분해될 수 있다.

제3강 문제의 표현

주요 학습 내용

- 상태묘사 및 연산자 정의
- 문제 표현의 예
- ☞ 교재 학습 범위 : 35~54쪽

학습 지침

제3강의 내용은 컴퓨터 내에 어떻게 문제를 표현해야 할 것인가에 대한 것이다. 문제를 표현하고, 문제의 상태를 효과적으로 변화시키기 위해 어떠한 자료구조가 적합한지 생각하며 학습한다.

1. 상태 묘사 및 연산자 정의

● 문제의 표현


- ▷ 초기상태, 목표상태, 연산자로 표현함.
 - 초기상태 : 최초에 주어진 문제를 나타내는 상태
 - 목표상태 : 주어진 문제가 풀이된 결과를 나타내는 상태
 - 연산자 : 문제의 상태를 변화시키는 수단

● 상태 : 문제풀이 과정 중 어느 한 시점에서 문제의 형태

- ▷ 상태 묘사(state description) : 상태를 적절한 자료구조를 이용하여 표현한 것
 - 벡터, 트리, 리스트 등 표현하고자 하는 대상에 적합한 자료구조 활용

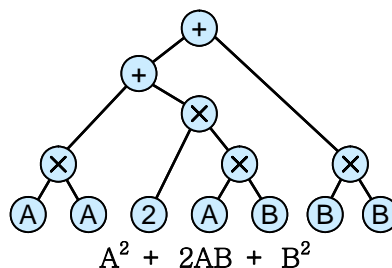
▷ 상태 묘사의 예

- 8-퍼즐 문제 : 2차원 배열로 표현할 수 있음.

	<pre>typedef struct { int BlankX, BlankY; char Board[3][3]; } PuzzleType;</pre>
---	--

▷ 상태 묘사의 예

- 수식처리 문제 - 트리를 이용하여 수식을 표현할 수 있음.



● 연산자

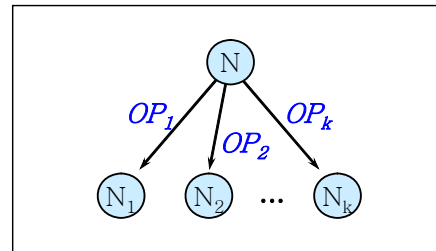
- ▷ 주어진 상태를 다른 상태로 변화시키는 것
- ▷ 연산자의 정의 : 하나의 상태 묘사를 다른 상태 묘사로 변화시키는 일종의 연산능력을 지닌 함수로 정의

(예) 8-퍼즐 문제

```
int OpUP(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;
}
```

● 상태공간의 표현 : 그래프 형태로 표현됨

- ▷ 그래프 : 노드의 집합과 노드를 연결하는 간선의 집합으로 구성
 - 노드는 상태를 표현함.
 - 간선은 연산자를 표현함.
 - 방향성 간선
 - 비용이 부가될 수 있다.



- 상태공간 : 초기상태로부터 시작하여 적용 가능한 연산자들을 반복적으로 적용하였을 때 나타날 수 있는 모든 상태들이 연결된 그래프

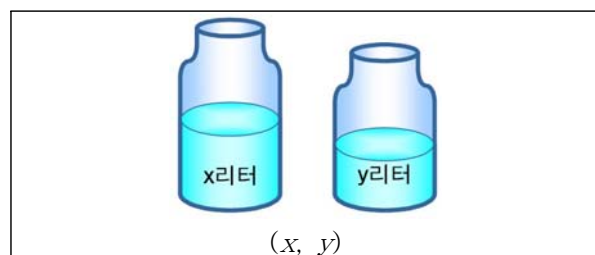
2. 문제 표현의 예 - 물병 문제

물병 문제 :

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.

● 상태 묘사

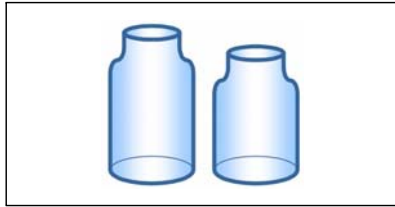
- ▷ 상태 : 4리터들이 및 3리터들이 물병에 들어있는 물의 양
 - ⇒ 두 통의 물의 양을 나열하여 상태를 묘사할 수 있다.
 - ⇒ 벡터 (x, y)



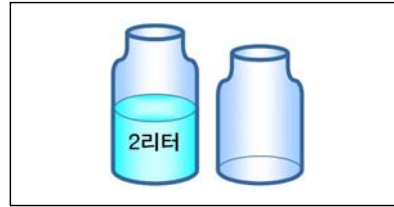
▷ 초기상태 및 목표상태

▫ 초기상태 : 두 물병이 모두 비어 있는 상태 $\Rightarrow (0, 0)$

▫ 목표상태 : 4리터들이 물병에 2리터의 물이 있고, 3리터들이 물병은 비어 있는 상태 $\Rightarrow (2, 0)$



초기상태
(0, 0)



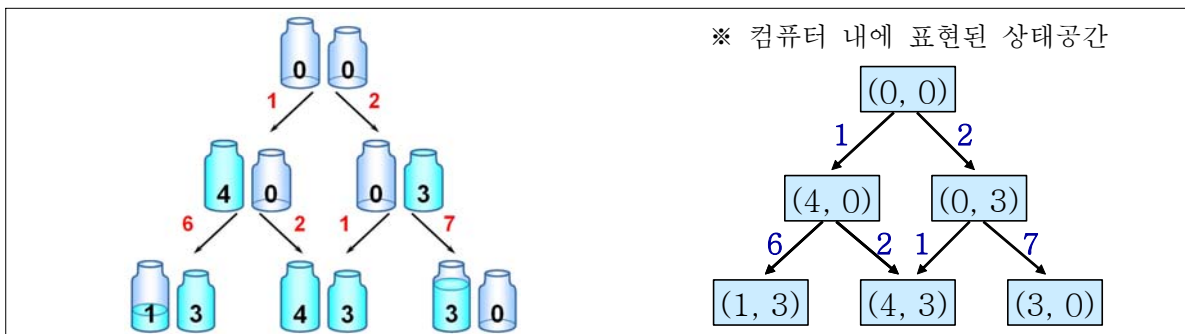
목표상태
(2, 0)

● 연산자의 정의

- ▷ 눈금이 없는 물병이므로 임의의 양의 물을 넣거나 옮기거나 버리는 연산은 의미가 없다.
- ▷ 조건 지정 - 불필요한 연산자를 반복 선택하여 적용하려는 노력을 줄일 수 있다.
- ▷ 연산자 : 교재 43쪽, [표3-1]

● 상태공간

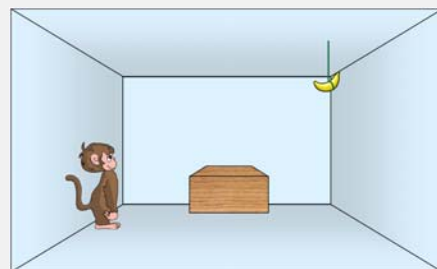
- ▷ 현재의 상태에 적용할 수 있는 연산자들을 이용하여 후계 상태를 생성
- ▷ 초기상태로부터 생성될 수 있는 모든 상태들이 연결된 그래프를 형성



3. 문제 표현의 예 - 원숭이와 바나나 문제

원숭이와 바나나 문제

어느 방 안에 원숭이가 한 마리 있고, 천장에는 바나나가 매달려 있다. 바나나의 높이는 원숭이가 잡기에는 너무 높다. 방 안에는 상자가 있어 그 상자 위에 올라서면 원숭이가 바나나를 잡을 수 있다. 원숭이가 바나나를 잡기 위해서는 어떻게 행동해야 하는가?



● 상태 묘사 - 4개의 요소로 구성

▷ 상태 : (w, x, y, z)

- w : 원숭이의 좌표
- x : 원숭이가 상자 위에 있는가를 나타내는 값, 0 또는 1
- y : 상자의 좌표
- z : 원숭이가 바나나를 가졌는지의 여부, 0 또는 1

※ 원숭이나 상자의 좌표값은 2차원 공간상에서 무한히 존재

⇒ 변수를 사용하여 표현

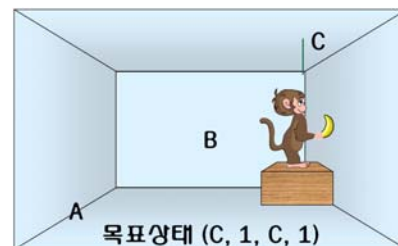
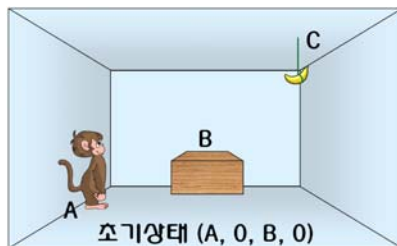
☞ 상태 묘사 스키마(상태 묘사 개요, state description schema)

● 연산자 - 원숭이의 행동을 연산자로 표현

▷ 원숭이가 취할 수 있는 동작은 매우 많지만 문제풀이에 필요한 4개의 동작을 연산자로 표현

- ① $goto(u) : (w, 0, y, z) \rightarrow (u, 0, y, z)$
- ② $pushbox(v) : (w, 0, w, z) \rightarrow (v, 0, v, z)$
- ③ $climbbox : (w, 0, w, z) \rightarrow (w, 1, w, z)$
- ④ $grasp : (c, 1, c, 0) \rightarrow (c, 1, c, 1)$

● 초기상태 및 목표상태



● 상태공간 : 교재 51쪽, [그림 3-8]

제4강 문제풀이 방식 - 탐색(맹목적 탐색)

주요 학습 내용

- 그래프 탐색의 개요
- 넓이우선 및 깊이우선 탐색
- 균일비용 탐색
- ☞ 교재 학습 범위 : 55~67쪽

학습 지침

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.

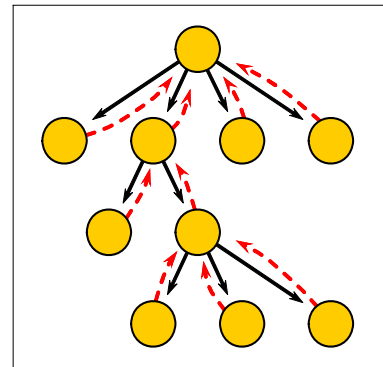
1. 그래프 탐색의 개요

※ 상태공간에서 목표상태의 탐색 : 초기상태에서부터 시작하여 연산자를 이용하여 새로운 차기상태들을 생성하고, 그 중 적절한 대상을 선택하여 다시 차기상태를 생성시키는 과정을 반복

- ⇒ 상태들 간의 의존 관계가 형성
- ⇒ 그래프로 표현

● 탐색과정

- ① 주어진 상태에 적용할 수 있는 모든 연산자를 가하여 모든 차기상태들을 생성
 - ⇒ 후계 노드의 집합 생성
 - ※ 확장 : 어떠한 상태에 적용 가능한 모든 연산자를 적용하여 모든 후계 상태를 생성하는 것
- ② 후계 노드에는 부모 노드를 가리키는 포인터를 첨부 - 탐색경로를 알 수 있도록 함.
- ③ 정해진 기준에 따라 다음 노드를 선택
 - ※ 앞의 기본적인 처리 스텝에 탐색순서를 결정하기 위한 몇 가지 처리 스텝이 포함된다.



● 탐색과정 중 사용하게 될 자료구조

- OPEN : 앞으로 확장하여야 할 노드들을 저장하는 리스트
- CLOSED : 이미 확장된 노드들을 저장하는 리스트

● 탐색에 사용되는 정보에 따른 분류

- ▷ 맹목적 탐색 : 목표 노드의 위치에 관계없이 단순히 미리 정한 순서에 따라 탐색을 하는 방법

▷ 경험적 탐색 : 목표 노드의 위치에 대한 경험적 정보를 사용하여 가장 타당한 노드를 다음 확장대상으로 선정

※ 경험적 지식 : 많은 시행착오를 거쳐 경험적으로 얻은 지식으로, 항상 맞아 떨어지는 완벽한 것은 아니지만, 많은 경우 유용하게 사용할 수 있는 지식

● 탐색의 목표에 따른 분류

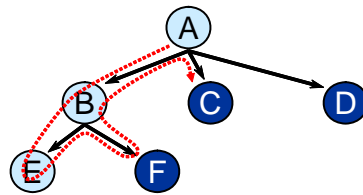
- ① 어떤 경로를 거치든 목표상태에 도달하는 경로를 찾기만 하면 되는 문제
- ② 초기상태로부터 목표상태에 도달하는 최적의 경로를 탐색하는 문제

정보 사용 \ 목적	임의 경로 탐색	최적 경로 탐색
맹목적 탐색	깊이우선 넓이우선	균일비용
경험적 탐색	언덕오르기 최적우선	A* 알고리즘

2. 맹목적 탐색 방법 - 깊이우선 및 넓이우선 탐색

● 깊이우선 탐색

(예)



OPEN	D	C	F					
CLOSED	A	B	E					

▷ OPEN, CLOSED - 초기에는 CLOSED는 비어 있고, OPEN에는 출발 노드를 넣은 상태에서 시작

▷ 노드를 확장한 결과 생성된 후계 노드는 OPEN의 끝에 추가

▷ 후계 노드 생성 후 : OPEN에 있는 노드 중 깊이방향의 노드를 선택

⇒ OPEN의 끝에서 1개의 노드를 꺼냄.

▫ 꺼내는 노드는 마지막으로 OPEN에 넣은 노드임.

⇒ OPEN은 스택 구조

▷ 계속해서 깊이방향으로 탐색해 들어갈 경우, 만약 이 경로에 해가 존재하지 않으면 탐색이 성공할 수 없다.

⇒ 깊이제한 설정

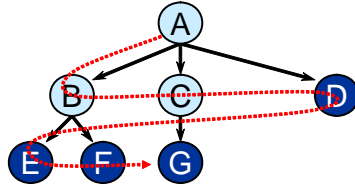
⇒ 깊이제한에 도달하면 탐색하지 않은 최근의 후계 노드가 존재하는 상태로 복귀하여 탐색을 계속

※ 백트래킹 : 다른 방향으로의 탐색이 가능한 최근의 상태로 복귀하는 것

▷ 깊이우선 탐색 알고리즘 : 교재 59쪽

◎ 넓이우선 탐색

(예)



OPEN	D	E	F	G			
CLOSED	A	B	C				

- ▷ OPEN, CLOSED - 초기에는 CLOSED는 비어 있고, OPEN에는 출발 노드를 넣은 상태에서 시작
- ▷ 노드를 확장한 결과 생성된 후계 노드는 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

- ▷ 넓이우선 탐색에 비해 적은 메모리 공간을 소비
- ▷ 우연히 아주 빠르게 풀이 경로를 찾을 수 있다.

◎ 넓이우선 탐색

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

3. 균일 비용 탐색 (uniform-cost search)

※ 상태의 변화는 비용을 요구

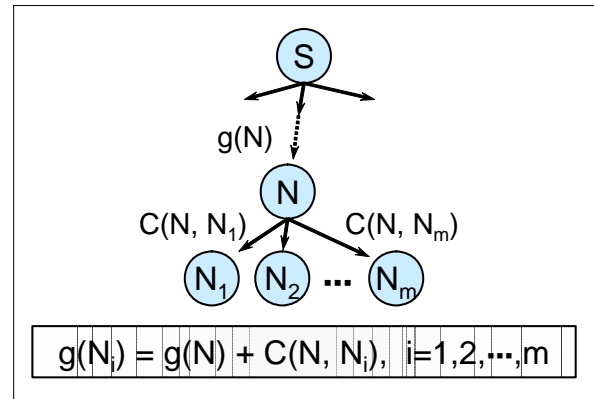
(예) 8-퍼즐 문제 : 조각을 이동시키는 비용 - 이동횟수

(예) 외판원 문제 : 도시를 이동할 때마다 이동거리에 해당되는 비용

⇒ 최소의 비용이 필요한 풀이경로는?

● 경로 비용의 계산

- ▷ $g(N)$: 출발노드(S)로부터 노드 N 까지의 최소 경로 비용
 - ▷ $C(N, N_i)$: 노드 N 과 후계 노드 N_i 사이의 경로 비용
 - ▷ 후계 노드 N_i 의 비용은 노드 N 의 최소 경로 비용에 노드 N 과 후계 노드 N_i 사이의 경로 비용을 합한 것
- $\Rightarrow g(N_i) = g(N) + C(N, N_i)$



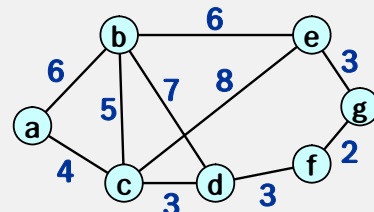
※ 최소 비용 경로의 탐색 : $g(N)$ 이 최소인 노드 N 을 우선 선택

● 균일비용 탐색 알고리즘 : 교재 64쪽

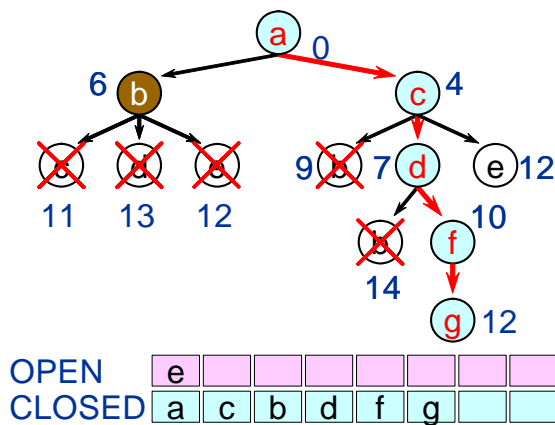
● 문제풀이 과정의 예

최단 경로 탐색 문제

a~g의 7개의 도시가 있다. 도시 사이에는 도로가 건설되어 있고, 각 도로의 거리는 그림과 같다. 도시a에서 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 목표 노드이면 그 경로는 최소 비용 경로
- ▷ 8-퍼즐 문제에서는 조각의 이동횟수가 비용이며, 따라서 모든 연산자의 적용 비용이 동일하므로 균일비용 탐색은 넓이우선 탐색과 동일한 탐색을 하게 된다.

제5강 문제풀이 방식 - 탐색(경험적 탐색)

주요 학습 내용

- 경험적 탐색의 개요
- 언덕 오르기 방법
- A*-알고리즘

교재 학습 범위 : 67~84쪽

학습 지침

경험적 지식이 도입된 평가함수가 탐색에 활용되는 점에 유의하며, 각각의 탐색방법에서 어떤 탐색결과를 얻을 수 있는가도 알고리즘별로 비교하며 학습한다.

1. 경험적 탐색의 개요

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

● 경험적 탐색

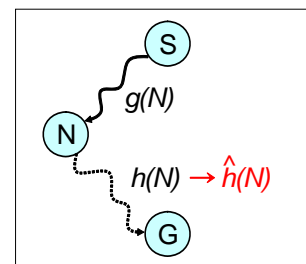
- ▷ 목표상태를 보다 신속하게 탐색하기 위해 경험적 지식을 활용하는 탐색 방법
- ▷ 평가함수 사용
 - 어떤 상태가 주어졌을 때 그 상태를 거쳐 가는 것이 목표상태로 가는 데 얼마나 바람직한가를 나타내는 함수
 - 해를 향해 가는 데 필요한 비용, 해로 향하는 경로상에 존재할 가능성 등

● 평가함수

- ▷ $g(N)$: 출발 노드 S 로부터 현재 상태를 나타내는 노드 N 까지의 경로비용
- ▷ $h(N)$: 노드 N 으로부터 목표 노드 G 까지의 경로 비용
 - 아직 탐색하지 않은 경로이므로 정확한 $h(N)$ 값을 알 수 없다.
 - ⇒ 예측 비용 $\hat{h}(N)$ 사용

※ 탐색 알고리즘에 따라 사용하는 요소에 차이가 있다.

- ▷ 언덕오르기 탐색 : $\hat{h}(N)$ 만 사용
- ▷ A* 알고리즘 : $g(N) + \hat{h}(N)$ 을 사용



2. 언덕오르기 탐색 방법

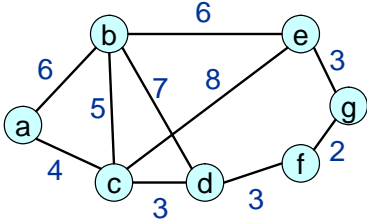
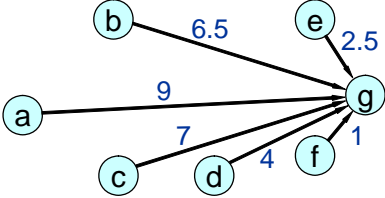
● 언덕오르기 탐색 알고리즘

- ▷ 깊이우선 탐색과 유사한 순서로 탐색

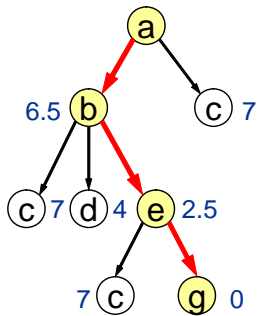
- ▷ 후계 노드의 평가함수를 계산하여, 가장 비용이 적은 노드를 다음 확장할 노드로 선택
- ▷ 평가함수는 후계 노드에서부터 목표 노드까지의 예측 경로비용을 사용하며, 후계 노드까지 도달하는 데 사용된 비용은 고려하지 않음.

● 경로 탐색 문제의 예

- ▷ 도시 a에서부터 도시 g까지 가는 경로를 탐색하는 문제
- ▷ 평가함수 : 목표도시까지의 직선거리

도시 간 거리	목표도시까지의 직선거리
	

▷ 탐색과정



※ 탐색 결과가 최소 비용 경로는 아님에 주의

● 계수 최적화 문제 : 최적의 결과를 낼 수 있도록 시스템의 계수를 조정하는 문제

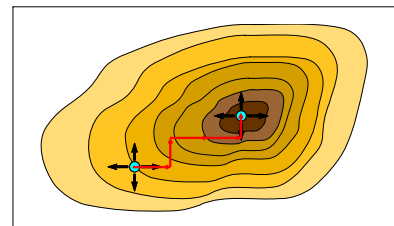
- ▷ 예 : 등산가 문제

등산가 문제

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



- 상태 : 좌표 및 고도
- 연산자 : 동,서,남,북 방향으로 정해진 거리만큼 이동
- 탐색과정 : 안개 때문에 일정 거리 이상의 높이는 측정할 수 없으므로 정해진 거리 이내로만 이동하는 것으로 제한
- 목표상태 : 어느 방향으로 이동하여도 고도가 낮아지는 지점



※ 계수 최적화 문제에서 발생할 수 있는 문제점

1. 지역 최대치(local maximum) 문제

- ▷ 정상 주변에 정상보다 낮은 봉우리들이 있을 때, 그 주변 봉우리에 오를 경우 그 곳이 정상인 것으로 판단할 수 있다.
- ⇒ 대처 방안 : 백트래킹을 통해 다른 가능한 경로를 탐색



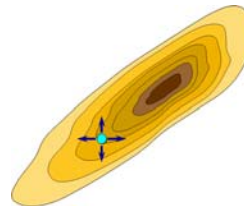
2. 고원(plateau) 문제

- ▷ 산 중턱에 존재하는 평평한 영역에 도달했을 경우 어느 방향으로 이동해도 현재 상황을 개선할 수 없다.
- ⇒ 대처 방안 : 동일 연산자를 반복 적용한 결과를 토대로 진행방향을 결정



3. 능선(ridge) 문제

- ▷ 산 중턱에 존재하는 평평한 영역에 도달했을 경우 어느 방향으로 이동해도 현재 상황을 개선할 수 없다.
- ⇒ 대처 방안 : 동일 연산자를 반복 적용한 결과를 토대로 진행방향을 결정

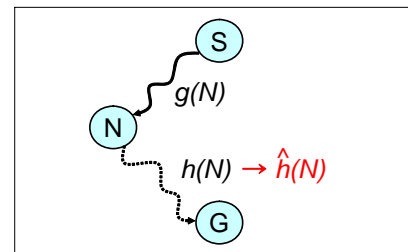


3. A* 알고리즘

※ 최적 경로를 탐색하기 위한 방법

● 평가함수

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



● A* 탐색 알고리즘

- ▷ 기본 골격은 타 알고리즘과 동일
- ▷ 특징적 내용
 - 비용 : 목표 노드로부터 현재 노드를 거쳐 목표 노드로 가는 경로의 예측비용
 - OPEN이나 CLOSED에 동일한 노드가 존재할 경우의 처리 방법

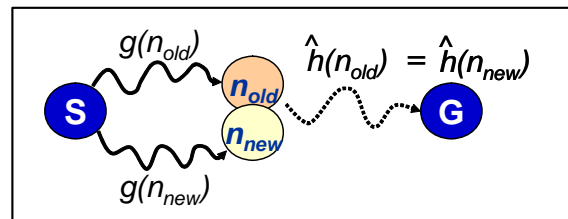
▷ 알고리즘

1. 출발 노드를 OPEN에 삽입한다. (\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 목표 노드라면 탐색 성공
 - 3) 노드 n 을 확장하여 후계 노드 n_1, n_2, \dots, n_i 를 생성
 - 4) 후계 노드들의 평가함수 $\hat{f}(n_1), \hat{f}(n_2), \dots, \hat{f}(n_i)$ 를 계산
 - 5) 후계 노드 $n_k, k=1, \dots, i$ 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

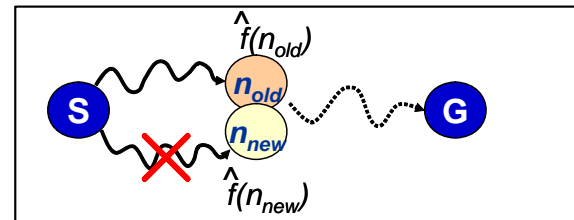
- ▣ 아직 어느 노드도 확장되지 않은 상태이면
로 평가함수 \hat{f} 이 큰 노드를 제거하면 됨
- ▣ $\hat{h}(n_{old})$ 와 $\hat{h}(n_{new})$ 는 동일
- ▣ g 값이 큰 노드를 제거한다.



▷ 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 된다.

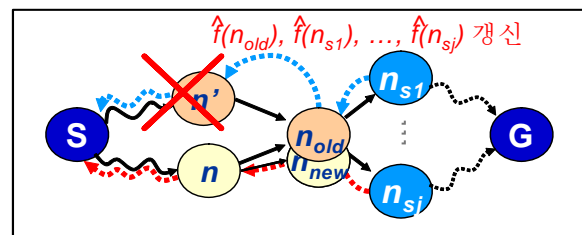


② $\hat{f}(n_{old}) > \hat{f}(n_{new})$ 인 경우

- ▣ n_{old} 를 제거해야 하는데, 이미 확장한 노드이기 때문에 후계노드들이 존재하므로 주의가 필요
- ▣ n_{old} 는 자신의 부모 노드를 가리키도록 포인터가 첨부되어 있고, n_{old} 의 후계 노드들이 OPEN이나 CLOSED에 들어 있다.

→ n_{old} 를 제거하는 대신 n_{new} 를 제거

- ▣ n_{old} 가 가리키던 부모 노드 포인터를 n_{new} 의 부모노드를 가리키도록 함
- ▣ n_{old} 및 n_{old} 의 후계 노드들의 평가함수 값을 재조정 ($g(n_{old})$ 가 바뀌었음)



● 최소 비용 경로 탐색을 위한 조건

만일 어떤 노드로부터 목표 노드까지의 경로 비용을 예측한 값이 항상 실제 비용보다 작다면, 즉 $\hat{h}(n) \leq h(n)$ 이 항상 만족된다면 A* 알고리즘은 최소 비용 경로를 탐색하는 것을 보장한다.

※ 증명

탐색과정 중 노드 n_1 을 거쳐 목표 노드 G가 생성되었고, OPEN의 노드 중에서 이 목표 노드 G의 평가함수 값이 최소여서 선택되었다고 하자.

- ▷ OPEN에 있는 임의의 노드 n_2 의 평가함수는 G의 평가함수보다 크다.

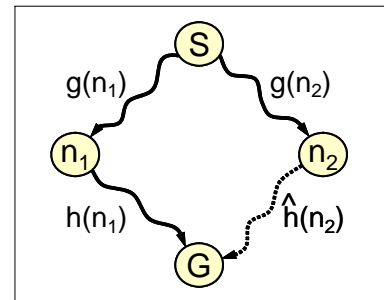
$$g(n_1) + h(n_1) \leq g(n_2) + \hat{h}(n_2)$$

- ▷ 앞의 조건에 따라 노드 n_2 로부터 목표노드까지의 실제 비용인 $h(n_2)$ 은 경로비용 예측치인 $\hat{h}(n_2)$ 보다 크거나 같다.

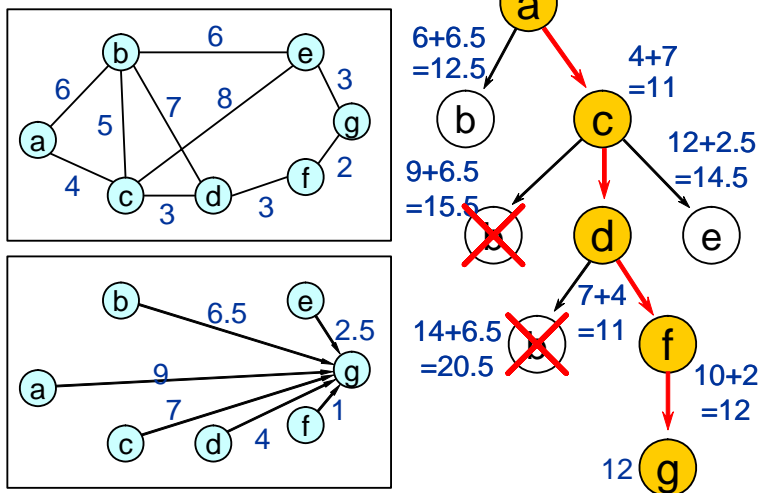
$$\begin{aligned} g(n_1) + h(n_1) &\leq g(n_2) + \hat{h}(n_2) \\ &\leq g(n_2) + h(n_2) \end{aligned}$$

⇒ OPEN에 있는 어느 노드도 n_1 을 거쳐 G에 도달한 경로에 비해 비용이 적을 수 없다.

⇒ 최소 비용 경로 탐색을 보장



(예) 최단 경로 탐색 문제



- ▷ 직선거리는 실제 거리보다 길지 않으므로 항상 $\hat{h}(n) \leq h(n)$ 이다.

⇒ 최단 거리 경로를 탐색하는 것을 보장한다.

제6강 문제 축소에 의한 문제 풀이 방식 (1)

주요 학습 내용

- 문제 축소의 개념
- AND/OR 그래프
- 문제 축소 기법
- ☞ 교재 학습 범위 : 85~97쪽

학습 지침

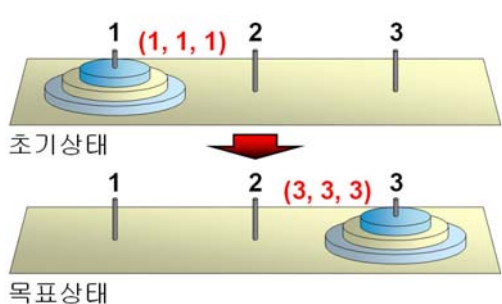
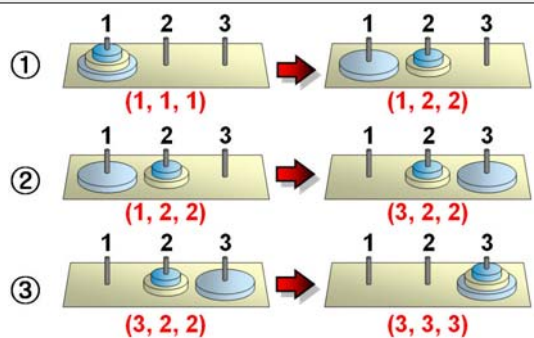
AND/OR 그래프와 문제 축소의 관계에 관심을 가지고 학습한다. 차이-연산자 표를 원숭이와 바나나 문제의 여러 가지 상황에 적용해 보면 문제 축소 과정을 쉽게 이해할 수 있다.

1. 문제 축소의 개념

● 문제 축소에 의한 문제풀이

- ▷ 주어진 문제를 몇 개의 부분 문제로 분할하여 각각의 부분 문제를 풀이함으로써 전체 문제를 풀이하는 접근방법
- ▷ 각각의 부분 문제들도 유사한 문제 축소 과정을 반복하며, 이때 더 이상 분해할 필요가 없는, 즉 해가 분명하거나 알려져 있는 문제(원시 문제)로 분할되었을 때 축소과정을 마친다.

(예) 하노이 탑 문제

주어진 문제	문제 축소
 <p>초기상태</p> <p>목표상태</p> <p>▷ 초기상태 (1 1 1), 목표상태 (3 3 3)</p>	 <p>※ 세 부분 문제가 모두 풀이되어야 주어진 문제가 풀이된다.</p>

※ 원시 문제 : 쉽게 풀이될 수 있거나 해가 완전히 알려져 있어서 더 이상 분해할 필요가 없는 문제

- ▷ 하노이 탑 문제의 부분 문제에서 문제 ②는 원시 문제이다.

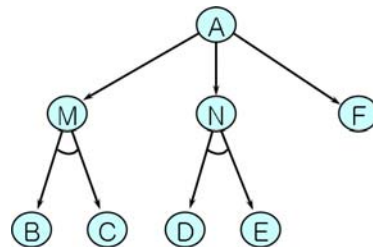
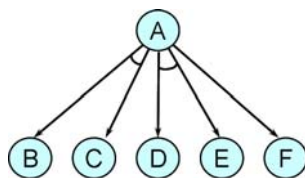
● 한 문제에 대해 부분 문제로 나눌 수 있는 방법은 여러 가지일 수 있다.

- ▷ 그 중 어느 하나라도 풀이되면 주어진 문제는 풀이된다.

2. AND/OR 그래프

- 어떠한 문제를 분해한 부분 문제 : 모든 부분 문제가 풀이되어야 주어진 문제가 풀이되므로 부분 문제들 간의 관계는 AND 관계
- 문제를 분해하는 방법이 여러 가지 있을 때 : 각각의 부분 문제 조합들 중 하나라도 풀이되면 주어진 문제는 풀이된다. → OR 관계
⇒ AND/OR 그래프로 표현할 수 있다.

AND/OR 그래프



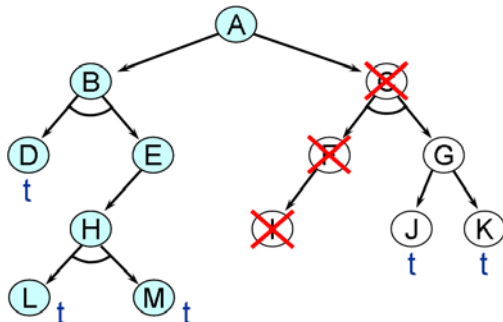
- ▷ 노드 : 부분 문제
- ▷ 묶은 간선 : AND 관계
- ▷ 묶이지 않은 간선 : OR 관계
- ▷ 종단 노드 : 원시 문제 묘사를 나타내는 노드

풀이될 수 있는 노드와 풀이될 수 없는 노드

풀이될 수 있는 노드	풀이될 수 없는 노드
① 종단 노드 ② OR 후계 노드들을 갖는 경우 적어도 한 노드가 풀이될 수 있을 때 ③ AND 후계 노드들을 갖는 경우 모든 후계 노드들이 풀이될 수 있을 때	① 후계 노드가 없는 비 종단 노드 ② OR 후계 노드들을 갖는 경우 모든 후계 노드들이 풀이될 수 없을 때 ③ AND 후계 노드들을 갖는 경우 어느 하나라도 풀이될 수 없을 때

※ 문제의 풀이 여부의 결정 : 문제 축소 과정에서 루트 노드가 풀이될 수 있는 것 또는 풀이될 수 없는 것으로 판명될 경우

▷ 풀이된 트리의 예 (t는 종단 노드를 나타냄).



(예) 적분 문제 풀이기 : 부정적분 식을 입력으로 받아서 적분결과를 구하는 자동 처리기

▷ 적분표 사용 : 기본 적분식을 저장하고 있는 표 - 원시 문제에 해당

$$① \int u dv = \frac{u^2}{2}$$

$$② \int \sin u du = -\cos u$$

$$③ \int a^u du = a^u \log_a e$$

④

▷ 문제 축소 연산자 : 부분적분 법칙, 적분 합의 분리, 삼각함수 대입 등 적분에 사용될 수 있는 여러 가지 규칙

(예) 부분적분 법칙

$$\int u dv = u \int dv - \int v du \quad \Rightarrow \quad \begin{array}{c} \int u dv \\ \swarrow \quad \searrow \\ \int dv \quad \int v du \end{array}$$

3. 문제 축소 기법

● 문제풀이 과정의 적절한 중간상태를 알 수 있다면, 그 중간상태를 이용하여 문제를 축소할 수 있다.

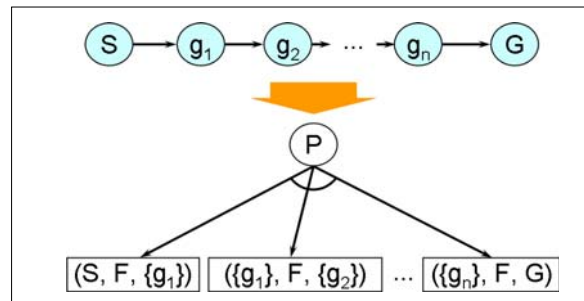
▷ g_1, g_2, \dots, g_n 이 거쳐가야 할 중간 상태라면 $n+1$ 개의 부분문제로 나눌 수 있다.

▫ 부분문제 1 : (S, F, g_1)

▫ 부분문제 2 : (g_1 , F, g_2)

.....

▫ 부분문제 $n+1$: (g_n , F, G)



※ 중간기점 상태를 쉽게 찾아내는 방법은?

⇒ 문제풀이 과정에서 사용해야 할 핵심적인 연산자(키 연산자)를 찾아내는 것이 용이한 경우가 많이 있다.

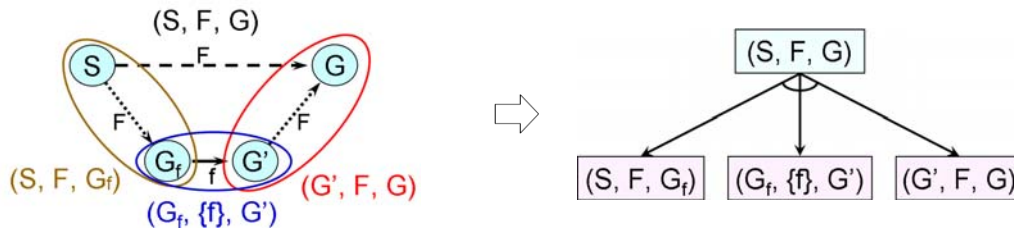
(예) 하노이 탑 문제 : 가장 큰 원판을 1번 말뚝에서 3번 말뚝으로 옮기는 연산자

⇒ 이 연산자를 적용시킬 수 있는 상태를 중간기점으로 하여 문제를 분할할 수 있다.

※ 키 연산자의 결정 : 초기상태와 목표상태의 차이를 구하여 이 차이를 해소할 수 있는 연산자를 키 연산자로 선택

▷ S와 G의 차이를 해소할 수 있는 연산자가 f라면 f를 적용할 수 있는 상태인 G_f 를 기점으로 분해

⇒ 세 개의 부분 문제로 분해된다.



(예) 원숭이와 바나나 문제

- ▷ 상태 : (w, x, y, z)
- ▷ 초기상태 : $(A, 0, B, 0)$
- ▷ 목표상태 : $(w, x, y, 1)$
- ▷ 연산자 : $goto(u)$, $pushbox(v)$, $climbbox$, $grasp$
- ▷ 바나나의 위치 : C
- ▷ 차이의 계산 : 차이-연산자 표 사용 (교재 95쪽, [표 5-1])
- ▷ 주어진 문제 P

P	$\{(A, 0, B, 0)\}, F, \{(w, x, y, 1)\}$
---	---

차이 : 원숭이가 바나나를 가지고 있지 않다.

➡ 키 연산자 : **grasp**

⇒ $grasp$ 을 적용하기 위해 필요한 전제 상태는 $(C, 1, C, 0)$

▷ 문제 P 의 분할

P	$\{(A, 0, B, 0)\}, F, \{(w, x, y, 1)\}$
---	---

grasp

P1	$\{(A, 0, B, 0)\}, F, \{(C, 1, C, 0)\}$
P2	$\{(C, 1, C, 0)\}, \{grasp\}, \{(C, 1, C, 1)\}$

※ $P2$ 는 원시 문제

※ $(C, 1, C, 1)$ 이 이미 목표상태이므로 세 번째 부분 문제는 필요없음.

▷ 원숭이와 바나나 문제의 문제 축소 그래프 : 교재 97쪽, [그림 5-6]

제7강 문제 축소에 의한 풀이 방식 [2]

▶ 주요 학습 내용

- AND/OR 그래프 탐색
- 게임 트리와 최대 최소 탐색
- ☞ 교재 학습 범위 : 97~111쪽

▶ 학습 지침

AND/OR 그래프 탐색의 기본적 틀은 기존 탐색 알고리즘과 동일하나, AND/OR 그래프의 풀이될 수 있는 노드 및 풀이될 수 없는 노드의 조건과 관련된 풀이 표시 및 풀이 불가 표시 알고리즘이 결합된다. 최대 최소 탐색은 예시된 그래프에 대해 평가함수를 계산해 보면 쉽게 이해할 수 있다.

1. AND/OR 그래프 탐색

● 풀이 표시 알고리즘 : 교재 101쪽

- ▷ 탐색 중 어떠한 노드가 풀이되었을 때, 이로 인해 풀이되는 조상 노드들을 판단하는 과정
- ▷ 풀이될 수 있는 노드가 되기 위한 조건을 판별하여 조상 노드로 풀이 표시를 전파함.
- ▷ S-LIST : 풀이 표시를 조상 노드로 전파해야 할 노드를 임시로 저장하는 리스트

● 풀이불가표시 알고리즘 : 교재 102쪽

- ▷ 탐색 중 어떠한 노드가 풀이될 수 없는 것으로 판정되었을 때, 이로 인해 풀이될 수 없다고 판명되는 조상 노드들을 판단하는 과정
- ▷ 풀이될 수 없는 노드가 되기 위한 조건을 판별하여 조상 노드로 풀이 불가 표시를 전파함.
- ▷ S-LIST : 풀이 불가 표시를 조상 노드로 전파해야 할 노드를 임시로 저장하는 리스트

● 깊이우선 탐색 : 교재 100쪽

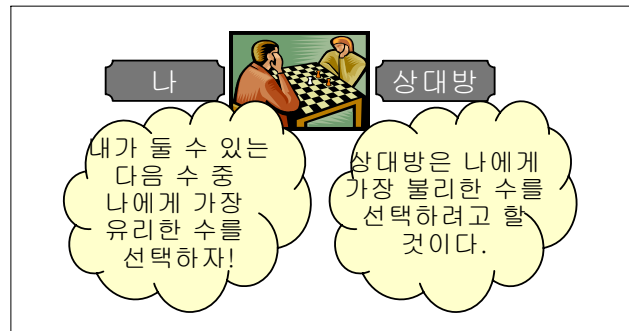
- ▷ 전반적인 알고리즘 구성은 일반적인 깊이우선 탐색과 동일함.
- ▷ 탐색과정 중 어떠한 노드가 풀이되면 풀이 표시 알고리즘을 적용
- ▷ 탐색과정 중 어떠한 노드가 풀이될 수 없는 것으로 입증되면 풀이 불가 표시 알고리즘을 적용

2. 게임트리와 최대 최소 탐색

● 상대가 있는 게임 : 장기, 바둑 등

▷ 나와 상대방이 번갈아 수를 둔다.

- 현재의 판의 상태에서 내가 둘 수 있는 수와 상대방이 둘 수 있는 수가 교대로 나타나는 트리가 구성
→ 최대 최소 탐색 트리



● 최대 최소 탐색 트리

- ▷ 루트 노드는 현재의 판의 상태이며, 이후 내가 둘 수 있는 수와 그 수에 대해 상대방이 둘 수 있는 수가 트리의 레벨에 따라 교대로 나타난다.
- ▷ 이러한 수의 예측은 시스템의 자원이나 시간적 제약이 허용하는 수준까지 이루어질 수 있으며, 이후는 각각의 판의 형태가 얼마나 나에게 유리한가를 평가함수에 의해 예측
- ▷ 나는 내가 둘 수 있는 여러 수들 중에서 나에게 가장 유리한 수를 두는 것이 좋으므로, 가장 큰 평가값을 선택 - 최대
- ▷ 상대방은 나에게 가장 불리한 수를 두려 할 것이라는 가정하에 가장 작은 평가값을 선택 - 최소
- ▷ 최대 최소 탐색 트리의 예 : 교재 107쪽, [그림 5-9]

● 예 : 삼목 게임

- ▷ 두 사람이 가로/세로 3×3 크기의 판에 수를 두어 한 행, 열, 또는 대각선을 모두 점유하면 이기는 게임

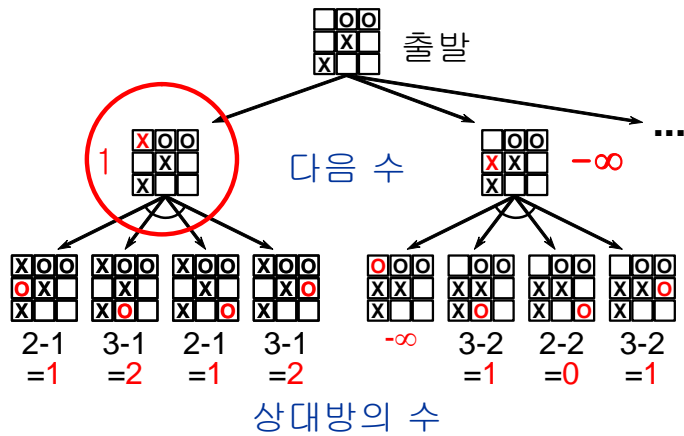
O		X
X	X	X
O		O

▷ 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

<table> <tr><td>X</td><td>O</td><td>O</td></tr> <tr><td>O</td><td>X</td><td></td></tr> <tr><td>X</td><td></td><td></td></tr> </table>	X	O	O	O	X		X			→	$F = 2 - 1 = 1$
X	O	O									
O	X										
X											
<table> <tr><td>X</td><td>O</td><td>O</td></tr> <tr><td></td><td>X</td><td></td></tr> <tr><td>X</td><td>O</td><td></td></tr> </table>	X	O	O		X		X	O		→	$F = 3 - 1 = 2$
X	O	O									
	X										
X	O										

▷ 다음 수의 결정



제8강 지식의 표현방법 (1)

▶ 주요 학습 내용

- 지식기반 시스템
- 지식의 표현
- 논리를 이용한 지식 표현
- 규칙을 이용한 지식 표현
- ☞ 교재 학습 범위 : 113~124쪽

▶ 학습 지침

지식의 사용은 인공지능 시스템에서 핵심적인 사항이다. 제8강에서는 지식 표현의 전반에 대하여 이해를 한다. 사람의 두뇌에 형성된 지식을 어떻게 정형화하여 표현하려 하는가의 관점에서 여러 가지 표현방법들을 학습한다.

1. 지식기반 시스템

● 지식기반 시스템(knowledge based system, KBS)

- ▷ 지능적 문제의 풀이는 궁극적으로 그 문제와 관련된 지식들을 어떻게 저장하고 이용할 것인가에 달려 있다 - 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 않다.
- ▷ 추론기관 : 주어진 문제의 상황에 대하여 지식 베이스 내의 지식을 이용해 추론함으로써 결론을 제시하는 역할을 한다.
- ▷ 사용자 인터페이스 : 사용자와 KBS와의 연결을 담당
 - 사용자가 지식 베이스에 저장된 지식을 접근하고 수정하는 데 도움을 주는 역할
 - 추론기관의 추론과정에 대한 설명 제공

2. 지식의 표현

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

▷ 역방향 사상 : 내부 지식 표현을 실세계의 표현으로 사상하는 것

※ 내부 표현방법이 실세계의 사실을 얼마나 근사하게 묘사하는가, 타당한 추론결과를 이끌어 내는 능력이 얼마나 우수한가에 따라 성능이 좌우된다.

(예) 실세계의 지식과 내부 지식 표현

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

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

● 지식의 표현방법들이 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 한다.
- ▷ 추론의 효율성 : 추론과정이 효율적으로 진행될 수 있어야 한다.
- ▷ 지식 획득 능력 : 새로운 지식을 쉽게 습득할 수 있는 능력 - 수동적 추가 ↔ 자동적인 학습능력

● 지식의 종류

- ① 문제영역에 대한 전문지식 : 해결하고자 하는 문제영역에서 문제와 관련된 사실들로부터 얻어지는 지식 - 구체적, 단편적인 지식들
- ② 대상세계에서 성립하는 경험법칙 : 문제해결 과정에서 지식 베이스의 지식을 사용하여 탐색하게 되는데, 효율적 탐색을 위해 대상세계에서 성립하는 경험적 지식이 사용될 수 있다.
- ③ 메타 지식(meta-knowledge) : 지식의 사용에 관한 지식
 - 어떤 규칙들이 다른 규칙들보다 우선적으로 적용하여야 할 조건을 제시
 - 특수한 문제의 풀이에 유용한 지식의 적용순서를 제시
 - 추론에 대한 설명을 통해 결과에 대한 신뢰감 제공

● 지식의 형태 : 절차적 지식 vs 선언적 지식

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 제어 정보는 지식 자체에 내포되어 있다.
- ▷ 추론의 적합성 면이나 지식 획득의 효율성 면에서 낮은 평가

2. 선언적 지식


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

3. 논리(logic)를 이용한 지식표현

● 술어 논리(predicate logic)

- ▷ 명제 : 참·거짓이 명백한 문장
 - 어떠한 객체와 그 객체를 수식하는 술어로 구성 - 술어와 객체를 분리하여 표현
- ▷ 논리 연산자 : AND, OR, NOT, 조건명제
- ▷ 기본적인 추론방식 : 연역법칙 사용
 - X 와 $X \rightarrow Y$ 가 참이라면 Y 가 참이라는 사실을 추론할 수 있다.

(예) 철수는 사람이다. $\Leftrightarrow man(\text{철수})$



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

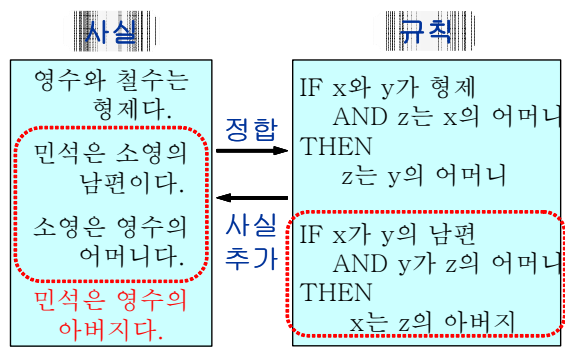
4. 규칙을 이용한 지식 표현

● 규칙

- ▷ 주어진 상황을 위한 권고·지시·전략을 나타내는 정형화된 표현방법
- ▷ 'IF ㉠ THEN ㉡' 형태의 표현
 - ㉠ : 가정, 전제조건, LHS
 - ㉡ : 결론, 실행할 동작, RHS
- ▷ 지식 베이스 : 현재의 상황을 나타내는 사실(또는 데이터 항목)들과 이에 대해 적용할 규칙들로 구성
- ▷ 추론기관이 현재 상태에 의해 만족되는 규칙을 선택하여 실행

● 지식 사용 예 : 전방향 추론

- ▷ 주어진 사실들로부터 만족되는 규칙을 규칙의 조건부와 정합에 의해 선택
- ▷ 선택된 규칙의 결론부의 내용을 실행하거나 사실에 추가
- ▷ '민석은 영수의 아버지이다'라는 사실의 추론



제9강 지식의 표현방법 [2]

주요 학습 내용

- 시맨틱 네트를 이용한 지식 표현
- 프레임을 이용한 지식 표현
- ☞ 교재 학습 범위 : 124~136쪽

학습 지침

시맨틱 네트와 프레임은 계층적 지식, 개념 사이의 관계 등을 표현할 수 있는 지식 표현 방법이다. 특성 상속의 활용, 절차적 지식의 활용 등을 강의에 제시된 예를 통해 학습하면 이해가 쉽다.

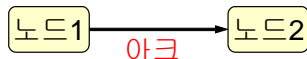
1. 시맨틱 네트를 이용한 지식 표현

※ 규칙을 이용한 지식 표현 방법은 간단하고 이해하기 쉬우나 다음과 같은 문제점이 있다.

- ① 구조화되어 있지 않다.
- ② 규칙의 개수가 증가하면 처리나 수정이 어렵다 - 규칙 상호간 종속관계, 규칙 사이의 상충 등이 발생

● semantic(어의에 관한, 의미론적인) network

▷ 네트워드를 기초로 한 지식 표현 방법

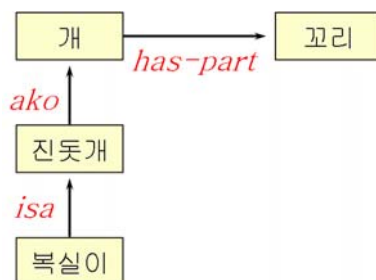


- ▣ 노드 : 객체, 개념, 사건 등을 표현
- ▣ 아크 : 노드들 사이의 관계를 표현

● 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, 원소 관계
 - ▷ *has-part* : 어떤 객체의 부속품을 나타내는 속성
- ※ 문헌에 따라서는 *ako* 아크를 *isa* 아크로, *isa* 아크를 *instance* 아크로 표기

(예)



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

● 특성 상속(property inheritance)

- ▷ 상위 클래스의 속성과 값을 하위 클래스 또는 사례가 이어받도록 하는 추론형태
- ▷ 상위 개념의 지식을 하위 개념이 공유하는 중앙 집중적 방법의 지식 표현
- ▷ 특성 상속이 이루어지는 아크 : *isa* 아크와 *ako* 아크

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

● 특성 상속 알고리즘 : 교재 127쪽

● 특성 상속의 장점

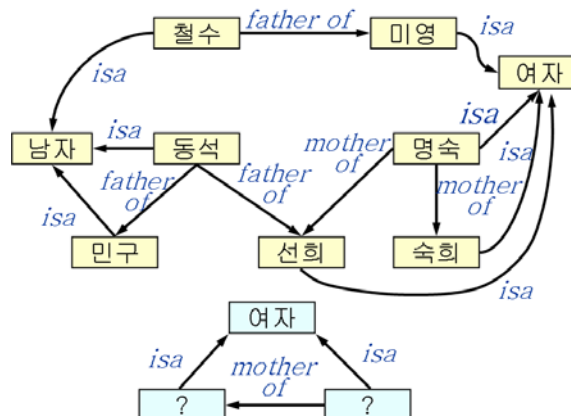
- ▷ 지식을 구성하기 쉽다 : 동일한 유형에 속하는 여러 개의 객체에 공통적인 속성을 개별적으로 저장할 필요가 없다.
- ▷ 표현된 지식의 오류를 쉽게 수정할 수 있다.
- ▷ 최신의 지식을 유지하기 쉽다.
- ▷ 지식의 분배가 자동적으로 이루어진다.

● 시맨틱 네트에서 지식의 검색

1. 활성화영역 전개

- ▷ 두 노드 사이의 연결관계를 파악하기 위한 추론방법
- ▷ 각각의 노드에 연결된 노드를 활성화하는 과정을 반복하여 겹치는 부분을 찾음.

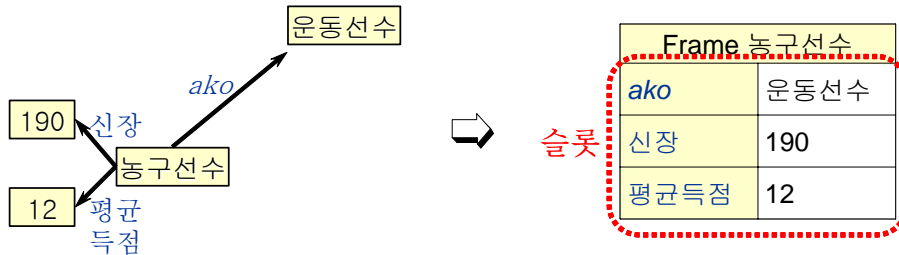
2. 정합



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

2. 프레임을 이용한 지식 표현

- ※ 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 객체들의 집합을 표현하는 프레임
- ▷ 부클래스 프레임 : 어떠한 클래스에 속하는 일부 객체들을 표현하는 프레임
- ▷ 사례(instance) 프레임 : 어떤 클래스에 속한 특정 객체를 표현하는 프레임
- ▷ 특성 상속

(예)

Frame 인간		Frame 성인남자		Frame 홍길동	
<i>ako</i>	동물	<i>ako</i>	인간	<i>instance</i>	성인남자
이동	직립보행	연령		연령	35
지능	(내정값=100)	키	(내정값=170)	키	175
		체중	(내정값=65)	체중	70
		결혼관계		결혼관계	기혼
		배우자		배우자	이영숙

● 부가 프로시저

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

(예) : 교재 132쪽, [그림 6-10]

- ‘체중’ 슬롯 : IF-needed(필요) 프로시저
 - 체중값이 지정되어 있지 않은데 값을 요구하면 동작
 - ‘연령’과 ‘키’ 슬롯의 값에 따라 계산
 - ‘연령’과 ‘키’ 슬롯의 값을 구할 수 없을 때에는 계산을 할 수 없으므로 내정값인 65를 ‘체중’ 슬롯의 값으로 사용
- ‘결혼관계’ 슬롯 : IF-written(기록) 프로시저

- 배우자 이름을 입력하도록 지시하여 '배우자' 슬롯의 값을 채우고, 그 값에 해당되는 프레임 내의 '결혼관계 및 '배우자' 슬롯을 수정하도록 메시지를 보낸다.

(예) FRAME '홍길동'의 '체중' 슬롯의 값 요구

- ⇒ '체중' 슬롯의 값이 지정되어 있지 않고, IF-needed 프로시저가 동작하기 위해 필요한 '연령' 및 '키' 슬롯의 값이 지정되어 있으므로 IF-needed 프로시저 동작
- ⇒ '연령' 슬롯의 값이 35보다 작으므로 키에서 110을 뺀 65라는 값이 제공됨.

Frame 홍길동		Frame 이영숙	
<i>instance</i>	성인남자	<i>instance</i>	성인여자
연령	29	연령	27
키	175	키	163
체중			
결혼관계			
배우자			

(IF-needed :

IF 연령 > 35 THEN

체중 ← 키 - 100

ELSE

체중 ← 키 - 110

end-IF)

※ 부가 프로시저를 첨부함으로써 절차적 지식을 포함시킬 수 있다.

※ 프레임의 특성

- ▷ 각각의 객체의 고유한 구조 및 행동을 캡슐화시킬 수 있다.
- ▷ 상속을 통해 일반적 개념을 수용할 수 있다.
- ▷ 복잡한 영역에서도 지식의 모듈 특성을 유지할 수 있고, 문제영역의 깊이있는 지식을 표현할 수 있다.

제10강 논리에 의한 지식표현 - 명제논리

▶ 주요 학습 내용

- 2진 논리의 개념
- 명제 논리식의 표현 및 이용
- 도출 연역
- ☞ 교재 학습 범위 : 137~148쪽

▶ 학습 지침

명제논리의 정형식 및 표준형, 특히 연언표준형은 도출 연역을 위해 숙지해 둔다. 추론의 기본적인 긍정식 및 부정식과 도출과정을 연결하여 학습한다.

1. 2진 논리의 개념

- ※ 논리학 : 인간의 사고과정에 대하여 연구하는 학문 분야
- ※ 형식논리(formal logic) : 문장의 내용보다는 형식을 통해 진위의 판단을 내리는 논리
 - ▷ 기호논리 : 명제 및 명제 간의 관계를 기호로 표현
 - ▷ 참 또는 거짓의 두 값을 다루는 2진 논리
 - ▷ AND, OR, NOT 등의 논리 연산기호를 도입
 - ▷ 수학적 도구 - 부울 대수
- ※ 퍼지 논리 - 다치 논리를 다루기 위한 도구

● 명제

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

● 기본명제

- ▷ 더 이상 분해할 수 없는 최소 단위의 명제

(예) p = 영수는 한국인이다.
 q = 존은 미국인이다.

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

● 복합명제(합성명제)

- ▷ 2개 이상의 명제를 논리 연산자로 결합한 것

(예)

p = 영수는 한국인이다. q = 존은 미국인이다.	→	$p \text{ AND } q$ =영수는 한국인이고, 존은 미국인이다.
-------------------------------------	---	--

● 논리 연산자

▷ \vee (OR), \wedge (AND), \sim (NOT)

☞ 완전집합

▷ \rightarrow (조건명제), \leftrightarrow (동치)

☞ \vee , \wedge , \sim 으로 표현할 수도 있지만 특히 의미있는 연산자이므로 별도로 고려

● 조건명제

▷ p 가 참인 경우에만 q 에 따라 조건명제의 값이 변화하고, p 가 거짓인 경우에는 q 의 값에 관계없이 참이다.

▷ $\sim p \vee q$ 와 동일한 진리표를 갖는다.

● 명제 논리식

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.

※ 논리 연산자 \sim 과 \rightarrow 만 언급했지만 \wedge 와 \vee 도 \sim 과 \rightarrow 의 조합으로 구현할 수 있기 때문에 포함된 것과 같다.

● 논리식의 표준형

▷ 명제기호와 논리 연산자를 조합하여 다양한 형태의 논리식이 결합될 수 있으나, 이를 표준화된 방법으로 표기할 필요가 있다.

연언 표준형	선언 표준형
$F_1 \wedge F_2 \wedge \dots \wedge F_i \wedge \dots \wedge F_n, n \geq 1$ $F_i = P_{i1} \vee P_{i2} \vee \dots \vee P_{im}$ P_{ij} : 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} : 리터럴

● 항진식 : 논리식을 구성하는 논리 변수의 값에 관계없이 항상 참인 논리식

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

▪ 진리표 : 교재 147쪽, [표 7-3]

● 추론방법

▷ 긍정식(modus ponens) : p 와 $p \rightarrow q$ 가 모두 참이면 q 도 참이다.

▷ 부정식(modus tollens) : $p \rightarrow q$ 와 $\sim q$ 가 사실이라면 $\sim p$ 이다.

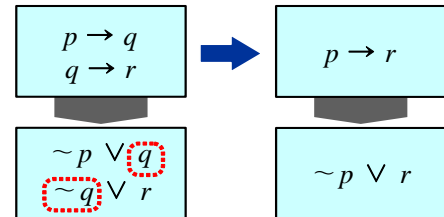
2. 도출 연역에 의한 정리 증명

※ 논리로 표현된 지식을 활용하는 대표적인 분야는 정리 증명

⇒ 도출연역을 사용

● 도출

- ▷ $p \rightarrow q$ 이고 $q \rightarrow r$ 이면 $p \rightarrow r$ 이다.
- ▷ 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 논리합으로 묶은 결론을 도출



● 기본 도출식

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

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

● 도출 연역에 의한 정리 증명 알고리즘

1. 증명하고자 하는 정리를 부정하여 공리의 리스트에 첨가
2. 공리 리스트를 연언 표준형으로 표현 후, 절 분리
3. 도출 가능한 쌍이 없을 때 까지 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

- ▷ CNF로 변환 후, 절(clause)들을 분리
- ▷ 증명하고자 하는 정리를 부정하여 공리 리스트에 첨가

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

● 도출 연역 과정 : 교재 156쪽, [그림 7-2]

- ▷ 증명하고자 하는 정리를 부정함으로써 *false*가 도출되며, 참인 공리로부터 유도된 결과가 거짓이 되므로 모순임.
- ▷ 따라서 증명하고자 하는 정리인 C 는 참임이 증명됨.

제11강 논리에 의한 지식표현 - 술어논리

▶ 주요 학습 내용

- 명제 논리의 한계 및 술어논리의 필요성
- 술어 논리식의 표현
- 술어 논리식의 도출 연역
- ☞ 교재 학습 범위 : 148~161쪽

▶ 학습 지침

술어, 객체, 객체변수, 함수 등의 기호를 활용하는 술어논리 정형식 표현방법을 학습한다. 술어 논리식의 절 분리 및 도출 연역 과정을 강의내용을 참조하여 직접 손으로 따라 해 보면 쉽게 이해할 수 있다.

1. 술어논리의 필요성

※ 명제논리에서 사실의 표현 : 명제기호

- ☞ 표현된 결과는 문장의 의미를 표현하는 것이 아니라 단지 기호일 뿐이다.
- ⇒ 기호로부터 의미를 구하여 추론할 수 없다.

(예) 명제 논리로 표현된 사실의 추론

p : 소크라테스는 사람이다. q : 플라톤은 사람이다. r : 모든 사람은 죽는다.	$\xrightarrow{?}$	u : 소크라테스는 죽는다. v : 플라톤은 죽는다.
---	-------------------	--------------------------------------

- 문장의 의미로 추론한다면 ‘소크라테스는 죽는다’, ‘플라톤은 죽는다’ 라는 사실을 얻을 수 있지만, p , q , r 이라는 기호로부터는 이러한 추론을 할 수 없다(\because p , q , r 은 단지 기호일 뿐이다).

● 술어 논리를 이용한 표현

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

(예) 소크라테스는 사람이다. $\rightarrow \text{Man}(\text{SOCRATES})$

플라톤은 사람이다. $\rightarrow \text{Man}(\text{PLATO})$

- ⇒ ‘소크라테스(SOCRATES)’와 ‘플라톤(PLATO)’이 모두 ‘사람이다(Man)’라는 공통적인 술어를 갖는다.
- ⇒ ‘모든 사람은 죽는다($\forall x\{\text{Man}(x) \rightarrow \text{Mortal}(x)\}$)’라는 형태로 표현된 명제에 의해 ‘소크라테스는 죽는다($\text{Mortal}(\text{SOCRATES})$)’와 ‘플라톤은 죽는다($\text{Mortal}(\text{PLATO})$)’를 추론할 수 있다.

2. 술어 논리식의 표현

● 1차 논리(1st order logic)

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

(예) 소크라테스는 사람이다. $\Leftrightarrow \text{Man}(\text{SOCRATES})$

X 가 Y 의 위에 있다. $\Leftrightarrow \text{On}(X, Y)$

- 객체의 표현 : 객체 상수, 변수, 함수가 사용될 수 있다.

(예) 변수의 사용

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

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

- 전칭기호(universal quantifier, \forall) : 모든 원소에 대하여 성립함.

(예) 깃털을 가지고 있는 것은 모두 새이다.

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

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

(예) 농구선수 중에는 키가 작은 사람도 있다.

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

※ 한정기호에 대한 등식

$$\textcircled{1} \sim(\exists x) P(x) \equiv (\forall x) \sim P(x)$$

$$\textcircled{2} \sim(\forall x) P(x) \equiv (\exists x) \sim P(x)$$

$$\textcircled{3} (\forall x) P(x) \wedge Q(x) \equiv (\forall x) P(x) \wedge (\forall y) Q(y)$$

$\Leftrightarrow P(x) \wedge Q(x)$ 는 $P(x)$ 와 $Q(x)$ 가 모두 참이어야 참이 되므로

$$\textcircled{4} (\exists x) P(x) \vee Q(x) \equiv (\exists x) P(x) \vee (\exists y) Q(y)$$

$\Leftrightarrow P(x) \vee Q(x)$ 를 만족하는 x 가 존재한다는 것은 P 를 만족하는 객체가 존재하거나 Q 를 만족하는 객체가 존재함을 의미한다.

$$\textcircled{5} (\forall x) P(x) \equiv (\forall y) P(y)$$

$$\textcircled{6} (\exists x) P(x) \equiv (\exists y) P(y)$$

- 함수기호 : 객체 사이의 관계를 나타내는 기호

(예) $\text{Korean}(\text{father}(\text{철수}))$

$\Leftrightarrow \text{father}$: 인수로 전달된 객체의 아버지에 해당되는 객체

● 술어 논리식의 정형식

- 항(term)

- ① 객체상수, 객체변수
- ② f (항₁, 항₂, ..., 항_n), f : 함수 기호
- ③ ①, ②에 의해 정의되는 것만이 항이다.

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

- ① 참(T), 거짓(F)
- ② P (항₁, 항₂, ..., 항_n) P : 술어 기호
- ③ P, Q 가 *wff*일 때 $\sim P, P \rightarrow Q$
- ④ P 가 *wff*이고, x 가 객체 변수일 때 $\forall xP, \exists xP$
- ⑤ ① ~ ④에 의해 정의되는 것만이 *wff*

3. 술어 논리식에 대한 도출 연역

※ 술어 논리의 도출 연역에서 고려할 사항

- 절(clause) 형태로의 변환
- 단일화(unification) - 변수의 사용

● 술어 논리식의 절 분리 과정의 예 : 교재 153쪽, [예제 1]

(예) 도출 연역에 의한 증명 예

※ 술어의 의미

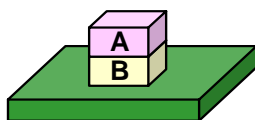
- $On(x, y)$: x 가 y 의 바로 위에 있음
- $Above(x, y)$: x 가 y 보다 위에 있음

□ 주어진 공리와 관측된 사실

공리

$$(\forall x)(\forall y)[On(x, y) \rightarrow Above(x, y)]$$

$$(\forall x)(\forall y)(\forall z)[Above(x, y) \wedge Above(y, z) \rightarrow Above(x, z)]$$

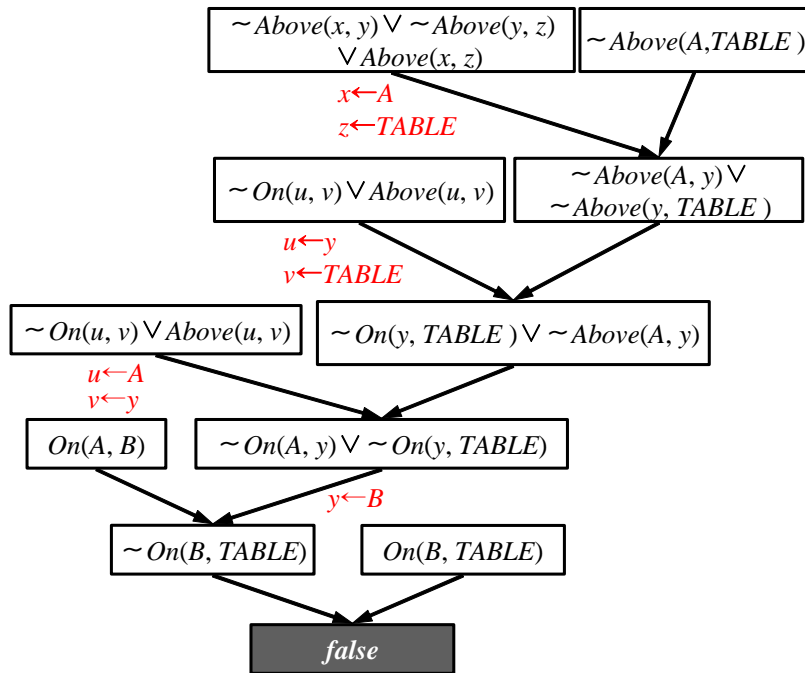


$On(A, B)$
 $On(B, TABLE)$

- 증명할 사실 : $Above(A, TABLE)$
- 공리를 절 형태로 변환 : ①~④
- 증명하고자 하는 정리를 부정하여 첨가 : ⑤

- ① $\sim On(u, v) \vee Above(u, v)$
- ② $\sim Above(x, y) \vee \sim Above(y, z) \vee Above(x, z)$
- ③ $On(A, B)$
- ④ $On(B, TABLE)$
- ⑤ $\sim Above(A, TABLE)$

□ 도출 과정



□ *false*가 얻어졌으므로 증명하고자 하는 사실 '*Above(A, TABLE)*'은 참임이 증명된다.

※ 전통적인 논리는 단조적(monotonic)

□ 기존 공리에 위배되는 지식을 수용할 수 없다.

(예) 모든 새는 날 수 있다. ↔ 펭귄은 날 수 없다는 사실이 첨가되면?

☞ 비단조 논리(nonmonotonic logic)

제12강 퍼지 논리에 의한 지식 표현

주요 학습 내용

- 퍼지 이론의 기본 개념
- 퍼지 집합의 개념 및 연산
- 퍼지 논리
- 퍼지 이론의 활용

교재 학습 범위 : 163~186쪽

학습 지침

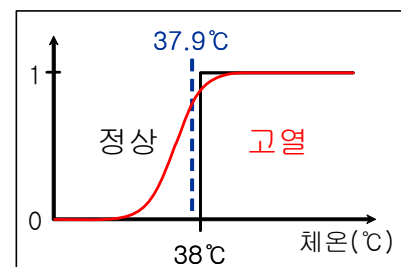
퍼지 집합은 소속함수로 집합의 소속관계를 표현하며, 0과 1로 소속함수값이 한정된 고전집합의 일반화로 생각하면 이해가 쉽다. 퍼지 집합 및 퍼지 논리의 각 연산에서 소속함수가 어떻게 계산되는가를 예를 통해 계산해 보면 이해가 쉽다. 고전집합 및 고전논리의 연산 특성이 유지되는 점을 확인해 보되, 예외사항에 주의해야 한다. 퍼지 추론의 경우 방대한 내용을 간략히 다루고 있어 이해가 쉽지 않을 수 있다. 전반적인 처리 과정 및 개념 중심으로 학습한다.

1. 퍼지 이론의 기본 개념

※ 컴퓨터는 0과 1로 표현되는 값을 다루는 2진 논리를 사용하나, 2진 논리로 접근하기 어려운 문제가 많다.

(예) 정상 체온과 고열의 판단

- 어떠한 판단작업을 2진 논리로 처리하는 일반적인 방법은 적절한 임계치를 주어 그 값을 기준으로 판단한다.
- 체온의 판단 : 38℃를 기준으로 이를 넘으면 고열, 그렇지 않으면 정상으로 판단
- 체온이 37.9℃이면?



※ 퍼지 이론 : 0과 1의 경직된 논리 추론을 탈피하여 보다 유연한 판단과정을 구현하기 위한 이론

※ 1960년대 초 미 캘리포니아 대학의 Zadeh 교수가 퍼지 집합의 개념을 제시

※ 1965년 Information and Control誌에 “Fuzzy Sets”라는 논문을 발표

※ 제어, 예측, 의사 결정, 전문가 시스템 등 많은 분야에서 활용

2. 퍼지 집합(fuzzy sets)

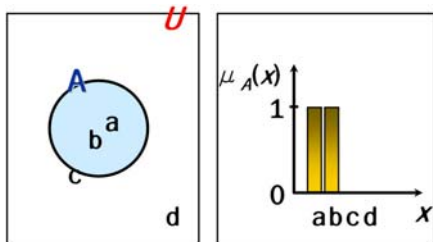
※ 퍼지 집합과 구분하기 위해 기존의 집합 이론을 따르는 집합을 고전집합이라 칭함.

● 소속함수 $\mu_A(x)$

□ x 가 집합A의 원소일 가능성을 나타내는 함수

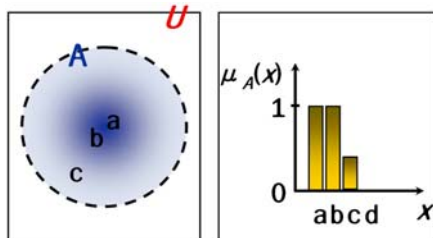
고전집합	퍼지집합
<p>□ x가 집합A의 원소이면 1 그렇지 않으면 0</p> $\mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$	<p>□ x가 집합A의 원소일 가능성을 0과 1사이의 값으로 표현</p> $\mu_A(x) \in [0, 1]$

● 고전집합 : 집합의 경계가 명백하여, 포함 여부가 참/거짓으로 명백하게 결정된다.



- a, b : 집합 A의 원소
- c, d : 집합 A의 원소가 아니다.
- a와 b의 소속함수값은 1, c와 d의 소속함수값은 0

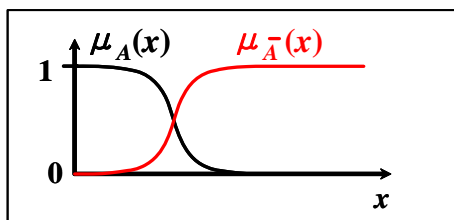
● 퍼지 집합 : 집합의 경계가 확실히 구분되지 않으며, 포함관계가 모호하다.



※ 퍼지 집합에 대한 연산 : 고전집합에서 사용되는 연산에 해당되는 집합 연산을 퍼지 집합에 대하여 정의

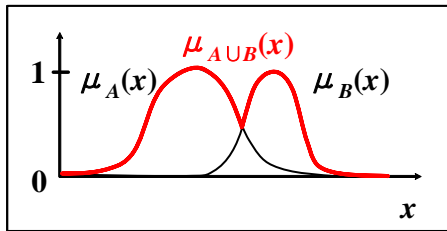
● 집합의 연산

□ 여집합



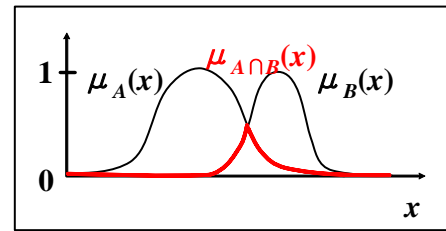
※ $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$

□ 합집합



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

□ 교집합



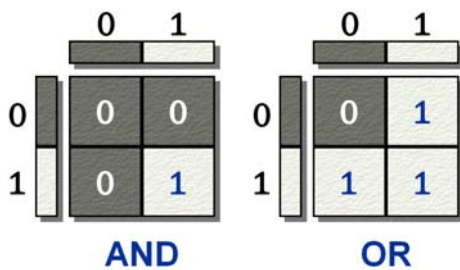
$$\ast \mu_{A \cap B}(x) = \text{Min}[\mu_A(x), \mu_B(x)], \forall x \in U$$

(예) 퍼지집합의 연산

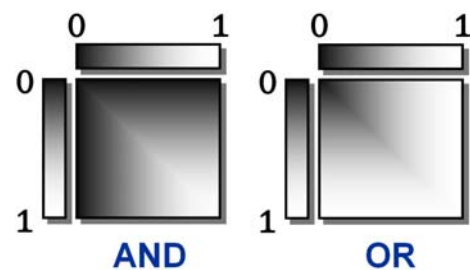
x	$\mu_A(x)$	$\mu_B(x)$	$\mu_{A \cup B}(x)$	$\mu_{A \cap B}(x)$	$\mu_{\bar{A}}(x)$
a	1	0	1	0	0
b	0.7	0.2	0.7	0.2	0.3
c	0.4	0.5	0.5	0.4	0.6
d	0	1	1	0	1

3. 퍼지 논리(fuzzy logic)

● 고전 논리 : 어떠한 명제의 참·거짓을 0과 1로 표현



● 퍼지 논리 : 논리 변수의 값을 0과 1 만으로 제한하지 않고, 0과 1 사이의 값으로 표현



● 퍼지 논리의 연산자

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

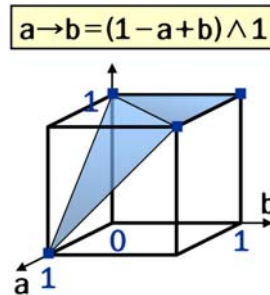
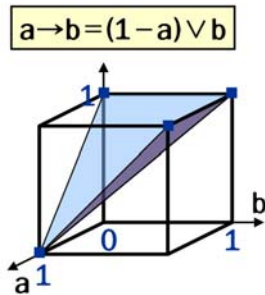
□ 퍼지논리 연산의 정의는 여러 가지 유형이 있으며, 위의 연산자 정의는 그 중 대표적인 한

예임

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

▪ 고전논리의 조건명제 $a \rightarrow b$ 의 4개의 진리값을 보간하는 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 같다.

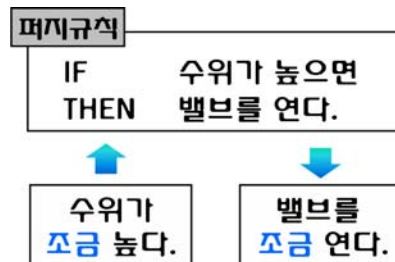
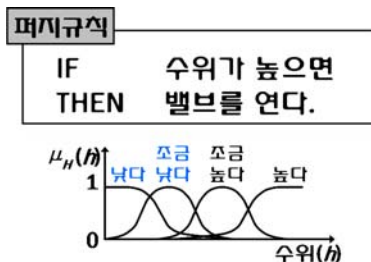


※ 차량 제어 시스템 : $a \rightarrow b$ 를 $a \wedge b$ 로 구현함으로써 보다 좋은 결과를 얻고 있다.

4. 퍼지 이론의 활용

● 퍼지 추론(근사 추론, approximate reasoning)

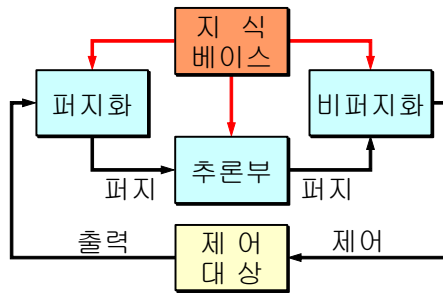
- 퍼지 이론을 이용한 추론
- 언어적 형태의 규칙을 사용 - 조건의 정도를 나타내는 수식어를 통해 융통성 있게 규칙을 적용할 수 있도록 함.
- 물탱크의 수위 조절을 위한 규칙의 예



- 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 값이다.

→ 퍼지 규칙을 활용

● 퍼지 제어기의 구조



● 차량 운전 제어 시스템의 예

□ 규칙의 예

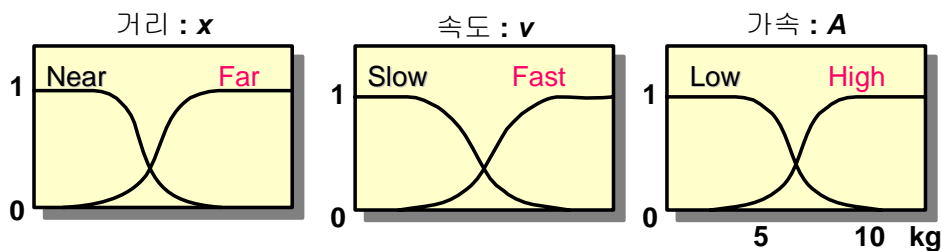
규칙 1

IF 거리가 멀다 *and* 속도가 느리다
THEN 가속을 높인다

규칙 2

IF 거리가 가깝다 *and* 속도가 빠르다
THEN 가속을 낮춘다

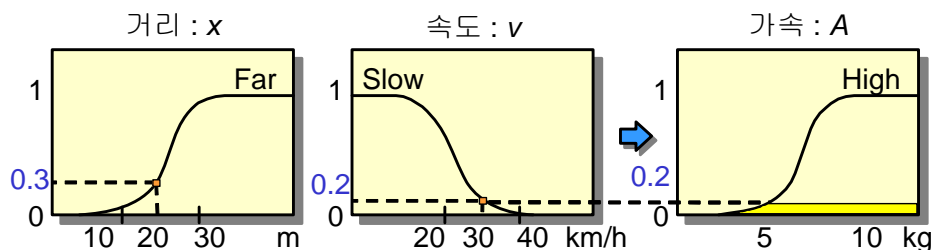
- 규칙의 조건부 및 결론부에 언어적 변수를 포함
- 일반적 규칙과는 추론절차가 다르다.
- 거리 및 속도에 따라 가속 정도를 제어
- 거리, 속도, 가속에 대한 퍼지 소속함수를 정의한다.



- 현재 앞 차와의 거리가 20m이고, 현재의 속도가 30km/h일 때, 얼마나 가속을 해야 하는가?

⇒ 추론과정

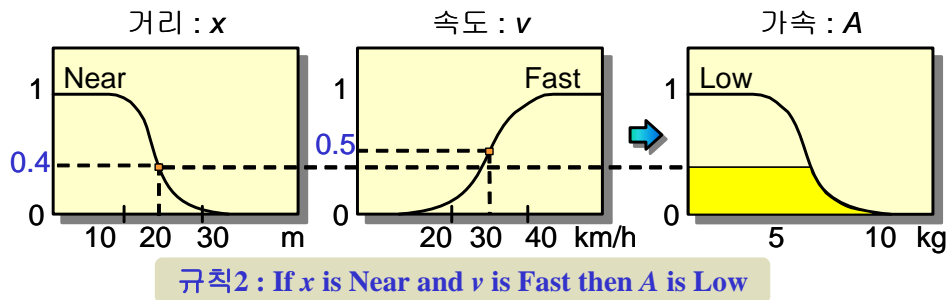
- 규칙1의 적용



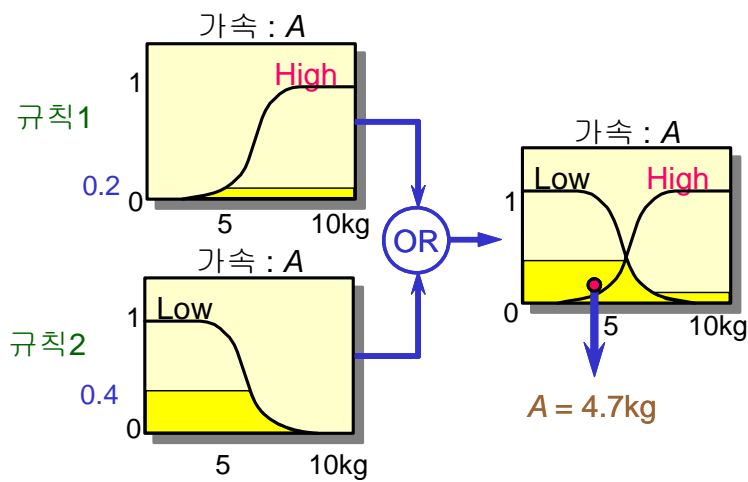
규칙1 : If x is Far and v is Slow then A is High

- 두 조건이 AND로 연결되어 있으므로 이들의 퍼지 논리값의 최소값을 구하고, 이 값으로 결론의 소속함수를 제한함

▪ 규칙2의 적용



▪ 결과의 종합



제13강 생성 시스템

▶ 주요 학습 내용

- 생성 시스템의 구조
- 추론방법

교재 학습 범위 : 187~208쪽

▶ 학습 지침

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 시스템

● 생성규칙(production rule) : 주어진 패턴에 따른 응답절차

▷ ‘IF 조건부 THEN 결론부’ 형태의 규칙

- 조건부 : 규칙이 선택되기 위한 조건
- 결론부 : 규칙이 선택되어 실행한 결과 제시하게 되는 결론 또는 행동

● 규칙 기반 시스템(rule-based system) : 문제 분야에서 적용될 수 있는 많은 규칙들을 저장하여 활용 - 생성 시스템(production system)

● 생성 시스템의 구성

- ① 생성 메모리 : 생성규칙들을 모아 저장하고 있는 부분
- ② 작업 메모리 : 현재의 상태를 표현하는 사실들의 집합
 - 외부로부터 입력되거나 추론에 의해 얻은 사실들이 저장
 - 작업 메모리의 내용은 상황에 따라 변화될 수 있는 것이기 때문에 단기 메모리(short-term memory, STM)이라고도 부른다.
- ③ 인터프리터 : 현재의 작업 메모리의 내용에 따라 규칙을 선택하고, 이를 실행하는 역할

※ 생성 메모리-프로그램을 저장하고 있는 부분, 작업 메모리-데이터를 저장하는 부분, 인터프리터-프로그램을 실행시키는 것으로 생각할 수 있다.

▷ 프로그램의 실행은 미리 주어진 알고리즘에 따라 이루어지지만, 생성 시스템의 동작은 현재의 상태에 따라 규칙이 선택되어 실행된다.

※ 인터프리터의 실행주기

- ① 정합(matching) : 생성 메모리 내의 많은 규칙들 중 작업 메모리의 내용에 의해 만족되는 것을 찾는다.
- ② 충돌해결 : 현재 상태에 의해 정합이 이루어지는 규칙이 여러 개 있을 수 있다. 이때 그 중 실행할 규칙을 선택하는 과정

③ 선택된 규칙의 실행

2. 추론

● 추론 방법

▷ 연역법(deduction)

- 'A'와 'IF A THEN B'로부터 'B'를 추론
- 정확한 전제로부터 정확한 결론을 추론할 수 있다.

▷ 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 있다(예 : 고래).

▷ 귀납법(induction)

- 공통적인 내용을 표현하는 사실들로부터 일반적 규칙을 찾아내는 추론 방법
- 항상 참인 규칙만을 만들어 내지는 못한다.
- 학습과 관련
- (예) 여러 가지 새에 대한 정보를 제공

제비는 날 수 있다.
 참새는 날 수 있다.
 독수리는 날 수 있다.

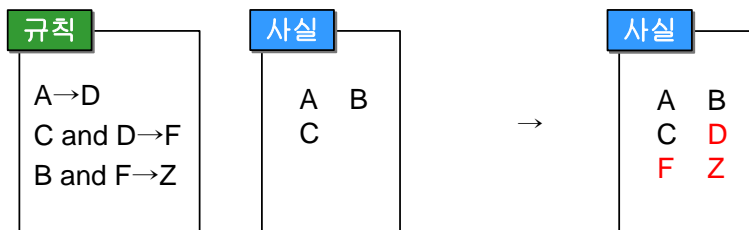
→ 새는 날 수 있다.

※ 규칙 기반 시스템에서 주로 사용하는 추론방법은 연역법

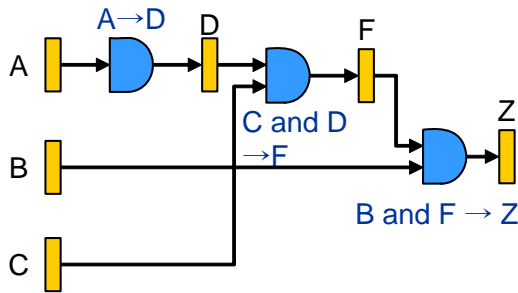
● 추론의 방향

- ▷ 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)를 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 참이다.

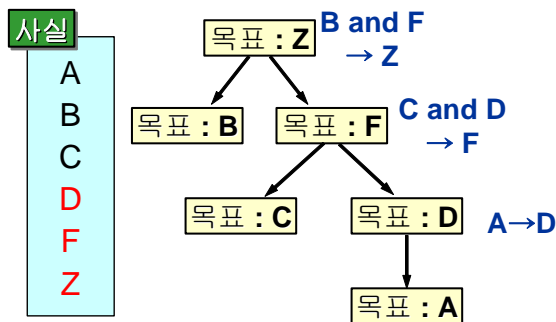
(예) 전방향 추론



⇒ 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 크다.
- ▷ 색인화
 - 규칙의 전제조건에 포함된 사실들을 이용해 색인을 구성
 - 규칙의 전제조건이 특정 상황의 명확한 기술인 경우 적합
- ▷ 필터의 사용
 - 규칙의 술어에 해당되는 특징을 정의
 - 작업 메모리의 변화에 의해 영향을 받는 규칙을 필터링

(2) 메타 지식의 사용

- ※ 메타 지식(meta-knowledge) : 지식의 사용에 대한 지식
- ▷ 어떠한 규칙의 집합을 선호하게 되는 조건을 제시 : 규칙 베이스의 규칙을 순차적으로 비교하게 하지 않고, 정해진 조건에서는 특정한 규칙집합을 탐색하도록 한다.

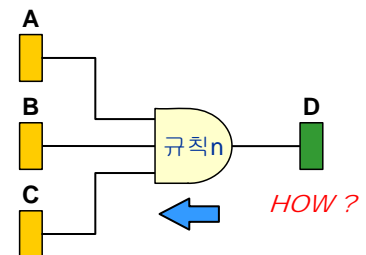
- ▷ 개별적인 문제풀이에 적합한 규칙의 적용순서 제시 : 규칙을 적용함으로써 얻게 되는 예상 효과, 다른 규칙들과의 상호작용 등에 대한 지식의 활용
- ▷ 어떠한 목표(또는 하부 목표)들을 풀이하는 순서를 제시 : 주어진 문제에 대한 하부 목표들이 있을 때 어느 것을 먼저 풀이해야 하는가에 대한 순서 제시

(3) 설명기능

- ▷ 사용자가 시스템에게 추론 과정에 대한 질문을 했을 때 응답을 할 수 있게 하는 기능
- ▷ 추론사슬을 보여 줌으로써 응답
- ▷ 시스템이 내린 결론에 대한 신뢰감을 제공한다.
- ▷ 추론과정을 보여 줌으로써 올바르게 규칙이 활용되고 있는지 확인하는 데 도움이 된다.

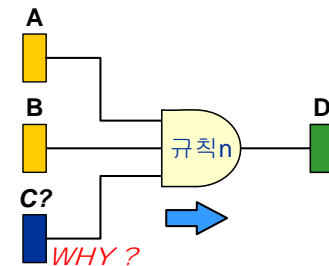
▷ HOW? 에 대한 설명

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



▷ WHY? 에 대한 설명

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



제14강 전문가 시스템

주요 학습 내용

- 전문가 시스템의 개요
- 전문가 시스템의 구조
- 전문가 시스템 개발도구
- ☞ 교재 학습 범위 : 209~234쪽

학습 지침

전문가 시스템은 인공지능 연구를 실용적으로 활용하기 시작한 분야이다. 지식기반 시스템의 전반적인 구조, 전문가 시스템 구축방법의 전반적인 사항을 학습한다.

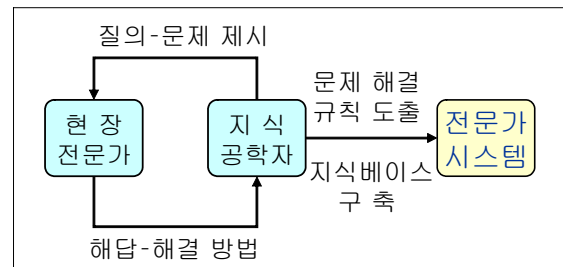
1. 전문가 시스템의 개요

● 전문가 시스템

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

● 전문가 시스템의 구축

- ▷ 현장 전문가(domain expert) : 문제 분야의 전문지식을 가지고 있는 전문가
- ▷ 지식 공학자(knowledge engineer) : 현장 전문가가 가지고 있는 전문지식을 체계적으로 지식 베이스화하는 역할을 담당



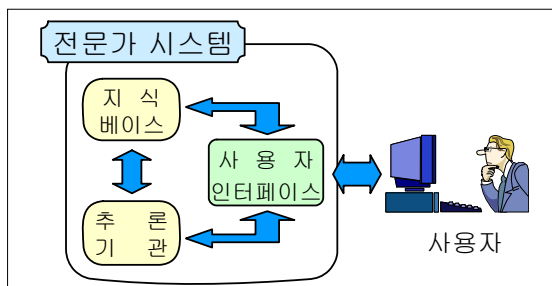
● 전문가 시스템의 장점

현장 전문가	전문가 시스템
한시적	영구적 ☞ 인간과는 달리 영구적으로 사용할 수 있다.
지식의 전달이 어렵다	지식의 전달이 쉽다. ☞ 한 사람의 전문가를 양성하기 위해서는 교육을 위해 많은 시간이 소비되나, 전문가 시스템은 프로그램이므로 쉽게 많은 복사본을 만들 수 있다.
문서화하기 어렵다.	지식을 문서화하기 쉽다.
일관적이지 않다.	문제 풀이 과정이 일관적이다. ☞ 신체적 상황 등에 따라 다른 결과를 내지 않는다.
많은 비용 소비	저렴하다.

● 전문가 시스템의 단점

현장 전문가	전문가 시스템
창조적	영감이 없다. ☞ 창조적 능력이 부족
적응적	새로운 상황에 적응하기 어렵다. ☞ 사람은 새로운 상황이 발생했을 때, 이에 대처함은 물론 그 경험을 바탕으로 새로운 지식을 축적할 수 있는 능력을 소유
복잡하고 다양한 입력형태 사용	정해진 기호로 입력되어야 한다. ☞ 사람은 다양한 형태의 입력을 받아들일 수 있으나, 전문가 시스템은 이를 기호화된 형태로 변환하여 입력해야 한다.
거시적 판단 기준	미시적 판단 위주 ☞ 거시적 판단 능력의 부족
상식적 지식 포함	전문적 지식만을 포함 ☞ 상식적 지식의 활용 능력이 부족

2. 전문가 시스템의 구조



● 지식 베이스

- ▷ 주로 규칙이나 프레임을 이용하여 지식 표현
- ▷ 사실 및 규칙을 저장
- ▷ 완전하지 않거나 확률적인 정보도 포함할 수 있다.
 - 경험적 지식

● 추론 기관

- ▷ 지식 베이스를 이용하여 문제를 풀이기 위한 일반적인 제어지식을 포함하고 있는 부분
- ▷ 추론기관의 구성
 - 규칙 해석기 : 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 용이하다.

3. 전문가 시스템 개발 도구

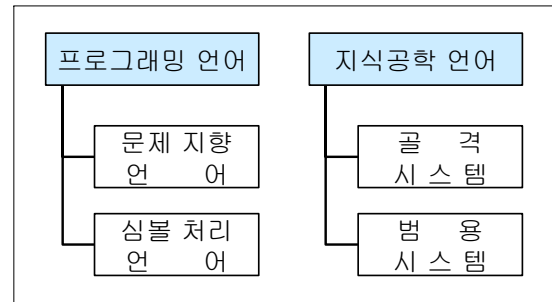
● 전문가 시스템 개발 도구

▷ 일반적인 프로그래밍 언어

- 문제 지향 언어 : PASCAL, C 등 일반적으로 사용하는 프로그래밍 언어를 이용한 구현
- 심벌 처리 언어(인공지능용 언어) : LISP, PROLOG 등 심벌 처리에 적합한 언어

▷ 지식 공학 언어 : 지식 기반 시스템 구성을 위해 제공되는 언어 - 전문가 시스템 셸(shell)

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



● 프로그래밍 언어를 이용한 지식 표현 예

▷ PROLOG

- 사실 : Alen은 Bob의 아버지이다.
⇒ father(alan, bob)
 - 규칙 : X가 Y의 아버지이면 X는 Y보다 나이가 많다.
⇒ older(X,Y) if father(X,Y)
- ☞ 추론기관을 포함하고 있어 이를 이용한 추론이 가능하다.

▷ LISP

- 사실 : Alen은 Bob의 아버지이다.
⇒ (Father Alan Bob)
 - 규칙 : X가 Y의 아버지이면 X는 Y보다 나이가 많다.
⇒ (IF (Father x y) THEN (Older x y))
- ☞ 심벌의 매칭, 추론과정 등을 직접 프로그램으로 구현해야 한다.

※ 프로그래밍 언어를 이용할 경우의 장·단점

- 장점 : 설계자가 프로그래밍에 따라 원하는 임의형태의 추론을 할 수 있다.
- 단점 : 지식의 표현, 추론방법 등에 대한 외부의 지원 없이 전적으로 시스템 개발자의 노력에 의해 만들어야 한다.

● 지식공학 언어 : 지식기반 시스템을 구축하기 위한 목적으로 만들어진 도구

▷ 골격 시스템, 범용 지식공학 언어

(1) 골격 시스템

- ▷ 이미 존재하는 전문가 시스템에서 지식 베이스를 제거하고 추론기관과 기타 지원 부분만으로 구성된 도구 (예 : EMYCIN)
- ▷ 골격 시스템의 장 · 단점
 - 장점 : 지식 베이스만 구축하면 시스템을 구성할 수 있다.
 - 단점 : 추론방법이 고정되어 있어 응용 분야에 맞는 추론방법을 자유롭게 구사하기 어렵다.

(2) 범용 지식공학 언어

- ▷ 지식공학을 위해 특별히 개발된 언어 (예 : OPS5, ROSIE 등)
- ▷ 골격 시스템에 비해 다양한 자료 접근, 탐색, 추론기법을 제공 - 특정한 응용에 적합한 틀에 기초하고 있지 않기 때문에
- ▷ 다양한 응용 분야에 활용할 수 있다.

제15강 에이전트

▶ 주요 학습 내용

- 에이전트의 개념
- 에이전트의 학습
- 에이전트 기술의 활용

교재 학습 범위 : 235~252쪽

▶ 학습 지침

에이전트가 갖는 특성, 에이전트의 전반적인 동작개념을 이해한다.

1. 에이전트란 무엇인가

● 에이전트의 정의

- “누군가를 위하여 정해진 일을 대신 해 주는 컴퓨터 프로그램을 통칭한 것”

● 에이전트의 특성

- 자율성 : 사용자를 위해 자율적으로 행동할 수 있다.
 - 허가를 받지 않고, 심지어는 통지하지도 않고 자율적으로 행동
- 적응성 : 스스로 배우거나 명시적으로 지시를 받는 과정을 통해 개별 사용자에게 맞게 적응할 수 있다.
- 능동성 : 단순히 주어진 환경에 반응하여 행동하는 것이 아니라 주도권을 가지고 목표 지향적으로 행동한다.
- 사회성 : 사용자 및 다른 에이전트와 상호작용을 할 수 있다.
- 지속성 : 일회성으로 주어진 작업을 실행한 후 종료하는 것이 아니라 지속적으로 실행하는 데몬(demon)과 같은 프로세스이다.

● 에이전트-환경 상호작용

- 에이전트는 자신이 속한 환경을 지속적으로 감지하여 상황에 맞는 적절한 행위를 함으로써 환경에 영향을 미친다(교재 238쪽, [그림 11-1]).
- 감지기 : 소속한 환경의 상황을 입력
 - ⇒ 환경의 상태를 인식
 - ⇒ 정해진 방식에 따라 조건을 만족하는 규칙을 선택
 - ⇒ 효과기를 통해 행위를 취함으로써 소속된 환경에 영향을 준다.

- 에이전트의 개략적 실행 사이클 : 교재 239쪽, [그림 11-2]

2. 에이전트의 유형

● 숙고형 에이전트

- 자신의 행동에 대한 결과를 예측하고, 그 예측과 목적을 고려하여 행동을 결정(교재 244쪽, [그림 11-7])
- 에이전트는 자신이 존재하는 세계에 대한 모델과 자신의 행동이 그러한 세계에 미치는 영향에 대한 모델을 가지고 있어야 한다.
- 일련의 행동들은 시뮬레이션을 통하여 안전하고 효과적이라는 것이 밝혀진 후 실행된다.

● 반응형 에이전트

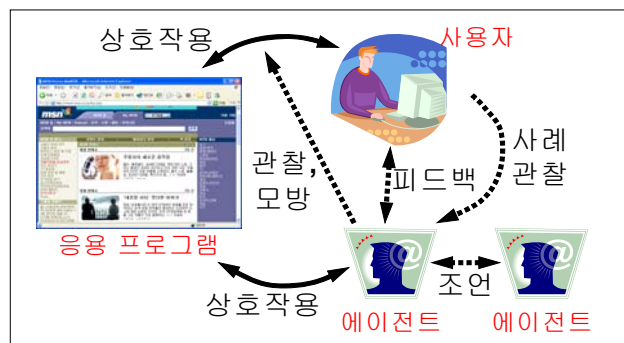
- 실세계에 대한 모델을 갖지 않음.
 - ⇒ 행동에 따른 결과, 변화된 상태를 예측하기 어려움.
- 실세계로부터 센서를 통해 감지한 신호 그 자체를 처리함으로써 곧바로 행동을 결정(교재 244쪽, [그림 11-3])
- 단순 반응형 에이전트
 - 어떤 내부 상태도 갖지 않고 단지 환경으로부터 감지된 현재 신호에 따라 행동을 결정하고 실행

3. 에이전트의 학습

● 에이전트는 사용자의 관심이나 기호를 학습하여 사용자 모델을 생성

● 학습방법

- ① 사용자 행위의 관찰을 통한 학습 : 에이전트는 오랜 시간 동안 사용자의 활동을 관찰하고, 그 행동을 기록하여 일관성이나 반복적인 패턴을 찾아냄.
 - 예 : 인터넷 쇼핑 사이트에서 로그인한 사용자의 구매 패턴을 관찰하여 상품을 추천
- ② 사용자의 피드백을 통한 학습 : 에이전트의 행동에 대한 직/간접적인 피드백을 활용
 - 예 : 에이전트의 추천에 대한 반응
- ③ 사례를 통한 학습
 - 예 : 메일 에이전트 - 특정 메시지(특정인으로부터 전달, 특정 제목 등)를 지정 폴더로 옮기도록 지시
- ④ 다른 에이전트의 충고를 통한 학습 : 동일한 유형의 작업을 하는 다른 사용자의 에이전트에게 질의하여 충고를 받아들임.



4. 에이전트 기술의 활용

● 웹 에이전트

- 사용자의 요구와 취향을 반영하고, 관심사항을 학습하여 효율적이고 편리한 웹 환경을 제공
- 예 : 인터넷 쇼핑물 - 로그인한 사용자의 취향, 관심 분야에 따라 적절한 상품을 추천

● 시맨틱 웹

- 컴퓨터가 정보 자원의 의미를 이해하고, 논리적 추론을 통해 자체적으로 필요한 일을 처리할 수 있는 차세대 지능형 웹
- 컴퓨터가 정보의 의미를 이해할 수 있도록 온톨로지 기반의 메타데이터를 작성
 - ※ 온톨로지 : 단어와 관계들로 구성된 일종의 사전
 - 그 속에는 특정 도메인에 관련된 단어들이 계층적으로 표현되어 있고, 추가적으로 이를 확장할 수 있는 추론규칙이 포함되어 있음.
- ⇒ 컴퓨터가 스스로 사람의 취향과 요구에 따라 스스로 전자상거래를 하거나 일정과 선호도에 맞게 여행일정을 결정하고 필요한 예약을 하는 등의 작업을 에이전트를 통해 할 수 있게 한다.

● 오락 산업

- 게임을 반복해서 즐기다 보면 게임 내의 캐릭터의 위치나 행동을 익히게 됨.
 - ⇒ 게임의 등장인물들을 에이전트로 표현하여, 주어진 환경에 따라 변화를 인식하여 학습함으로써 자율적으로 추론하여 행동할 수 있도록 함.
- ※ 기타 의료 산업, 정보 관리, 모바일 응용 등

제16강 패턴 인식 (1)

▶ 주요 학습 내용

- 패턴인식 개요
- 패턴인식 시스템

교재 학습 범위 : 253~265쪽

▶ 학습 지침

패턴 인식은 인간의 인식 능력을 컴퓨터에 구현하는 실용적인 연구 분야이다. 제16강은 패턴 인식을 수행하기 위한 처리단계를 학습한다. 이 처리단계의 세부 사항은 응용 분야에 상당히 의존적이지만, 각 단계마다 어떠한 처리가 이루어지는지 개괄적으로 이해한다.

1. 패턴 인식이란

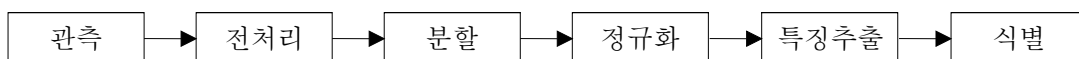
● 패턴 인식

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

● 패턴 인식 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 디지털 값으로 변환하기 위한 장치
- 입력된 패턴을 가공하고 식별하기 위한 소프트웨어가 탑재된 컴퓨터 시스템

2. 패턴 인식의 처리 단계



- 관측 : 대상 패턴을 입력 장치를 통하여 취득하는 단계
- 전처리 : 입력된 데이터를 가공하여 보다 처리하기 적합하게 만드는 작업
- 분할 : 입력된 신호로부터 인식대상 패턴만을 분리하는 과정
- 정규화 : 표준화된 형태로 패턴을 변환
- 특징 추출 : 실제 인식에 사용될 데이터를 뽑아내는 단계
- 식별

※ 처리의 순서는 응용대상에 따라 차이가 있을 수 있다.

2. 1. 관측

● 패턴의 관측

□ 카메라, 마이크, 센서 등 대상에 맞는 입력장치를 통해 대상물을 전기적 신호로 변환

⇒ 아날로그 신호를 디지털 신호로 변환(A/D 변환)

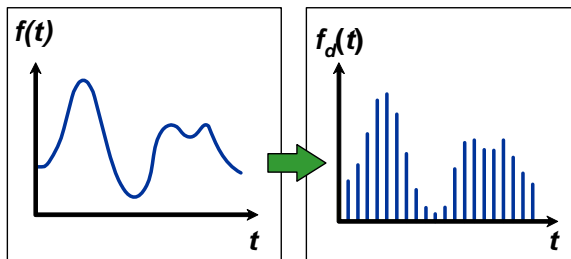
↳ 표본화 및 양자화

● 표본화(sampling)

□ 컴퓨터는 이산적 값을 대상으로 처리

⇒ 연속적인 신호를 이산적인 신호로 변환하는 것

(예) 1차원 신호의 표본화



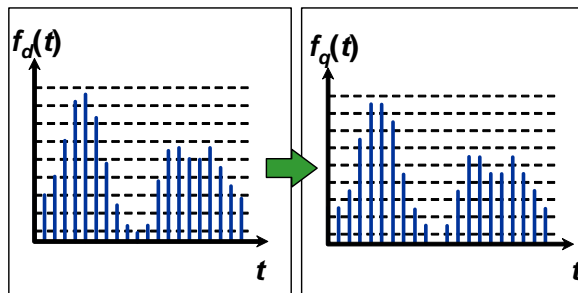
※ 나이퀴스트 간격보다 조밀하게 표본화하면 원래의 신호를 완전히 복원할 수 있다.

● 양자화(quantization)

□ 개개의 표본값을 디지털 값으로 취득

□ 어떠한 값을 유한 개의 구분된 구간으로 나누어 대표값으로 표현하는 것

(예) 1차원 신호에서의 예



□ 값의 표현구간을 나눈다.

□ 각각의 구간에 대해 대표값을 부여

※ 양자화 오차 - 일정 범위의 값을 하나의 대표값으로 표현하기 때문에 차이가 발생

※ 표본화 및 양자화의 정도는 원 데이터의 보존과 처리용량을 고려한 선택이 필요하다.

2. 2. 전처리(Preprocessing)

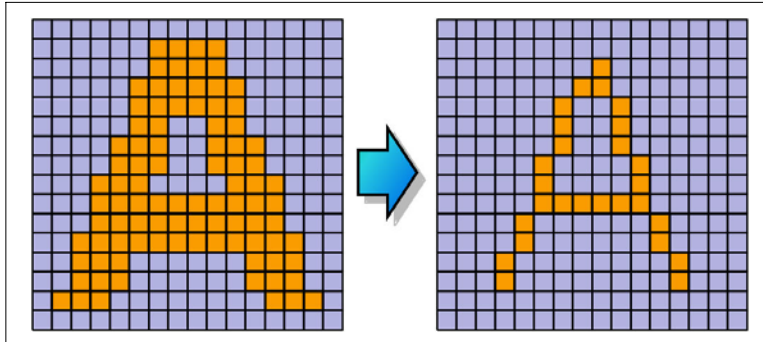
● 전처리 : 잡음 제거, 신호의 개선 등

(예) 필터를 사용한 잡음 제거

- 중앙값(median) 필터, 평활화(smoothing) 필터 등
- 잡음을 제거함으로써 인식하고자 하는 패턴을 정확하게 추출할 수 있다.

(예) 세선화

- 패턴 구성 요소의 연결성을 잃지 않는 범위에서 불필요한 화소를 제거
- 원래의 패턴으로부터 추출하기 어려운 구조적 정보를 추출하기가 용이하다.



2. 3. 분할

- 분할 : 인식하고자 하는 대상 패턴이 다른 패턴들이나 배경과 함께 있을 때, 인식대상 패턴을 분리해 내는 작업
 - 대상 패턴이 고립되어 있을 수도 있고, 다른 패턴과 이어진 연속적인 패턴일 수도 있다.

2. 4. 정규화

- 정규화
 - 패턴 인식을 하기 위해서는 학습과정을 통해 대상 패턴에 대한 정보를 취득
 - 학습할 때와 동일한 조건에서 판단할 수 있는 정보를 취득하기 위해 정규화
 - 크기, 진폭, 위치 등

2. 5. 특징 추출

- 특징
 - 인식대상의 패턴들 중에서 어떠한 패턴을 다른 패턴들과 구분하기 위해 사용될 수 있는 정보
 - 심벌 형태 또는 수치적 데이터
- 특징 추출의 목적
 - 패턴 공간의 차수를 낮춘다.

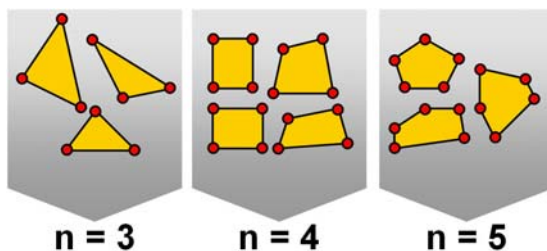
- 판별작업에 더욱 적합하도록 의미있는 값을 얻는다.

● 특징 선택의 조건

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

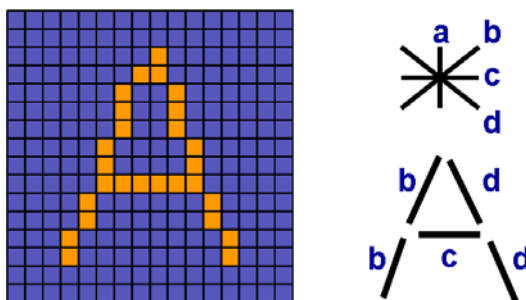
(예) 다각형의 식별

- 대상 패턴이 단순히 다각형만을 대상으로 식별하는 경우에는 불필요한 많은 영상 데이터를 사용할 필요없이, 꼭지점 수를 검출하여 이를 특징으로 삼으면 쉽게 식별할 수 있다.



(예) 세션화된 문자 패턴에서 획 특징의 추출

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



2.6. 식별

● 식별 : 입력된 패턴이 학습된 패턴 중 어느 것에 속하는가를 판별하는 단계

● 식별방법의 종류

- 결정이론적 방법 : 각각의 패턴에 대한 특징값들의 확률분포 형태를 분석하여 만든 식별함수를 통해 구분
- 언어이론적 방법
 - 원시 패턴을 정의
 - 원시 패턴들이 어떠한 규칙에 따라 나열되어 하나의 패턴을 구성한다고 가정
 - 원시 패턴의 나열규칙을 문법으로 정의하여 이에 따라 구문 분석을 함으로써 패턴을 인식
- 신경회로망, 퍼지 이론 등을 통한 방법

제17강 패턴 인식 (2)

▶ 주요 학습 내용

- 결정이론적 패턴 인식 방법
- 언어이론적 패턴 인식 방법

☞ 교재 학습 범위 : 266~282쪽

▶ 학습 지침

결정이론적 패턴 인식 방법에 대해서는 특징공간과 특징공간에서의 결정경계를 어떻게 구하는가에 관심을 가지고 학습한다. 거리측정자는 특징공간상의 좌표 사이의 거리를 구하는 기준이므로 기본적 내용이다. 시간상 강의에서는 다루지 않았지만 교재를 통해 학습하기 바란다.

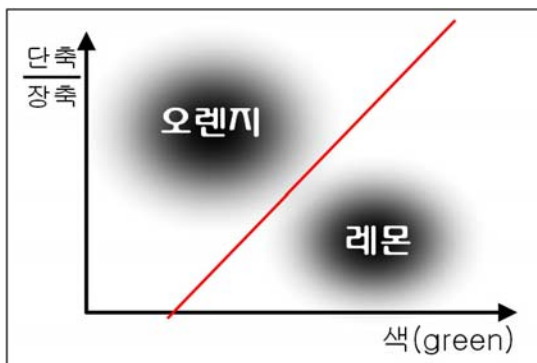
1. 결정이론적 패턴 인식 방법

※ 어떠한 패턴은 특징 추출 과정에서 구해진 특징으로 표현

● 특징공간 : 패턴을 구성하는 특징요소들을 좌표축으로 하여 형성된 공간

(예) 오렌지와 레몬의 식별

- 특징 : 단축과 장축의 비, 색
- 특징공간



□ 패턴의 인식 : 특징공간의 두 그룹을 분리하는 함수를 정의하여 인식

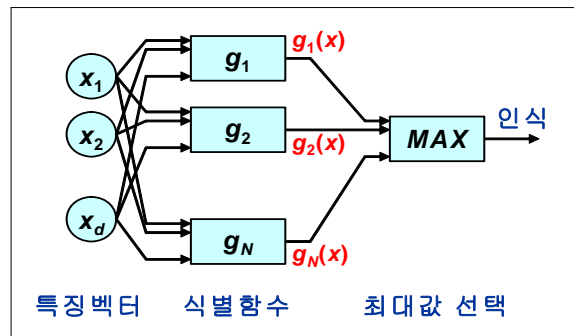
☞ 장·단축 비와 색의 함수

● 결정이론적 패턴 인식 방법

- 패턴을 구성하는 특징값들의 통계적 분포를 이용하여 인식하는 확률통계적 방법
- 특징값의 형태 : 수치값
- 각각의 패턴은 각각의 특징을 한 축으로 하는 특징공간상의 한 점으로 표현
- 특징공간상에 분포하는 각 패턴 그룹들의 경계를 찾아, 그 경계를 기준으로 패턴을 식별한다.

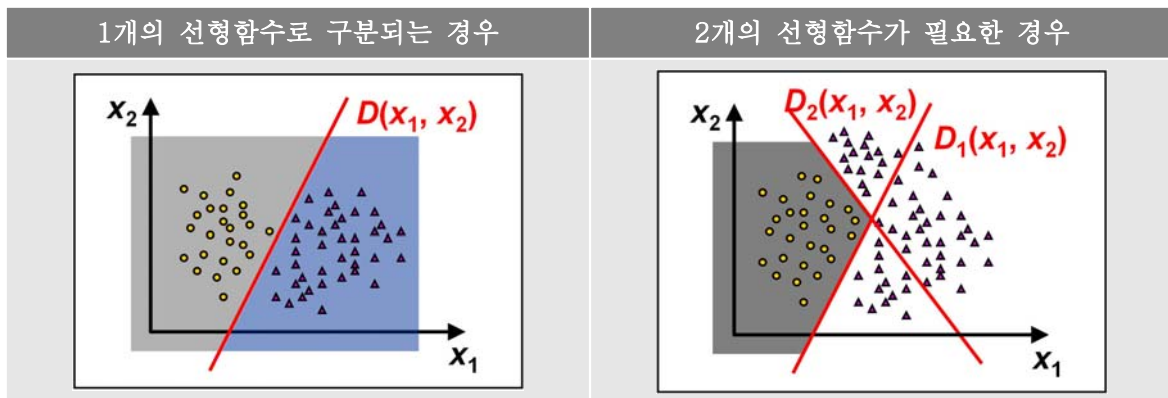
● 패턴 인식기의 개념도

- 전체 패턴을 d 개의 특징으로 표현
 $\Rightarrow [x_1, x_2, \dots, x_d]$ 로 표현되는 특징 벡터
- 구분 하고자 하는 패턴의 종류 : N 가지
 \Rightarrow 해당 부류에 속할 가능성이 높을수록 큰 값을 내도록 정의된 N 개의 식별함수
- 식별함수의 값이 가장 큰 부류로 인식



● 식별함수

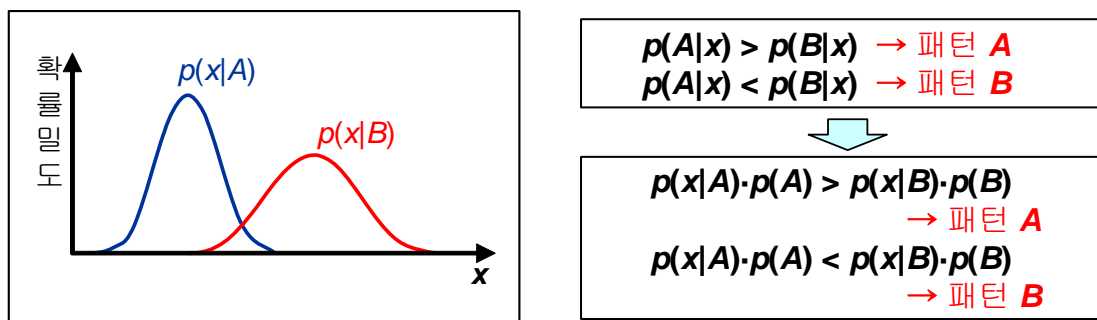
- 특징공간상에서 서로 다른 패턴들 사이의 경계를 구분하는 함수
- 1개의 선형함수로 표현될 수도 있고, 비선형함수 또는 여러 개의 선형함수의 결합에 의해 정의될 수도 있다.



※ 패턴 인식기의 구성 : 특징 벡터를 입력으로 하는 식별함수를 정의하는 것

- 학습 : 표본 패턴을 제시하여 식별함수를 결정하는 과정

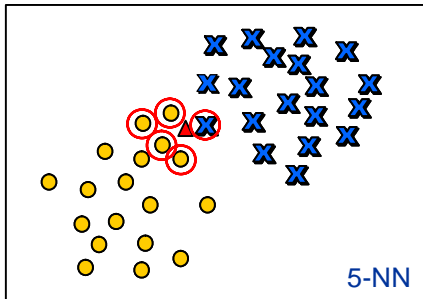
※ A 와 B 라는 2개의 패턴 집합을 x 라는 특징을 이용하여 식별하는 경우의 예



- 패턴 A 와 B 에 대한 표본들로부터 얻은 확률밀도함수 : $p(x|A)$ 와 $p(x|B)$
- 미지의 패턴으로부터 추출한 특징이 x 일 때, 그 패턴이 속하는 클래스의 결정
- ※ $p(A|x)$, $p(B|x)$ 를 구하는 방법 : k -근접이웃(k -NN), 파슨 창 등

(예) k -NN(nearest neighbor)

- 미지의 패턴과 가장 거리가 가까운 k 개의 표본을 취한다.
- 선택된 표본 중 해당되는 표본이 가장 많은 클래스에 속하는 것으로 판정
- 예 : 5-NN



⇒ 미지의 패턴 ▲와 가장 가까운 표본 5개를 선택한 결과 ●가 4개로 가장 많으므로 ▲는 ●에 해당되는 클래스에 속하는 것으로 판정한다.

● 결정이론적 패턴 인식 방법의 특징

- 확률이론을 토대로 하고 있다.
- 특징 벡터에 약간의 잡음이 포함되어도 이에 잘 대처할 수 있다.
- 단점 : 구조적 특징을 이용하기 어렵다.

2. 언어이론적 패턴 인식 방법

※ 언어이론적 패턴 인식 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
- 원시 패턴(primitives) : 패턴을 구성하는 기본 요소

● 언어이론적 패턴 인식 시스템 (교재 274쪽, [그림 12-17])

- 패턴의 분석 : 표본 패턴으로부터 원시 패턴 및 추출 방법을 정의하고, 이를 분석하여 문법을 정의 - 학습과정
- 인식과정
 - 입력 패턴으로부터 원시 패턴을 추출
 - 정의된 문법에 따라 구문 분석을 하여 받아들여지는 패턴으로 인식

● 패턴의 정의 : 문법

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

V_N : 비단말 기호

V_T : 단말기호 - 원시 패턴

P : 생성규칙

S : 시작기호 - 패턴

● 언어이론적 패턴 인식 방법의 특징

- 패턴의 구조적 해석에 의한 인식
- 패턴의 구조체계를 유지하는 범위 내에서 패턴의 변형 수용
- 원시 패턴 추출이 어렵다
- 생성규칙 자동 생성이 어렵다.

제18강 자연어의 이해

주요 학습 내용

- 자연어 이해의 문제점
- 자연어 이해의 처리과정
- ☞ 교재 학습 범위 : 284~300쪽

학습 지침

컴퓨터를 통해 언어의 의미 모델을 표현하는 측면에서 자연어의 특성을 이해하고, 문장에 의미를 부여하기 위한 방법들을 개념 중심으로 학습한다.

1. 자연어 이해의 문제점

● 자연어 이해의 문제점

(1) 목적 표현의 복잡성

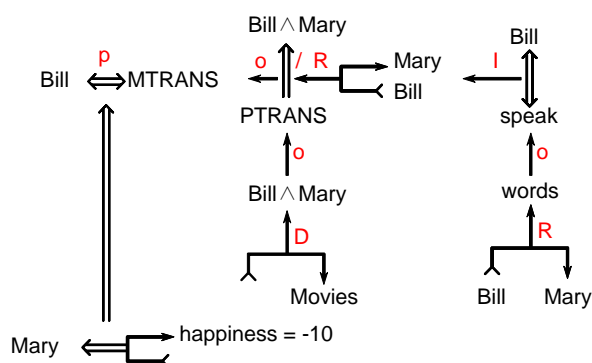
- 문장을 이해하는 것은 주어진 문장의 의미를 담고 있는 적절한 내부 표현 방법으로 변환하는 것
- 이해하고자 하는 수준에 따라 목적 표현이 단순할 수도 있고, 때로는 매우 심층적인 표현이 되어야 하는 경우도 있다.

(예) 핵심단어를 이용한 자료검색

- 문장 : “지난 번 대통령 선거에 대한 모든 자료를 검색하라.”
- 문장에서 얻고자 하는 것은 그 문장에 포함되어 있는 핵심단어
⇒ (SEARCH KEYWORDS = 선거 ∧ 대통령)

(예) 개념 의존도 모형을 이용한 표현

문장 : Bill은 Mary에게 함께 극장에 가지 않겠다고 말했다. 그녀는 기분이 상했다.



※ 참고

⇔ : 상태가 변환

^P ⇔ : actor와 actor가 행한 행동을 나타내는 관계

^O → : 사건과 객체의 관계

^R ⇔ : 행동과 그 행동을 한 사람 및 받는 사람의 관계

^I → : 행동과 도구의 관계

^D ⇔ : 물리적 source와 destination의 관계

MTRANS : 정신적 정보의 이동

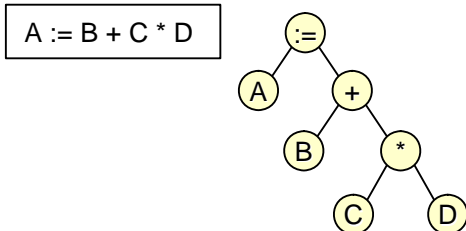
PTRANS : 물리적 장소의 이동

※ 응용에 따라 목적 표현의 복잡도를 선택

(2) 다양한 변환의 유형 : 1:1, 1:n, m:1, m:n

□ 일 대 일 변환

- 하나의 문장이 하나의 목적 표현과 대응
- 프로그래밍 언어



□ 다수 대 일 변환 (m:1)

- 여러 개의 문장이 동일한 목적 표현과 대응되는 경우

✦ 지난번 대통령선거에 대한 모든 것을 말해 주시오.
 ✦ 나는 지난번 대통령 선거에 대한 모든 것을 알고 싶습니다.
 ✦ 나는 지난번 대통령 선거에 대해 관심이 있습니다.

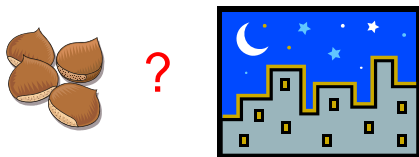


keyword = 선거 ∧ 대통령

□ 일 대 다수 (1:n)

- 하나의 문장을 여러 개의 목적 표현으로 해석될 수 있는 경우
- “나는 밤이 좋다.” - 어떤 ‘밤’을 의미하는가?

나는 밤이 좋다.



- “그는 긴 코끼리의 코를 보았다.” - 형용사 ‘긴’이 ‘코끼리’를 수식하는가, ‘코’를 수식하는가?

그는 긴 코끼리의 코를 보았다.



- 언어 자체가 가지고 있는 모호함에 기인하는 문제임 - 문맥에 따라 해석 가능

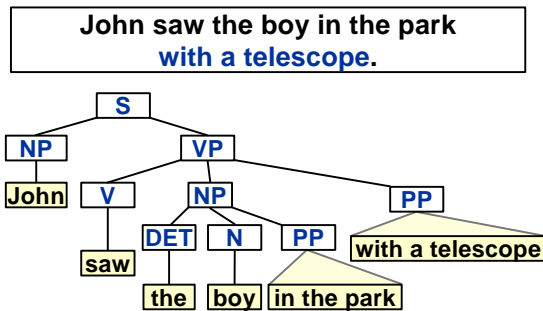
※ 자연어의 경우 m:1, 1:n이 모두 가능

⇒ m:n이 모두 가능

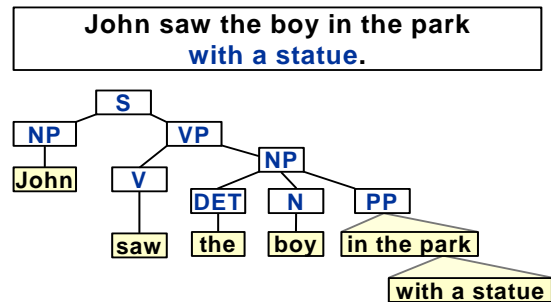
(3) 요소들 간의 상호작용 정도

- 문장을 구성하는 각각의 요소가 다른 요소들과 관계없이 변화될 수 있다면 변환과정의 부담을 크게 줄일 수 있다. 자연어의 경우 이러한 상호작용이 발생하여 1개의 단어만 바뀌어도 문장의 구조가 영향을 받을 수 있다.

(예) ‘telescope’라는 전치사 절의 명사가 ‘statue’라는 명사로 바뀔에 따라 그 전치사 절의 용도가 바뀐다.



⇒ 도구 - 망원경을 가지고 보았다.



⇒ 동상이 있는 공원

(예) 한글문장에서의 예

- 문장 : ‘나는 **학교로** 간다.’

⇒ 학교 : 목적지

- 문장 : ‘나는 **자전거로** 간다.’

⇒ 자전거 : 이동수단

※ 목적 표현의 복잡성, 변환의 유형, 요소들 간의 상호작용 정도 등 응용 분야에 따라 고려해야 하는 정도가 차이가 있다.

2. 자연어 이해의 처리과정

- 형태 분석 : 각각의 단어, 구두점 등을 분리시킨다.
- 구조 분석 : 문장 구성 요소들의 문법적 구조를 분석하여 적합한 모형을 찾는다.
- 어의 분석 : 문장 요소들의 의미를 파악한다.
- 문장 통합 : 인접 문장의 의미에 따른 해석
 - 한 문장의 의미가 이전 문장의 의미에 영향을 받을 수 있다.
 - 예 : 철수는 영희에게 선물을 주었다. 그녀는 매우 기뻐했다.
⇒ 그녀 = 영희
- 실용 분석 : 문장의 내용을 실제 의미하는 바에 따라 재해석
 - 예 : 지금 몇 시인지 아십니까?
⇒ 시간을 알려 달라는 뜻으로 해석

2.1. 구조 분석

● 구조 분석

- 문법 및 파서
 - 문법 : 비단말 호, 단말기호, 시작기호, 생성규칙
 - 파서 : 주어진 문장을 문법에 따라 분석
- 파싱 과정에서 주어진 문장이 문법에 맞는 구조를 취하고 있는가 판단
- 문법에 맞는 경우 그 문장의 구조모형을 부여

※ Context-free 문법이 자연어에 적합한가의 여부에 대해서는 많은 논란이 있음.

- Chomsky는 형식문법을 4가지로 분류 : 정규 문법, context-free 문법, context-sensitive 문법, 무제한 문법
- 자연어는 context-sensitive 언어라는 것이 일반적인 견해
- 자연어의 자동 처리를 위해서는 context-free 언어로 간주하는 것이 편리하나, 이 경우 설명하지 못하는 문장구조가 존재
- Context-sensitive 언어로 간주하면 경우의 수가 매우 많아 체계적으로 형식화하기가 어렵다.

※ 시스템에 따라서는 파싱을 하지 않는 경우도 있으나, 파싱을 할 경우

- 파싱 결과에 대해서만 어의를 고려하면 되므로 효율 증가
- 파싱을 하지 않고는 올바른 해석이 어려운 문장이 존재

2.2. 어의 분석

● 어의 분석

- 문장의 구조모형에 대해 의미를 할당하는 과정
- 하나의 문장에 대해 여러 가지의 구조 모델이 가능할 경우 문맥에 따른 제한을 통해 적절한 것을 선택할 수 있다.

● 단어 이해

- 사전으로부터 단어를 찾아 그 의미를 얻는 과정

※ 동일한 단어가 여러 가지 뜻을 가지고 있는 경우가 많이 있다.

(예)

단어 : 배
나는 지금 배가 부르다.
이 배는 참 맛있다.
나는 배를 타고 간다.

- ‘배’의 의미
 - 몸의 일부분
 - 과일
 - 수상 운송수단

단어 : diamond
I dropped my diamond.
I'll meet you at the diamond.

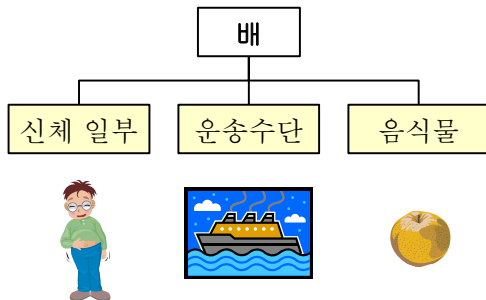
- ‘diamond’의 의미
 - 보석
 - 야구 경기장

● 단어 의미 명확화

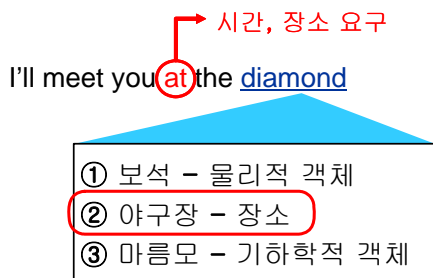
- 한 단어가 가지는 여러 가지 의미 중 적절한 것을 선택하는 과정
- 단어의 의미가 어떠한 특성을 가지고 있는가를 이용하여 적절한 것을 선택

● 의미 특징표

- 사전의 각 단어에 의미 특징표를 첨부
- 예 : ‘배’의 의미



- 예 : ‘diamond’의 의미



- ‘at’이라는 전치사가 시간이나 장소를 나타내는 단어를 필요로 함.
- ⇒ ‘야구장’이라는 의미로 해석

● 격 문법 : 문장의 의미 파악을 위한 문법 모델

- 격 언어 : 문장 구성 요소들의 위치가 비교적 자유로우며, 그 기능이 형태적인 특성에서 주어지는 언어 - 한국어, 일본어, 터키어 등
 - 철수가 온다. - 주격
 - 그 책을 철수에게 줘. - 여격
- Charles Fillmore : 격언어가 아닌 영어에 격 문법을 적용
 - 동사가 문장의 격 구조를 결정하는 중요한 요소라고 본다. - 행위의 주체, 대상, 목표 등 필요한 요소가 동사에 의해 결정된다.
 - 같은 동사라도 상이한 격 구조를 가지고 있는 문장을 형성할 수 있다.
 - ⇒ 다른 문장요소들에 의해 격을 결정

● 격의 종류

- Agent(위격-행위의 주체), Instrument(도구), Dative(여격-행위에 영향을 받은 피동자),

Factive(작위격-행위에 의해 생겨난 결과물), Locative(장소나 방향), Source(출발점), Goal (도착점), Beneficiary(수익자), Time(시간), Object(대상-그 외의 모든 것)

(예)

OPEN - O (I) (A)

□ 'open'에 부여된 격 : Object

▪ '도구'와 '위격'은 나올 수도 있다.

The door opened.

'The door'는 Object

John opened the door.

'John'은 Agent, 'the door'는 Object

John opened the door with a key.

'with a key'는 Instrument

제19강 학습

주요 학습 내용

- 학습의 개념
- 기계적 학습
- 개념 학습

교재 학습 범위 : 277~280, 301~311쪽

학습 지침

단순한 기억방법부터 보다 개념화된 학습방법에 이르는 여러 가지 학습방법을 학습한다. Winston의 학습 모델을 통해 이미 학습된 개념을 강화하는 방법과 유사한 패턴들을 그룹으로 묶는 군집화를 이해한다. 군집화 부분은 교재 12장의 패턴인식을 참조한다.

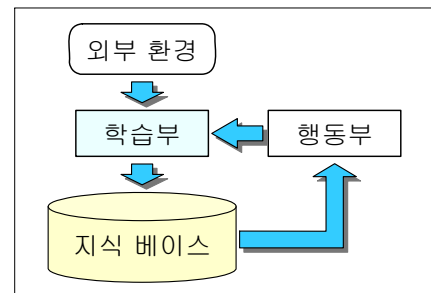
1. 학습의 개념

※ 학습이란

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

● 기본적인 학습 모델

- 학습부 : 주변의 정보로부터 일반화된 지식을 형성하여 지식베이스 구성한다.
- 행동부 : 지식 베이스를 이용하여 문제를 풀이한다.
- 학습부는 풀이 과정을 통해 새로운 지식을 추출하여 지식 베이스에 추가할 수 있다.



● 학습형태

① 지도 학습(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 비지도 학습(unsupervised learning, 또는 자율학습)

- 학습 데이터는 입력 데이터만으로 구성되며, 기대되는 출력은 제시하지 않는다.
- 유사한 입력에 대해서는 동일한 출력을 낼 수 있도록 학습이 이루어진다.

- 예) 군집화

2. 기계적 학습

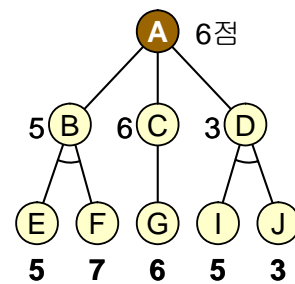
● 기계적 학습(rote learning)

- 가장 초보적인 학습 방법 : 문제와 풀이 결과를 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) 라는 문제가 나타나면 다시 풀이지 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

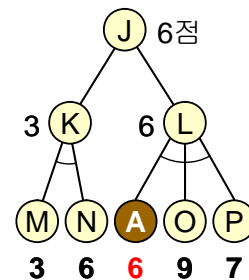
(예) Samuel의 Checker 프로그램

- 장기나 바둑 : 몇 수 앞을 머릿속에 두어 본 후 유리한 수를 둔다. 이때 더 깊이 수를 내다볼수록 좋은 수를 둘 수 있다.
- A라는 상황에서 수를 내다봄.
 - 무한히 깊이 볼 수 없으므로 정해진 깊이에 도달하면 더 이상 진행하지 않고, 정해진 정적인 평가함수에 의해 그 수의 점수를 부여 → 2수까지 본 결과 A는 6점
 - ⇒ (A, 6)이라는 값을 저장



➡ (A, 6) 저장

- 판의 형태가 J일 경우 수를 내다보는 과정에서 전에 보았던 A라는 상태가 나타나면 이를 검색하여 6점으로 부여
- ⇒ A의 점수는 이미 2수를 내다본 결과 얻은 것이므로 J로부터 4수 앞을 보고 계산한 점수의 효과를 갖는다.



● 기계적 학습시 필요한 능력

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

3. 개념학습

※ 학습 데이터를 그대로 기억하는 대신, 학습 데이터를 통해 제시된 공통적 개념을 형성하는 학습

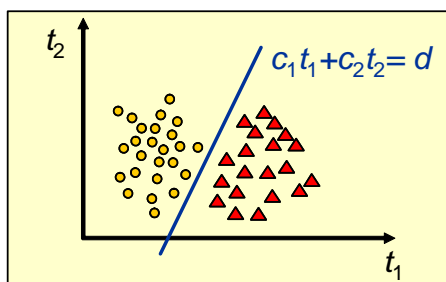
● 유형화(또는 분류, classification)

- 예의 집합으로부터 공통적 성질을 갖는 것끼리 분류된 유형을 정의하고, 어떠한 입력이 주어졌을 때 이를 분류하여 해당되는 유형으로 판단하는 과정
- 귀납적 학습

● 통계적 방법에 의한 유형화

- 식별함수의 계수 조정
- $f(t_1, t_2, \dots, t_n) = c_1 t_1 + c_2 t_2 + \dots + c_n t_n$
 - t : 패턴의 특징 (t_1, t_2, \dots, t_n 의 n 개의 변수를 갖는 함수)
- ⇒ 학습 예를 통해 c_1, c_2, \dots, c_n 의 값을 조정

(예) 식별함수의 계수 조정 (2개의 변수가 사용되는 예)



$$f(t_1, t_2) = c_1 t_1 + c_2 t_2 \begin{matrix} > d \rightarrow \text{yellow circle} \\ < d \rightarrow \text{red triangle} \end{matrix}$$

- 식별함수 : $c_1 t_1 + c_2 t_2 = d$
- 학습 예를 이용하여 c_1 과 c_2 를 조정
 - 계수 조정 방법의 예 : 퍼셉트론
- $f(t_1, t_2)$ 의 값에 따라 ○와 △를 분류

● 구조적 방법에 의한 유형

- 유형의 정의
 - 패턴 구성 요소의 구조적 결합 규칙에 의해 유형을 정의

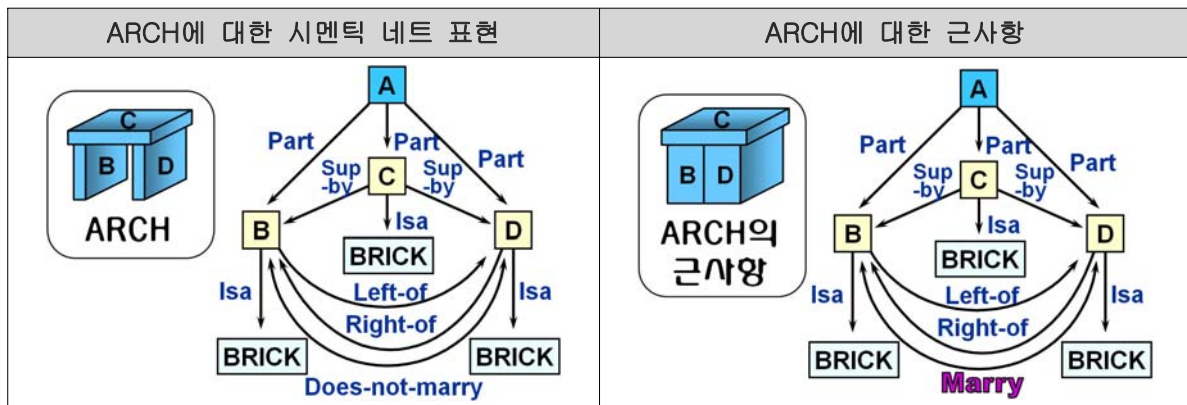
(예) Winston의 블록 세계 분석 프로그램

- 블록 세계의 선 그림(line drawing)을 분석
- 시멘틱 네트로 물체의 구조적 묘사를 함.
- 개념과 근사향

	개 념	근 사 양
HOUSE		
TENT		
ARCH		

- HOUSE, TENT, ARCH의 개념을 정의
- 근사항 : 하나의 개념과 유사하나 구분되어야 하는 것

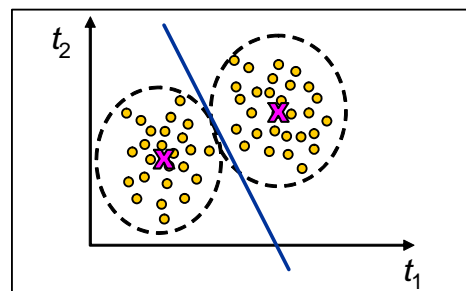
(예) ARCH의 학습



- ARCH와 정합 실시
 - B와 D가 'Marry'인 점이 다르다.
 - ⇒ ARCH의 기존 학습내용에서 'Does-not-marry'를 'Must-not-marry'로 바꿈.

● 군집화

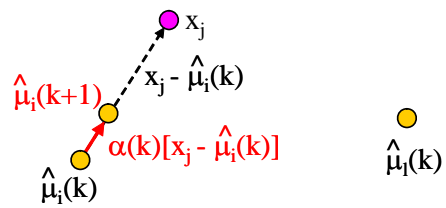
- 주어진 표본 집합을 여러 개의 군집으로 분할하는 것
- 비지도 학습의 한 형태이다.
- 군집을 대표할 수 있는 위치를 대표벡터로 선정하면, 이를 기준으로 군집들을 식별할 수 있다.



● 군집화의 예 - LVQ(learning vector quantization)

- 주어진 표본집합을 반복적인 방법을 통해 정해진 개수의 대표 벡터로 학습
- 대표 벡터들은 임의의 위치에 초기화한다.
- 각각의 표본에 대해 가장 가까운 대표 벡터를 그 표본방향으로 이동시키는 과정을 반복한다.

□ 예 x_j 에 가장 가까운 대표벡터가 $\hat{\mu}_i(k)$ 인 경우



$$\Rightarrow \hat{\mu}_i(k+1) = \hat{\mu}_i(k) + \alpha(k)[x_j - \hat{\mu}_i(k)]$$

※ $\alpha(k)$ 는 1보다 작은 양의 실수이며, k 에 따라 감소한다.

제20강 신경회로망

▶ 주요 학습 내용

- 신경회로망의 기본 구조
- 퍼셉트론

☞ 교재 학습 범위 : 311~323쪽

▶ 학습 지침

신경회로망의 정보 처리 방식이 일반적인 컴퓨터의 처리 방식과 어떻게 다른지 유념하여 학습한다. 뉴런의 동작 원리, 활성화함수, 연결형태 등을 이해한다. 퍼셉트론의 학습법은 신경회로망의 기본이 된다. 학습법과 퍼셉트론의 한계를 이해한다.

1. 신경회로망의 기본 구조

● 두뇌 속의 신경구조

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

※ 컴퓨터와의 비교

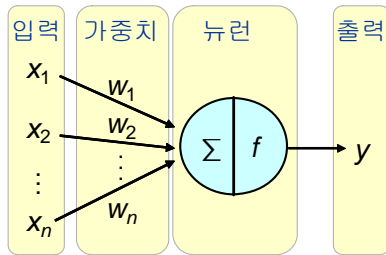
	CPU	뉴런
처리속도	□ 수 나노초(10^{-9} 초) 단위	□ 수 밀리초(10^{-3} 초) 단위
처리방법	□ 나열된 명령어를 순차적으로 실행	□ 대단위 병렬처리

※ 신경구조를 모델링하여 지능적 처리에 응용하고자 하는 시도 - 인공 신경회로망

● 신경회로망의 특성

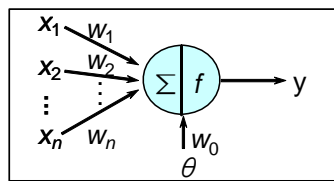
- 복잡한 기능을 수행하기 위해 많은 명령어를 포함하고 있는 CPU와는 달리 입력된 내용을 합하고, 간단한 변환을 하는 기능을 포함하는 많은 수의 뉴런으로 구성
- 각각의 뉴런은 다른 뉴런들과 방대한 연결을 유지
- 수많은 뉴런이 동시에 동작하는 병렬처리 기능 - 신경망 칩의 개발
- 정보의 저장 : 신경 연접의 연결 강도를 통해 저장
- 신경 연접은 수많은 뉴런들 간에 존재하며, 따라서 정보의 저장은 분산된다.
- 학습능력 : 학습 데이터에 따라 자동적으로 연결 강도를 조정할 수 있는 알고리즘 존재
- 고장 방지 능력 : 몇 개의 뉴런에 고장이 발생하더라도 전체 시스템의 성능이 크게 저하되지 않는다.

● 뉴런의 기본 구조



- 입력 : 외부 또는 이전 단으로부터 정보를 전달 받는다.
- 연결 가중치 : 학습에 따라 조정
- 뉴런 : 입력을 합한 결과에 변환(f)을 하여 출력

● 뉴런의 연산 기능

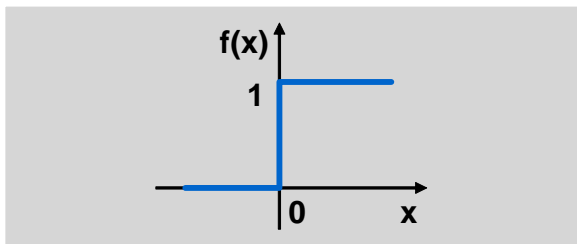


$$y = f \left(\sum_{i=1}^n x_i w_i - \theta w_0 \right)$$

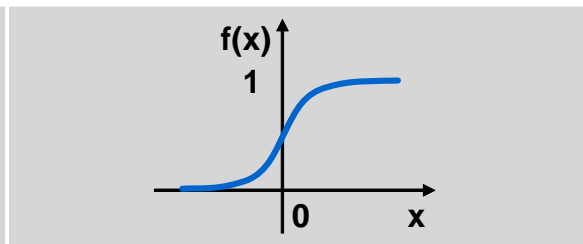
θ : 임계값 f : 활성화함수

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

● 활성화함수의 종류



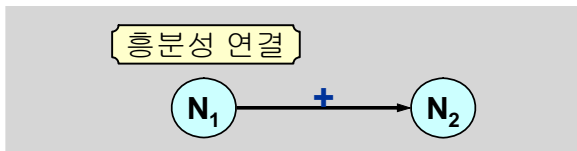
- Hard limit : 임계값을 취한 결과가 0보다 크면 1, 그렇지 않으면 0



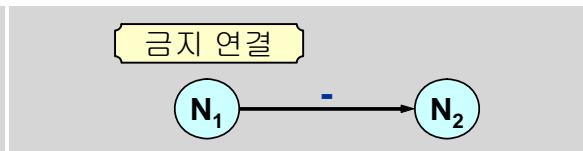
- Sigmoid
 - 미분 가능한 함수
 - 자동 이득 조절 특성 : 작은 입력에 대해서는 이득이 크고, 큰 입력에 대해서는 이득이 작다.

● 연결형태의 종류

(1) 흥분성 연결 및 금지 연결

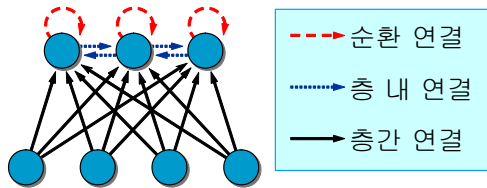


- 흥분성 연결 : 상대방을 강화



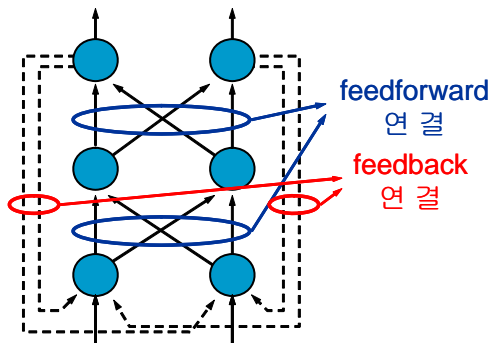
- 금지 연결 : 상대방의 출력을 약화시키는 연결

(2) 순환 연결, 층내 연결, 층간 연결



- 순환 연결 : 자기 자신에게 입력이 가해지는 연결
- 층내 연결 : 동일 층 내 뉴런과의 연결
- 층간 연결 : 서로 다른 층간의 연결

(3) Feedforward 연결 및 feedback 연결



- Feedforward 연결 : 입력단에서 출력단 방향으로 데이터가 이동되는 연결
- Feedback 연결 : 앞 단으로 되먹임되는 입력
 - 입력이 가해진 후 수렴과정을 거쳐 출력을 낸다.

● 학습방법

(1) 지도 학습(supervised learning)

- 입력 벡터와 그 입력을 가했을 경우 기대되는 출력의 쌍을 학습 데이터로 사용
- 입력된 벡터에 대한 출력이 기대되는 출력과 다르면 그 차이를 해소할 수 있도록 연결 가중치를 조정

(2) 비지도 학습(unsupervised learning)

- 입력 벡터만을 학습 데이터로 제공하고, 기대되는 출력은 제시하지 않는다.
- 입력 벡터의 집합을 벡터 공간 상에서 군집화 하여 유사한 벡터들은 동일한 출력을 낼 수 있도록 연결 가중치를 조정

2. 퍼셉트론(perceptron)

● 퍼셉트론 모델

- 1962년 Rosenblatt가 퍼셉트론의 학습이론을 제시하고, 선형 분리가 가능한 입력 벡터의 집합들에 대한 선형 분리 경계면을 학습할 수 있음을 입증
- 뉴런의 기능
 - 입력 벡터가 연결 가중치를 통해 입력되면 이를 합하여 활성화함수를 거쳐 출력
 - 활성화함수는 hard limit
- 지도 학습 - 델타 규칙을 사용

● 퍼셉트론의 학습

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

- 실제 출력과 원하는 출력의 차 δ

$$\delta = y_1 - y_2$$

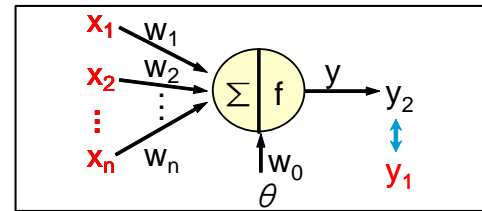
$$\Delta_i = \eta \delta x_i,$$

η : 학습률, $0 < \eta < 1$

- 모든 가중치를 δ 와 입력에 따라 조정

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

(예)



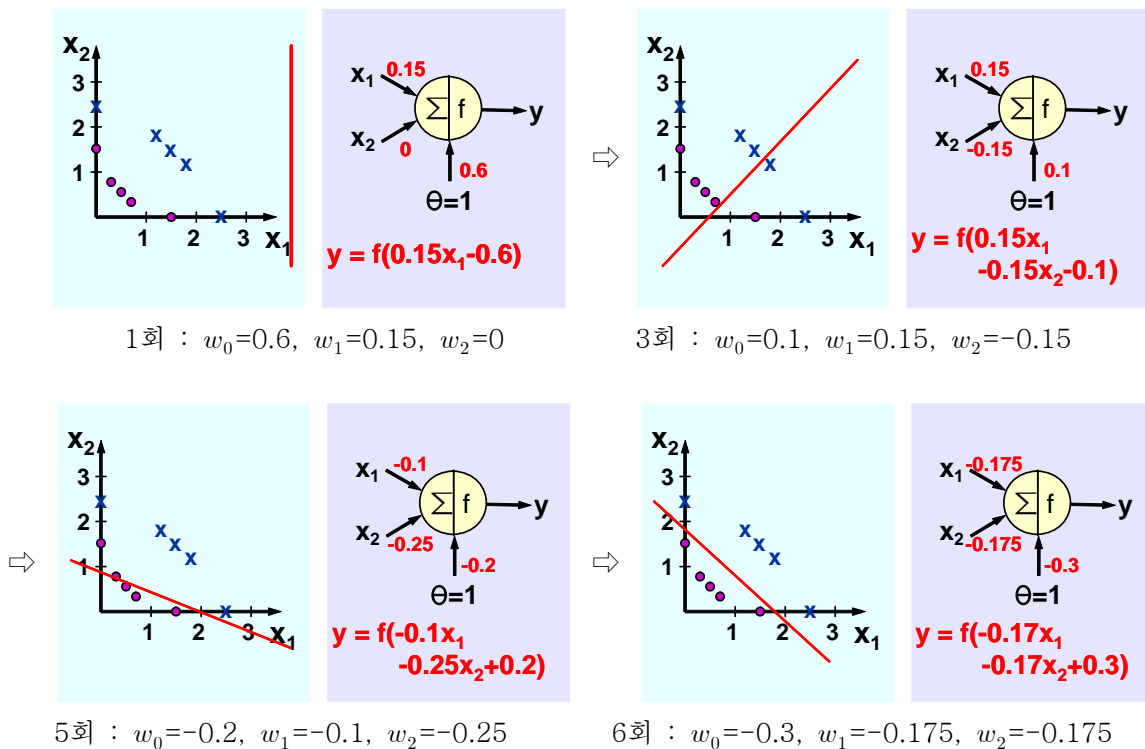
학습 데이터

1(0.25,0.75)	0(1.25,1.75)
1(0.5,0.5)	0(1.75,1.25)
1(0.75,0.25)	0(1.5,1.5)
1(0.0,1.5)	0(2.5,0.0)
1(1.5,0.0)	0(0.0,2.5)

$\Theta=1$

$y = f(x_1 w_1 + x_2 w_2 - w_0)$

- 10개의 학습 데이터
- w_0, w_1, w_2 의 초기값 = 1, 0.1, -0.3
- $\eta=0.1$ 로 하여 반복 학습
- 반복 학습에 따른 계수의 수렴 과정

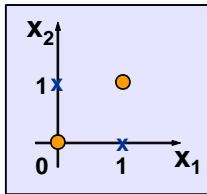


● 퍼셉트론 학습 모델의 특징

- 학습 데이터를 이용하여 반복 학습하면 이를 분리할 수 있는 선형 결정 경계를 형성
- 선형 분리가 되지 않는 경우에는 해결 불가
 - ⇒ Minsky : 단층 퍼셉트론은 XOR와 같은 단순한 함수도 생성할 수 없다고 비난

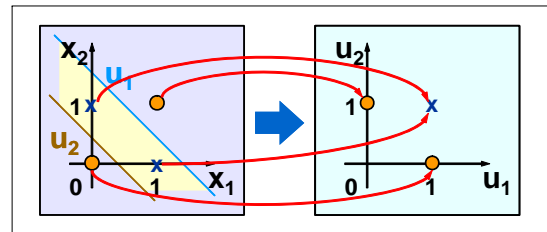
● XOR 문제

- 1개의 선형 함수로는 ○와 ×를 분리해 낼 수 없다.

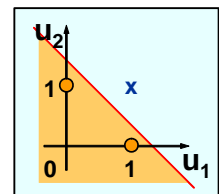


⇒ 2단계로 구현하면 분리 가능

- 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 분류한다. 이와 같은 방법으로 x_1-x_2 공간상의 점들을 u_1-u_2 좌표 공간으로 변환한다.

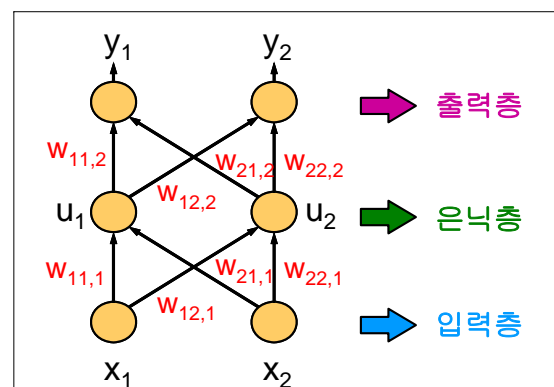


- ② u_1-u_2 공간에서는 선형 함수로 ○와 ×를 분리할 수 있게 되었으므로 선형 결정 경계를 구한다.



● 다층 퍼셉트론 구조

- 여러 단계의 층을 두어 선형 분리가 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 있었다.
- ⇒ 오차 역전도(error back propagation, BP) 모델이 제안되어 은닉층을 포함하는 다층 신경망 구조의 학습이 가능해짐.



중간시험 대비 문제와 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 대상 문제의 해를 찾아낸다.

[풀이] (㉠) 학습, (㉡) 이해, (㉢) 추론

2. 기호처리 시스템(symbol manipulation system) 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 하는데 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. 경험적 문제풀이 방식이란 무엇인가?

[풀이] 문제의 풀이방법이 정립되지 못한 경우 주어진 문제를 풀이하기 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.

4. 문제 축소에 의한 문제풀이 방식이란 무엇인가?

[풀이] 주어진 문제를 직접 풀이하기 어려울 경우, 이를 몇 개의 부분 문제로 분해한다. 이러한 문제의 분해과정은 각각의 부분 문제에 대해 반복적으로 실시된다. 이를 문제 축소(problem reduction)라고 한다. 이때 분해된 부분 문제가 아주 쉽게 풀이될 수 있는(해가 알려진) 문제가 되었다면 그 부분 문제는 풀이된 것이다. 만일 분해한 부분 문제들이 모두 풀이되면 그 해들의 집합으로 문제 전체의 해가 구해질 수 있다. 이와 같은 문제풀이 과정을 문제축소에 의한 문제풀이 방식이라고 한다.

5. 상태, 연산자, 그리고 상태공간에 대하여 설명하시오.

[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 변화 수순을 찾는 것이라 할 수 있는데, 이처럼 주어진 상태를 변화시키는 방법들의 집합을 연산자라 한다. 초기상태에 대해 적용할 수 있는 모든 연산자를 적용함으로써 새로운 상태들의 집합이 만들어지며, 이들 상태에 대해 또다시 적용 가능한 연산자를 모두 적용하면 이에 해당되는 후계상태들이 만들어지게 된다. 이와 같이 연산자들을 반복 적용하면 상태들이 서로 그래프 형태로 연결된 모든 상태들의 집합이 만들어지는데, 이를 상태공간이라 한다.

6. 상태 묘사란 무엇인가?

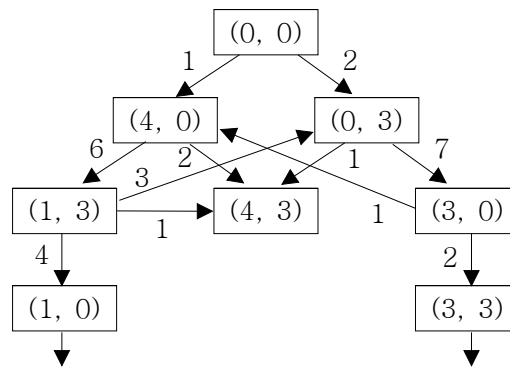
[풀이] 상태공간에서 문제를 풀이하고자 한다면 문제의 상태를 컴퓨터상에 표현할 수 있어야 한다. 문제풀이 과정 중 어느 한 시점에서 문제의 형태를 나타내는 상태는 대상문제를 적절히 표현할 수 있는 자료구조를 통해 컴퓨터 내에 표현될 수 있는데, 이를 상태 묘사(state description)라고 한다. 상태 묘사는 벡터, 트리, 리스트 등 다양한 자료구조들 중에서 해당 문제의 상태를 적절히 표현할 수 있는 방법을 선택하게 된다.

7. 4리터와 3리터들이 물병을 사용하는 물병 문제에 대해 다음과 같이 연산자가 정의되어 있을 때 상태공간의 일부를 보이시오. 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 상태 묘사 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.

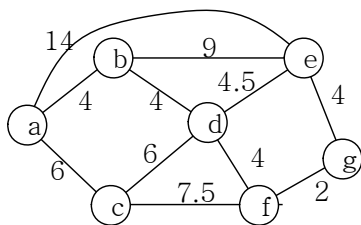


8. 상태 묘사 방법을 설명하는 다음 문장의 공란에 적합한 내용은?

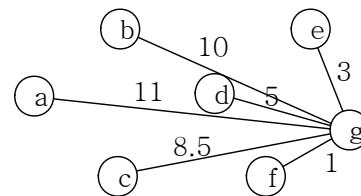
비슷한 성격을 지닌 상태들의 집합을 나타내기 위해 변수를 사용할 수 있다. 이와 같은 표현을 (㉠) (이)라고 하며, 이때 사용된 변수를 (㉡) (이)라고 한다.

[풀이] (㉠) 상태 묘사 스키마(상태 묘사 개요), (㉡) 스키마 변수(개요 변수)

9. 다음 [그림 1]은 7개의 도시와 이들을 연결하는 도로망을 보여 준다. 간선에 표시된 값은 도시 사이 도로의 거리이다. 또한 [그림 2]는 각 도시로부터 목적지인 도시 g까지의 직선거리를 나타낸다. 도시 a에서 출발하여 도시 g로 가기 위한 경로를 탐색하고자 한다.



[그림 1] 도시 사이의 도로거리



[그림 2] 도시 g까지의 직선거리

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

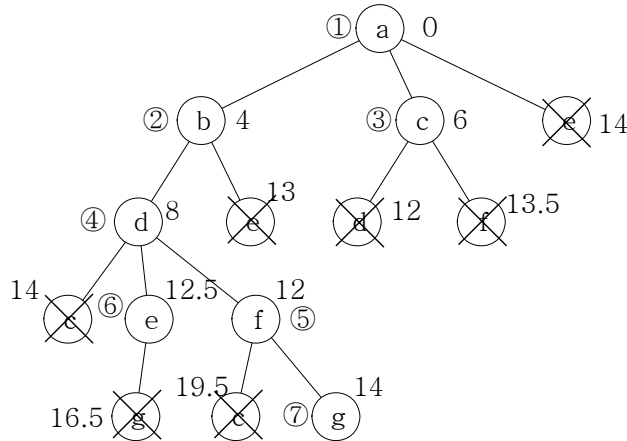
(나) 균일비용 탐색과 A* 알고리즘에 의한 탐색 트리를 각각 보이시오. 각 노드에는 그 노드의 평가함수 값을 표시하고, 노드가 확장되는 순서를 ①, ②, ...로 표시하시오.

[풀이] (가) 출발 노드로부터 탐색을 진행하여 특정 노드 n 까지 도달하였다고 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.

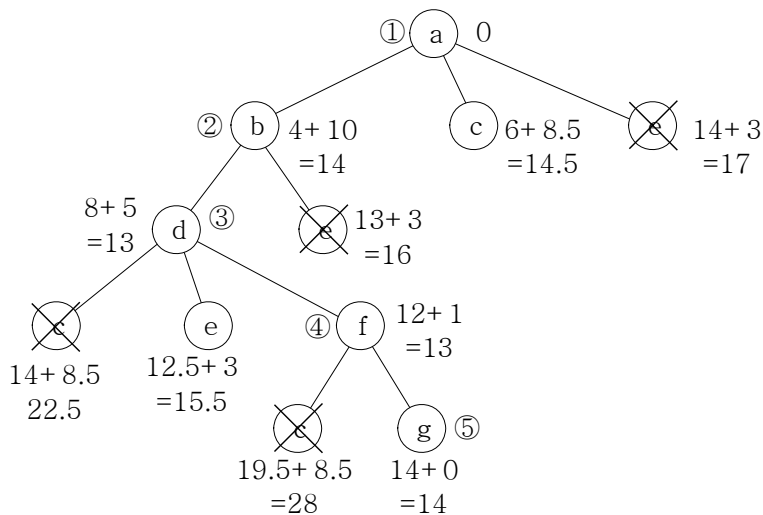
본 문제에서 $g(n)$ 은 출발 도시 a로부터 도시 n 까지 도달하는 동안 거친 도로의 거리를 합산한

것으로 정의할 수 있으며, $\hat{h}(n)$ 은 도시 n 으로부터 목적지인 도시 g 까지의 직선거리로 정의할 수 있다.

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



A* 알고리즘에서는 $\hat{f}(n)$ 이 최소인 노드를 선택하여 확장한다. 다음 그림은 A* 알고리즘의 탐색 트리이다. 예를 들어 ③번째로 확장되는 노드 d 의 경우 노드 d 까지 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.



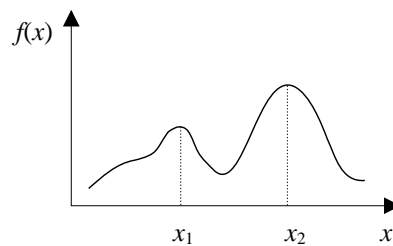
10. 지역최대치 문제란 무엇인가?

[풀이] 탐색의 목표가 정해진 평가치(수익, 성능 등)를 최대화하기 위해 시스템의 계수를 조정하는 문제에서 풀이에 실패하게 되는 상황 중 한 가지이다. 이러한 계수 최적화 문제의 일반적인 풀이방법은 현재 상태에서 시스템의 평가치가 가장 많이 개선되는 인접한 값으로 정해진 범위 내에

서 시스템의 계수를 조정하는 것이다. 그런데 이러한 문제에서 어떠한 평가치 값이 목표값이라고 절대적으로 정하기 어려운 경우가 많다. 대신 다음과 같이 목표 상태를 정의하게 된다.

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

그러나 탐색을 진행하는 과정에서 목표상태는 아니지만 아래 그림에서와 같이 지역적 극대값(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.



11. A* 알고리즘의 평가함수에 대하여 설명하고, 평가함수에 따른 최적경로 탐색 가능 여부에 대하여 설명하시오.

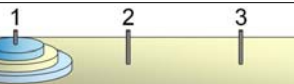

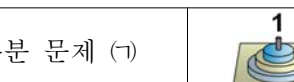

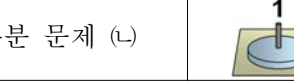
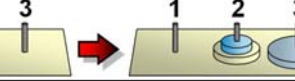
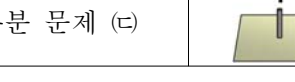

[풀이] 탐색 과정 중 임의의 노드 n 을 거쳐 목표노드에 도달하는 데 필요한 비용은 출발노드로부터 노드 n 까지 도달하는 데 소비한 경로비용 $g(n)$ 과 노드 n 으로부터 목표노드까지 도달하는데 필요한 비용 $h(n)$ 의 합이 된다. 이때 탐색 과정 중 노드 n 에 도달했을 때 $g(n)$ 은 이미 거쳐 온 경로의 비용을 합산하면 계산할 수 있으나, $h(n)$ 은 아직 탐색하지 않은 경로의 비용이므로 실제 비용을 알 수 없다. 경험적 탐색 알고리즘인 A* 알고리즘에서는 경험적 지식을 반영하여 $h(n)$ 의 예측치인 $\hat{h}(n)$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

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

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

기말시험 대비 문제와 해설

※ (1~2) 1번 말뚝의 원판을 모두 3번 말뚝으로 옮기는 하노이 탑 문제에 대한 질문에 답하시오. 연산자는 큰 원판이 작은 원판 위에 놓이지 않도록 한 장의 원판을 옮기는 것이다.

초기 상태		목표 상태	
			
부분 문제 (ㄱ)			
부분 문제 (ㄴ)			
부분 문제 (ㄷ)			

1. 주어진 문제를 부분 문제 (㉠)~(㉣)으로 축소하였을 때, 원시문제에 해당되는 것은?

- ① (ㄱ) ② (ㄴ)
③ (ㄷ) ④ 없음

[정답] ②

[해설] 문제에서 연산자를 “큰 원판이 작은 원판 위에 놓이지 않도록 한 장의 원판을 옮기는 것”으로 정의하였다. 원시문제는 쉽게 풀이될 수 있거나 해가 완전히 알려져 있어서 더 이상 분해할 필요가 없는 문제로서, 한 번의 연산자 적용으로 풀이될 수 있다면 이는 원시문제에 해당된다. (L)이 이러한 문제에 해당된다.

2. 이 문제 축소에 대한 올바른 설명은?

- ① (가)만 풀이되면 (나), (다)의 풀이에 관계없이 문제가 풀이된다.
- ② (나)만 풀이되면 (가), (다)의 풀이에 관계없이 문제가 풀이된다.
- ③ (가), (나), (다) 중 어느 하나라도 풀이되면 문제가 풀이된다.
- ④ (가), (나), (다)이 모두 풀이되어야 문제가 풀이된다.

[정답] ④

[해설] (가), (나), (다)은 주어진 문제를 나누어 놓은 부분 문제들이다. 각각의 부분 문제가 모두 풀이되어야만 전체 문제에 대한 해가 구해진다.

3. 다음 중 풀이될 수 있는 노드를 모두 나열한 것은?

- (ㄱ) 종단 노드(원시문제)
 (ㄴ) OR 후계 노드를 가지며 그 중 어느 하나가 풀이된 노드
 (ㄷ) 여러 개의 AND 후계 노드를 가지며 그 중 하나만 풀이된 노드
 (ㄹ) 종단 노드가 아니면서 후계 노드가 없는 노드

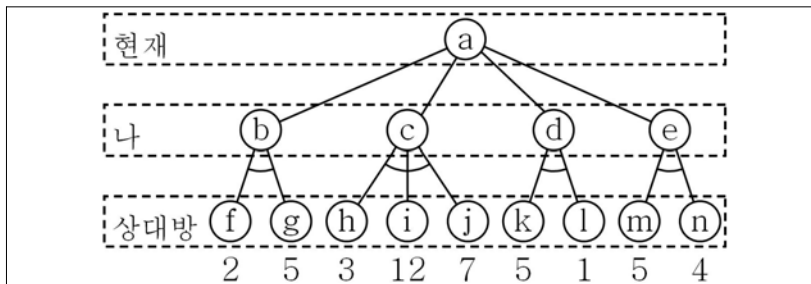
- ① (ㄱ), (ㄴ) ② (ㄱ), (ㄷ)
 ③ (ㄴ), (ㄹ) ④ (ㄷ), (ㄹ)

[정답] ①

[해설] 풀이될 수 있는 노드는 다음과 같이 정의된다.

- ☐ 종단 노드(원시문제를 나타내므로 풀이될 수 있는 노드임)
- ☐ OR 후계 노드들을 갖는 경우 적어도 한 노드가 풀이될 수 있을 때
- ☐ AND 후계 노드들을 갖는 경우 모든 후계 노드들이 풀이될 수 있을 때

※ (4~5) 다음의 최대 최소 탐색트리에 대한 질문에 답하십시오.



4. f~n에 표시된 평가값으로 구한 현재상태 a의 평가값은?

- ① 1 ② 4
 ③ 5 ④ 12

[정답] ②

[해설] b의 평가값은 f와 g의 최소값인 2, c의 평가값은 h~j의 최소값인 3, d의 평가값은 k와 l의 최소값인 1, e의 평가값은 m과 n의 최소값인 4이다. a의 평가값은 b~e의 최대값이므로 4가 된다.

5. 현재 상태에서 내가 선택할 수는?

- ① b ② c
 ③ d ④ e

[정답] ④

[해설] 나는 내가 둘 수 있는 수 중에서 가장 유리한 수를 선택한다. 그러므로 평가값이 가장 큰 수를 선택하여야 하며, 이에 따라 e를 선택한다.

6. 다음의 차이-연산자 표에서 차이 A가 발견되었을 때, 주어진 문제의 축소 결과 만들어지는 부분 문제에 속하는 것은? (S는 초기상태, G는 목표상태, F는 연산자 집합)

차이 \ 연산자	O ₁	O ₂	O ₃	O ₄
A	✓			
B		✓		✓
전제상태	S ₁	S ₂	S ₃	S ₄
적용후상태	N ₁	N ₂	N ₃	N ₄

- ① (S, {O₁}, G) ② (S, {O₁}, S₁)
 ③ (S₁, F, S) ④ (S₁, {O₁}, N₁)

[정답] ④

[해설] 차이-연산자 표를 통해 차이 A가 발견되었을 경우 이를 해소할 수 있는 연산자는 O₁임을 알 수 있다. 따라서 주어진 문제 (S, F, O)는 O₁을 적용할 수 있는 전제상태로 만드는 문제 (S, F, S₁)과 O₁을 적용하는 문제 (S₁, {O₁}, N₂), 그리고 N₂를 목표상태인 O로 만드는 문제 (N₂, F, O)로 나눌 수 있다.

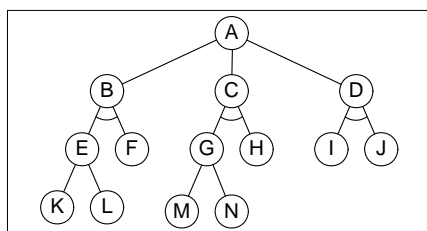
7. 문제 축소(problem reduction) 방식의 문제 풀이 방법에서 AND 후계 노드들에 대한 올바른 설명은?

- ① 주어진 문제의 해인가의 판단이 끝난 노드들의 집합이다.
 ② 하나라도 풀이되지 않으면 부모 노드는 풀이되지 않는다.
 ③ AND 후계 노드는 원시노드이다.
 ④ 어느 한 노드라도 풀이되면 부모 노드는 풀이된다.

[정답] ②

[해설] AND 후계 노드는 부모 노드의 문제를 해결하기 위해 분할하여 놓은 부분 문제이다. 부모 노드가 풀이되기 위해서는 AND 후계 노드들이 모두 풀이되어야 한다.

8. 다음 AND/OR 그래프의 리프 노드 중 원시 노드일 경우 문제 A가 풀이되는 것을 모아 놓은 것은?



- ① K, H, I ② L, M, N

③ K, L, M, N, J

④ M, H

[정답] ④

[해설] M은 G의 OR 후계 노드이므로 M이 원시문제이면 G가 풀이된다. G는 문제 C의 AND 후계 노드 중 하나이며, 나머지 AND 후계 노드인 H가 원시문제이므로 문제 C가 풀이된다. 문제 C는 A의 OR 후계 노드이므로 C가 풀이됨으로써 A가 풀이된다.

9. 어떤 문제 P의 초기상태를 S, 목표상태를 G, 연산자의 집합을 F라 하자($P=(S, F, G)$). S와 G의 차이를 해소할 수 있는 연산자를 f, f를 적용할 수 있는 전제상태를 G_1 , G_1 에 f를 적용한 결과를 G_2 라 하면, 주어진 문제P는 어떻게 축소될 수 있는가?

① (S, F, G_1) or (G_1 , {f}, G_2) or (G_2 , F, G)② (S, F, G_1) or (S, {f}, G_2) or (S, F, G)③ (S, F, G_1) and (G_1 , {f}, G_2) and (G_2 , F, G)④ (S, F, G_1) and (S, {f}, G_2) and (S, F, G)

[정답] ③

[해설] 초기상태와 목표상태를 해소할 수 있는 연산자를 이용한 문제 축소는 그 연산자를 적용할 수 있는 전제상태와 연산자를 적용한 결과에 해당하는 상태를 중간상태로 하여 이루어진다. 이때 축소된 부분 문제들은 이들이 모두 풀이되어야만 주어진 문제가 풀이되는 AND 후계상태들이 된다.

10. 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 문제 풀이에 필요한 지식을 저장하는 지식 베이스, 주어진 문제의 상황에 대하여 지식 베이스 내의 지식을 이용한 추론을 함으로써 결론을 제시하는 추론기관의 두 가지 핵심 요소와 함께 사용자와 지식기반 시스템 사이의 연결을 담당하는 사용자 인터페이스 등으로 구성된다.

11. 다음 중 지식의 사용에 관한 지식을 나타내는 것은?

① 문제영역 지식

② 사실

③ 메타지식

④ 경험적 지식

[정답] ③

[해설] 어떤 규칙들이 다른 규칙들 보다 우선적으로 적용하여야 할 조건을 제시하거나, 특수한 문제의 풀이에 유용한 지식의 적용순서를 제시하거나, 추론에 대한 설명을 통해 결과에 대한 신뢰감

제공하는 등 지식의 사용에 관한 지식을 메타 지식(meta-knowledge)이라 한다.

12. 다음 중 상호독립적, 단편적인 지식들을 나열해 놓은 형태의 지식으로, 별도의 추론기관에 의해 추론이 이루어지는 것은?

- ① 선언적 지식 ② 경험적 지식
③ 절차적 지식 ④ 프로그램화된 지식

[정답] ①

[해설] 선언적 지식은 상호 독립적, 단편적인 지식들을 나열해 놓은 형태의 지식으로, 지식 자체에 실행의 제어에 대한 정보가 없으며, 별도의 추론기관에 의해 추론에 사용된다. 반면 절차적 지식은 어떠한 경우에 무엇을 어떻게 할 것인가에 대한 지식으로, 지식 사용에 대한 제어 정보가 지식 자체에 내포되어 있다.

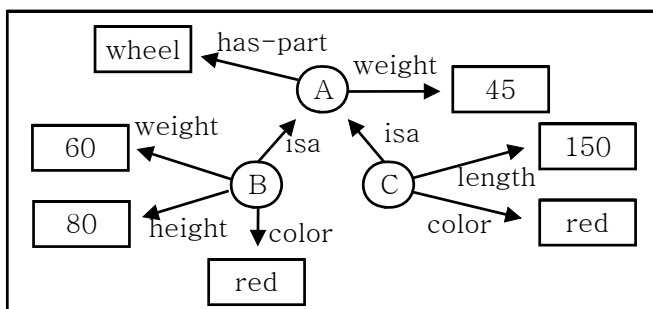
13. 프레임으로 지식을 표현할 경우 사용할 수 있는 기능이 아닌 것은?

- ① 특성 상속 ② 부가 프로시저
③ 내정값 사용 ④ 귀납 추론

[정답] ④

[해설] 프레임은 어떠한 객체를 표현하기 위한 속성 및 속성 값의 집합으로, 관련된 프레임들이 상위 개념, 하위 개념으로 분류되어 연결된다. 이때 상위 개념의 속성이 하위 개념에 상속되는 특성 상속을 사용할 수 있으며, 속성의 값을 읽거나 수정하는 등의 사용을 했을 때 그 사용과 관련되어 수행해야 할 동작을 지시하는 부가 프로시저를 사용함으로써 절차적 지식을 포함할 수 있다. 또한 어떠한 속성의 값이 지정되지 않았을 때 사용할 수 있는 내정값을 정해 놓을 수 있다.

14. 다음 시맨틱 네트에서 A, B, C가 가질 수 있는 속성과 그 값이 올바른 것은?



- ① A의 color = red ② C의 hight = 80
③ B의 weight = 45 ④ C의 has-part = wheel

[정답] ④

[해설] A의 속성은 has-part(=wheel)와 weight(=45)이다. B는 A와 isa 관계에 있으므로,

weight(=60) 및 height(=80), color(=red)와 함께 A로부터 상속받은 has-part(=wheel)라는 속성을 갖는다. C 역시 A와 isa 관계에 있으므로, length(=150) 및 color(=red)와 함께 has-part(=wheel)와 weight(=45)라는 속성을 갖는다.

15. 다음 중 선언적 지식에 대한 설명을 모두 모은 것은?

- (가) LISP과 같은 프로그래밍 언어로 작성된 명령어의 집합
(나) 지식 사용에 대한 제어 정보가 지식 자체에 내포되어 있다.
(다) 상호 독립적이고 단편적인 지식들을 나열해 놓은 형태이다.
(라) 별도의 추론기관이 있어, 이에 의해 추론을 한다.

- ① $(\neg), (\bot)$
② $(\bot), (\top)$
③ $(\neg), (\exists)$
④ $(\top), (\forall)$

[정답] ④

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

16. 조건과 그 조건을 만족할 경우 내리게 되는 결론, 취할 행동 등을 기술하는 형태의 지식 표현 방법은?

- ① 규칙 ② 시맨틱 넷트
③ 프레임 ④ LISP 프로그램

[정답] ①

[해설] 규칙은 ‘IF 조건부 THEN 결론부’ 형식의 지식 표현 방법으로, 현재의 상태에 의하여 조건부가 만족되는 규칙을 찾아 결론부의 행동을 실행한다.

17. 프레임을 이용한 지식 표현에 대한 올바른 설명은?

- ① 특성 상속을 사용한다.
- ② 절차적 지식만을 포함한다.
- ③ 절차적 지식을 포함시킬 수 없다.
- ④ 형식논리를 이용한 지식 표현 방법이다.

[정답] ①

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

18. 다음 중 부가 프로시저를 통해 절차적 지식을 표현할 수 있는 지식 표현 방법은?

- ① 프레임 ② 술어논리

③ 규칙

④ 시멘틱 네트

[정답] ①

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

19. 다음 중 항상 참인 추론은?

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

[정답] ①

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

20. (부모절 \Rightarrow 도출절)의 쌍이 올바른 것은?

- ① $p, p \vee q \Rightarrow q$ ② $p \wedge q, \sim p \wedge q \Rightarrow q$
- ③ $\sim p, p \Rightarrow \text{false}$ ④ $p \vee q, q \vee r \Rightarrow p \vee r$

[정답] ③

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

21. 다음은 도출 연역에 의한 정리증명 알고리즘이다. 공란에 넣을 내용이 올바른 것은?

1. 증명하고자 하는 ① 공리의 리스트에 첨가
2. 공리 리스트를 ② (으)로 표현 후, 절 분리
3. 도출 가능한 쌍이 없을 때 까지 다음을 반복
 - 1) 도출 가능한 절의 쌍을 찾아 도출절을 구한다.
 - 2) 도출절을 공리 리스트에 추가한다.
 - 3) ③ 가 얻어지면 정리가 참임이 증명된다.
4. 정리가 ④ 임을 알리고 끝낸다.

- ① 정리를 그대로 ② 선언 표준형
 ③ false ④ 참

[정답] ③

[해설] 도출연역에 의한 정리증명 알고리즘에서는 증명하고자 하는 정리를 부정한 후 거짓을 추론해냄으로써, 증명하고자 했던 정리가 참임을 증명하는 방법을 사용한다. 그러므로 ①은 ‘정리를 부정하여’, ②는 ‘연언표준형’, ④는 ‘거짓’이 된다.

22. 술어논리식으로 표현된 다음 절들에 의해 도출될 수 있는 것은? (P, Q, R, S는 술어, A, B는 객체상수, x, y, z는 객체변수임)

- | | | |
|-----------------------------|---------------------|------------------------|
| $\sim P(x, y) \vee Q(x, y)$ | $R(x, A) \vee S(A)$ | $P(x, y) \vee R(x, y)$ |
| $\sim R(A, B)$ | $P(A, z)$ | |

- ① S(A) ② Q(A, y)
 ③ R(A, y) ④ $Q(x, y) \wedge R(x, y)$

[정답] ②

[해설] $\sim P(x, y) \vee Q(x, y)$ 와 $P(A, z)$ 를 대상으로 도출을 할 수 있다. $\sim P(x, y) \vee Q(x, y)$ 에서 객체변수 x를 객체상수 A로 단일화하면 $\sim P(A, y) \vee Q(A, y)$ 가 된다. 또한 $P(A, z)$ 에서 객체변수 z를 y로 통일하여 $P(A, y)$ 로 표현한다. 이제 $\sim P(A, y) \vee Q(A, y)$ 와 $P(A, y)$ 에 대한 도출절인 $Q(A, y)$ 를 얻을 수 있다.

23. 다음 중 연언 표준형에 해당되는 논리식은?

- ① $(P \wedge Q) \vee (R \wedge S)$ ② $P \wedge (Q \vee R \wedge S)$
 ③ $P \wedge \sim (Q \vee R \vee S)$ ④ $(\sim P \vee Q) \wedge (R \vee \sim S)$

[정답] ④

[해설] 연언 표준형은 리터럴의 논리합으로 연결된 절들을 논리곱으로 연결한 형식의 표준형 논리식이다. ①은 선언 표준형이다.

24. 다음 중 술어논리의 정형식에 해당되는 것은? (단, P, Q는 술어기호, x, y는 객체변수, f는 함수기호이다.)

- $$\begin{array}{ll} \textcircled{1} P(Q) & \textcircled{2} \sim P(x) \rightarrow y \\ \textcircled{3} f(P(x) \vee Q(x)) & \textcircled{4} \exists x \sim P(f(x)) \vee Q(x) \end{array}$$

[정답] ④

[해설] 객체상수 및 객체변수는 항이며, 함수기호(항) 역시 항이다. 참, 거짓, 술어기호(항)는 정형식이며, 정형식의 부정, 정형식 간의 조건명제 역시 정형식이다. 또한 객체변수를 포함하는 경우 정형식에 전칭기호나 존재기호를 사용한 것 역시 정형식이다.

25. $(P \wedge Q) \rightarrow (\sim R \wedge S)$ 를 도출연역을 하기 위해 절 형태로 변환한 것은?

- ① $\sim P \vee \sim Q \vee \sim R, \sim P \vee \sim Q \vee S$
- ② $P \vee \sim R, Q \vee \sim R, P \vee S, Q \vee S$
- ③ $\sim P, \sim Q, \sim R \wedge S$
- ④ $P \wedge Q, \sim R \wedge S$

[정답] ①

$$\begin{aligned} \text{[해설]} \quad (P \wedge Q) \rightarrow (\sim R \wedge S) &= \sim(P \wedge Q) \vee (\sim R \wedge S) \\ &= (\sim P \vee \sim Q) \vee (\sim R \wedge S) \\ &= (\sim P \vee \sim Q \vee \sim R) \wedge (\sim P \vee \sim Q \vee S) \\ \Rightarrow \quad &\sim P \vee \sim Q \vee \sim R, \quad \sim P \vee \sim Q \vee S \end{aligned}$$

26. $P(x) \vee Q(y) \vee R(x,y)$ 와 $\sim R(A,B) \vee S(C)$ 에 의해 도출되는 결과는? (단, x, y 는 객체변수, A, B, C 는 객체상수이다.)

- ① $P(x) \vee Q(y)$
② $S(C)$
③ $P(A) \vee Q(B) \vee S(C)$
④ $P(A) \vee Q(B) \vee \sim R(A,B)$

[정답] ③

[해설] $R(x, y)$ 와 $\sim R(A, B)$ 에서 x 를 A , y 를 B 로 단일화한 후 나머지 논리식들의 논리합으로 도출절을 구한다.

27. $P \rightarrow Q$ 가 참일 때, modus ponens에 의한 올바른 추론은?

- ① P가 참이면 Q도 참이다.
- ② P가 거짓이면 Q도 거짓이다.
- ③ Q가 참이면 P도 참이다.
- ④ P가 거짓이면 Q는 참이다.

[정답] ①

[해설] modus ponens(긍정식)은 P라는 사실과 $P \rightarrow Q$ 라는 명제로부터 Q라는 사실을 추론한다.

28. 퍼지 집합 A의 소속함수 $\mu_A(x)$ 에 대하여 적절히 표현한 것은? ([a, b]는 집합 $\{y \mid a \leq y \leq b \text{인 실수}\}$ 를 나타낸다.)

- ① $\mu_A(x) \in \{0, 1\}$ ② $\mu_A(x) \in \{-1, 0, 1\}$
 ③ $\mu_A(x) \in [-1, 1]$ ④ $\mu_A(x) \in [0, 1]$

[정답] ④

[해설] 퍼지 집합의 소속함수 값은 0부터 1까지의 범위에 속한다.

29. 퍼지 추론 및 제어기에 대한 올바른 설명을 모두 나열한 것은?

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

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

[정답] ②

[해설] 퍼지 추론은 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 하기 위해서는 특정한 값으로 변환하는 비퍼지화를 하여 사용한다.

※ (30~31) 다음과 같이 정의된 퍼지 집합 A와 B의 연산에 대한 질문에 답하십시오.

x	$\mu_A(x)$	$\mu_B(x)$	$\mu_{A \cup B}(x)$	$\mu_{\overline{A}}(x)$
a	1	0	(ㄱ)	(ㄷ)
b	0.6	0.5	(ㄴ)	(ㄹ)
c	0.2	0.7	(ㄷ)	(ㄱ)
d	0	0	(ㄹ)	(ㅇ)

30. (ㄱ)~(ㅇ)에 넣을 값을 순서대로 나열한 것은?

- ① 1, 0.6, 0.7, 0 ② 0, 0.5, 0.2, 0
 ③ 1, 0.6, 0.2, 0 ④ 0, 0.5, 0.7, 0

[정답] ①

[해설] 퍼지 집합의 합집합은 소속함수값의 최대값을 취한다.

31. (ㄱ)~(ㅇ)에 넣을 값을 순서대로 나열한 것은?

- ① 1, 0.6, 0.2, 0 ② 0, 0, 0, 1
 ③ 0, 0.4, 0.8, 1 ④ 0, 0.4, 0.8, 0

[정답] ③

[해설] 퍼지 집합의 여집합은 원 소속함수값을 1에서 뺀 값을 소속함수값으로 계산한다.

32. 퍼지 집합에 대한 올바른 설명은?

- ① 퍼지 집합은 고전집합의 한 종류이다.
 ② 원소의 소속함수값은 0 이상, 1 이하의 범위에 속한다.
 ③ 최소한 한 개의 원소는 소속함수가 1이다.
 ④ 모든 원소의 소속함수값을 더하면 1이다.

[정답] ②

[해설] 퍼지 집합은 0부터 1까지의 범위에 속하는 값으로 원소의 소속함수값이 정의된다. 고전집합은 이러한 퍼지 집합 중 소속함수가 0 또는 1 중 하나인 특수한 경우로 생각할 수 있다. 소속함수값은 확률과는 다르며, 따라서 확률밀도함수와 같이 소속함수값을 더하면 1이 되는 것은 아니다.

33. 다음 퍼지 논리식의 관계가 올바른 것은?

- ① $a \wedge (b \vee c) = (a \wedge b) \vee c$ ② $a \vee 1 = a$
 ③ $a \wedge \sim a = 0$ ④ $a \vee (a \wedge b) = a$

[정답] ④

[해설] ④의 퍼지논리값은 $\max(a, \min(a, b))$ 이다. 이때 $a \geq \min(a, b)$ 이므로 $\max(a, \min(a, b)) = a$ 이다. 이는 $a \wedge (a \vee b) = a$ 와 함께 흡수법칙에 해당된다.

34. 퍼지 집합 A와 B의 교집합은?

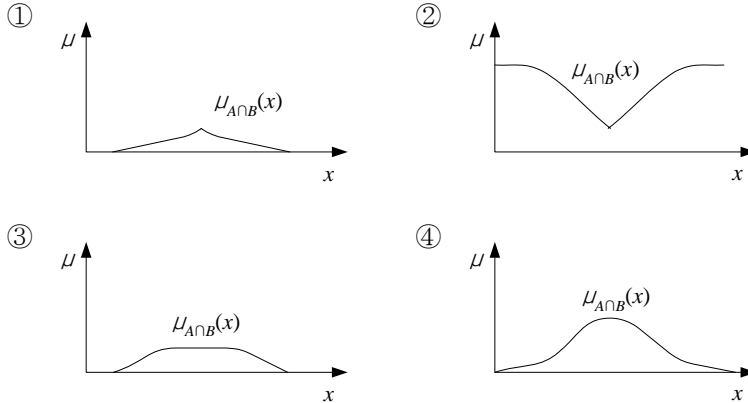
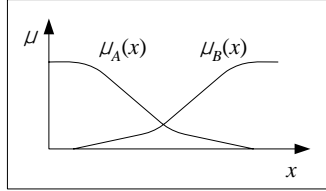
$A = \{(a, 1.0), (b, 0.7), (c, 0.2)\}$ $B = \{(b, 0.5), (c, 0.4), (d, 0.1)\}$

- 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] 퍼지 집합의 교집합은 소속함수값의 최소값을 취한다. 원소 a와 d의 경우는 각각 집합 B와 A에 나열되어 있지 않으므로 소속함수값이 0인 것으로 계산해야 한다.

35. 퍼지 집합 A와 B의 소속함수가 다음과 같을 때 $A \cap B$ 의 소속함수는?



[정답] ①

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

36. 생성 시스템에 대한 올바른 설명은?

- ① 생성 메모리는 현재의 상태를 나타내는 사실들을 저장한다.
- ② 작업 메모리의 내용에 만족되는 생성규칙이 실행된다.
- ③ 생성 메모리는 단기 메모리이다.
- ④ 인터프리터는 생성규칙의 편집을 수행한다.

[정답] ②

[해설] 생성 메모리에는 생성규칙들이 저장되어 있으며, 작업 메모리에는 현재의 상태를 표현하는 사실들이 저장되어 있다. 인터프리터는 작업 메모리의 현재 상태에 의해 조건이 만족되는 생성규칙을 선택하여 실행한다.

37. 추론방법에 대한 설명으로 올바른 것은?

- ① 연역법에 의한 추론은 완전하지 않다.
- ② 유도법은 'if A then B'와 'B'로부터 'A'를 추론한다.
- ③ 귀납법은 'A'와 'if A then B'로부터 'B'를 추론한다.
- ④ 귀납법에 의한 추론은 완전하다.

41. 생성 시스템의 구성요소 중 규칙을 저장하고 있는 것은?

- ① 생성 메모리 ② 단기 메모리
③ 인터프리터 ④ 작업 메모리

[정답] ①

[해설] 작업 메모리는 단기 메모리라고도 하며, 현재의 상태를 저장한다. 인터프리터는 작업 메모리의 내용에 의해 만족되는 생성 메모리의 규칙을 선택하여 실행한다.

42. 규칙기반 시스템에서 주로 연역추론을 사용하는 이유는?

- ① 정확한 전제와 규칙이 있다면 정확한 결론을 추론할 수 있다.
- ② 공통적인 내용을 표현하는 사실로부터 일반적 규칙을 유추해 낼 수 있다.
- ③ 조건부의 사실이 주어지지 않아도 결론부의 결론을 도출할 수 있다.
- ④ 스스로 규칙을 만들어 낼 수 있다.

[정답] ①

[해설] 연역추론은 P와 $P \rightarrow Q$ 로부터 Q를 추론하는 것으로, 추론결과는 항상 정확하다. ②는 귀납법에 대한 설명이다.

43. 다음 중 현장 전문가와 비교하였을 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 업무 실행 능력이 다를 수 있으나, 전문가 시스템은 문제풀이 과정이 항상 일관적인 특성을 갖는다.

44. 전문가 시스템을 구축하는 과정에서 지식공학자의 역할은 무엇인가?

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

[정답] ④

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

45. 전문가 시스템을 구성하는 핵심요소는?

- ① 지식 베이스와 지식 편집기
- ② 추론기관과 지식 편집기
- ③ 지식 편집기와 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 구성요소이다.

46. 전문가 시스템의 추론기관에 대한 잘못된 설명은?

- ① 생성 시스템의 인터프리터에 해당된다.
- ② 규칙 해석기는 현재의 사실을 바탕으로 적용할 규칙들을 찾는다.
- ③ 스케줄러는 규칙들이 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.

47. 전문가 시스템 개발도구에 대한 설명이 올바른 것은?

- ① 문제 지향 언어 : 추론기관을 내장하고 있다.
- ② 기호처리 언어 : 과학 및 상업 분야의 수치 계산에 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 요소들 중 문제 분야의 지식 베이스를 제외한 나머지 요소들로 구성된 전문가 시스템 구성도구를 골격 시스템이라고 한다. 지식 베이스만 구축함으로써 손쉽게 전문가 시스템을 구성할 수 있다는 것이 장점이다.

48. 다음 중 지능형 에이전트의 특성에 해당되지 않는 것은?

- ① 지속적으로 실행하는 데몬(demon)과 같은 프로세스이다.
- ② 사용자의 지시에 따라 수동적으로 행동한다.
- ③ 주도권을 가지고 목표 지향적으로 행동한다.

- ④ 사용자 및 다른 에이전트와 상호작용을 할 수 있다.

[정답] ②

[해설] 지능형 에이전트는 사용자를 위해 자율적으로 행동할 수 있다. 즉, 사용자의 허가를 받지 않고, 심지어는 통지하지도 않고 자율적으로 행동하기도 한다.

49. 숙고형 에이전트가 반응형 에이전트와 다른 점은?

- ① 행동에 따른 결과를 예상하여 행동을 결정한다.
- ② 주변 환경의 상태를 인식한다.
- ③ 효과기를 통해 결정된 행동을 실행하여 환경을 변화시킨다.
- ④ ‘조건-행위’ 규칙을 사용한다.

[정답] ①

[해설] 숙고형 에이전트는 자신의 행동에 대한 결과를 예측하고, 그 예측과 목적을 고려하여 행동을 결정한다.

50. 에이전트의 일반적 특성에 해당되는 것은?

- ① 사용자를 위해 자율적으로 행동
- ② 주변 환경의 변화에 반응하지 않음
- ③ 일회성 작업 처리
- ④ 사용자나 다른 에이전트로부터의 격리

[정답] ①

[해설] 에이전트는 자율성, 적응성, 능동성, 사회성, 지속성 등의 특성을 갖는다.

51. 다음 중 숙고형 에이전트에 대한 설명은?

- ① 행동에 의한 결과를 예상하여 바람직한 행동을 선택한다.
- ② 에이전트가 존재하는 세계에 대한 모델이 불필요하다.
- ③ 실세계의 현재 신호에만 근거하여 행동한다.
- ④ 어떠한 내부 상태도 가지고 있지 않다.

[정답] ①

[해설] 에이전트는 자신이 존재하는 세계에 대한 모델과 자신의 행동이 그러한 세계에 미치는 영향에 대한 모델을 가지고 있어야 한다. 일련의 행동들은 이러한 모델 하에서 시뮬레이션을 통하여 안전하고 효과적이라는 것이 밝혀진 후 실행된다.

52. 다음 중 에이전트가 학습을 하는 경로로 적합하지 않은 것은?

- ① 사용자의 피드백 ② 사용자의 활동 관찰
- ③ 타 에이전트의 충고 ④ 무작위 행동 선택

[정답] ④

[해설] 에이전트는 사용자의 행위를 관찰하거나 사용자의 피드백, 사용자가 제시한 사례, 다른 에이전트의 충고를 통해 학습한다.

53. 반응형 에이전트란?

- ① 현재의 내부 상태와 외부의 입력을 처리하여 곧바로 행동을 결정한다.
- ② 지식을 사용하지 않는다.
- ③ 행동에 대한 결과를 예상하여 목표에 도달하는 최적의 행동을 선택한다.
- ④ 소속되어 있는 환경을 변화시키지 않는다.

[정답] ①

[해설] 반응형 에이전트는 숙고형 에이전트와는 달리 실세계에 대한 모델을 가지고 행동의 결과를 예측하지 않고, 실세계로부터 센서를 통해 감지한 신호 그 자체를 처리함으로써 곧바로 행동을 결정한다.

54. 개인 웹 에이전트의 특성은?

- ① 사용자와의 상호작용을 통해 사용자의 취향을 파악하여 개인에게 적합한 정보를 제공한다.
- ② 모든 사용자에게 동일한 정보를 제공한다.
- ③ 모든 사용자의 취향을 하나의 프로파일에 저장한다.
- ④ 학습기능이 없는 정적인 에이전트이다.

[정답] ①

[해설] 웹 에이전트는 사용자에게 필요한 정보를 사용자의 요구와 취향에 맞게 제공함으로써 유용한 정보를 효과적으로 취할 수 있게 한다

55. 패턴 인식 단계 중 크기, 위치 등이 변형된 인식대상 패턴을 학습시의 기준 패턴과 같은 조건으로 변환하는 과정은?

- ① 관측 ② 표본화
- ③ 양자화 ④ 정규화

[정답] ④

[해설] 관측은 대상 패턴을 입력 장치를 통하여 취득하는 것, 표본화는 연속적인 신호를 이산적 신호로 변환하는 것, 양자화는 각각의 표본값들을 디지털 값으로 변환하는 과정을 의미한다.

56. 패턴 인식을 위한 특징 추출의 목적에 해당되는 것은?

- ① 여러 유형의 패턴에 대해 유사한 값을 얻기 위해
- ② 특징 계산을 위한 알고리즘의 시간 복잡도를 높이기 위해
- ③ 식별에 도움이 되는 정보만을 취하여 인식에 사용하기 위해
- ④ 아날로그 신호를 디지털 신호로 변환하기 위해

[정답] ③

[해설] 특징은 인식대상의 패턴들 중에서 어떠한 패턴을 다른 패턴들과 구분하기 위해 사용될 수 있는 정보를 추출한 것으로, 원 데이터 대신 특징을 추출하여 사용하는 이유는 패턴 공간의 차수를 낮추어 처리의 효율성을 높이고, 판별작업에 더욱 적합하도록 의미있는 값을 얻고자 하는 것이다.

57. k-NN에 대한 올바른 설명은?

- ① 언어이론적 패턴 인식 방법이다.
- ② k 개의 값으로 이루어진 특징을 이용한 인식 방법이다.
- ③ 학습표본 중 인식대상과 가장 가까운 k 개의 표본을 구한다.
- ④ 식별함수가 1개의 선형함수로 표현되는 경우 사용한다.

[정답] ③

[해설] k-NN은 결정이론적 패턴 인식 방법 중 하나로, 미지의 패턴과 가장 거리가 가까운 k개의 표본을 취한 후, 선택된 표본이 어느 클래스에 가장 많이 속하는가에 따라 판정한다.

58. 언어이론적 패턴 인식 시스템에서 사용되는 문법의 구성요소들 중 인식대상 패턴을 나타내는 것은?

- ① 비단말기호 ② 단말기호
③ 생성규칙 ④ 시작기호

[정답] ④

[해설] 문법을 구성하는 요소들 중 비단말 기호는 몇 개의 원시패턴들이 결합된 부분패턴, 단말 기호는 원시패턴, 시작기호는 인식 대상 패턴이 된다.

59. 패턴 a 와 b 사이의 도시 블록 거리에 해당되는 것은? (단, d 는 특징 공간의 차수, x_{ik} 는 패턴 k 의 i 번째 특징값이다.)

- $$\begin{array}{ll} \textcircled{1} & \left\{ \sum_{i=1}^d (x_{ia} - x_{ib})^2 \right\}^{1/2} \\ \textcircled{2} & \sum_{i=1}^d (x_{ia} \oplus x_{ib}) \\ \textcircled{3} & \sum_{i=1}^d (x_{ia} - x_{ib}) \\ \textcircled{4} & \sum_{i=1}^d |x_{ia} - x_{ib}| \end{array}$$

[정답] ④

[해설] 도시 블록 거리는 특징 공간의 각 차원별 거리의 합산으로 정의된다. ①은 유클리드 거리, ②는 특징값이 0 또는 1 중 하나일 경우에 사용하는 해밍 거리이다.

60. 패턴 인식에 사용할 특징의 조건으로 적절하지 않은 것은?

- ① 계산에 의해 구할 수 있는 것
- ② 서로 다른 패턴은 서로 다른 값을 갖는 것
- ③ 처리해야 할 데이터의 양을 늘릴 수 있는 것
- ④ 패턴의 유용한 정보를 잃지 않는 것

[정답] ③

[해설] 특징은 계산에 의해 구할 수 있어야 하고, 다른 패턴과의 구분에 도움이 되는 것이어야 하며, 유용한 정보를 잃지 않으면서 데이터를 처리할 수 있는 양으로 줄일 수 있어야 한다.

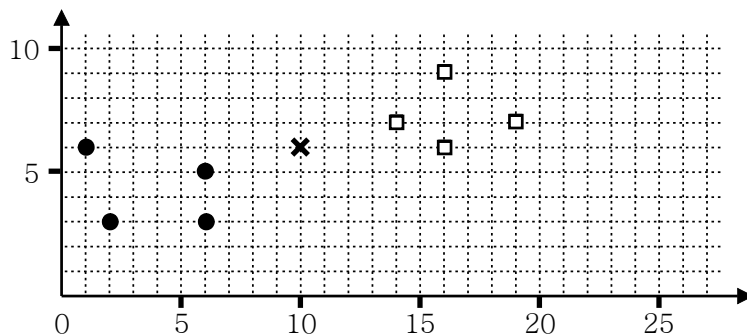
61. 계량공간이 만족해야 하는 거리 측정자 J 의 조건에 해당되는 것은?

- ① $J[x, y] \leq 0$
- ② $J[x, y] = 0 \text{ iff } x=y$
- ③ $J[x, x] \neq 0$
- ④ $J[x, y] = J[x, z] + J[z, y]$

[정답] ②

[해설] 거리는 음수가 될 수 없으므로 $J[x, y] \geq 0$ 이어야 하고, 자기 자신과의 거리는 0이므로 $J[x, x] = 0$ 이어야 한다. 두 점 사이의 거리는 제3의 점을 거쳐 가는 거리보다는 길지 않아야 하므로 $J[x, y] \leq J[x, z] + J[z, y]$ 이다.

62. k -NN을 인식방법으로 사용할 때, 거리 측정자가 (ㄱ) 도시 블록 거리를 사용하는 경우와 (ㄴ) 유클리드 거리를 사용하는 경우 x 를 어느 패턴으로 인식하게 되는가? (단, $k=3$)



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

[정답] ②

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

63. 자연어 문장 이해 과정의 올바른 설명은?

- ① 구조 분석 후 형태 분석을 행한다.
- ② 구조 분석이 의미분석과 함께 실행되기도 한다.
- ③ 하나의 문장은 하나의 의미모형을 갖는다.
- ④ 구조 분석에는 정규문법을 사용한다.

[정답] ②

[해설] 문장의 구조 분석은 문장 구성 요소들의 문법적 구조를 분석하여 적합한 모형을 찾는 것이고, 의미 분석은 문장 요소들의 의미를 파악하는 것이다. 자연어의 경우 문장의 의미정보를 배제한 상태에서는 완전한 구조 분석이 어려운 경우가 있으며, 이 두 과정이 결합되어 처리되기도 한다.

64. 격 문법의 용도는 무엇인가?

- ① 문장의 의미를 파악하기 위한 문법 모델이다.
- ② 정규 문법으로 문법을 정의하기 위한 것이다.
- ③ 두 문장 사이의 관계를 파악하기 위한 것이다.
- ④ 문장이 내포하는 실제 의미를 파악하기 위한 것이다.

[정답] ①

[해설] 격 문법은 문장 요소들을 행위의 주체, 대상, 목표 등 문장 요소의 격 구조에 따라 의미를 파악할 수 있도록 하는 문법 모델이다.

65. 자연어 이해 시스템에 대한 올바른 설명은?

- ① 문장 중에서 하나의 단어를 바꾸더라도 문장의 문법적 구조는 변화하지 않는다.
- ② 하나의 문장이 여러 가지 목적 표현으로 변환될 수도 있다.
- ③ 자연어 문장은 정규 문법으로 생성된다.
- ④ 한 문장에서 쓰인 단어는 다른 문장에서도 같은 의미를 갖는다.

[정답] ②

[해설] 자연어는 여러 개의 문장이 동일한 의미를 갖기도 하고, 하나의 문장이 여러 의미를 가질 수도 있다. 후자의 경우는 한 문장이 여러 가지 목적 표현으로 변환될 수 있다.

66. 한 단어가 여러 가지 의미를 가지고 있을 때 문장에 맞는 의미를 선택하기 위해 사용되는 것은?

- ① 의미 특징표 ② 정규 문법
③ 파스 트리 ④ 개념 의존도 모형

[정답] ①

[해설] 의미 특징표란 사전의 각 단어에 그 단어가 갖는 의미에 대한 특징을 첨부하여 놓은 것이다. 예를 들면 ‘배’의 경우 ‘신체의 일부’, ‘운송수단’, ‘음식물’과 같은 서로 다른 의미 특징이 해당되는 단어의 의미에 첨부될 수 있다.

67. 다음의 자연어 이해의 처리과정 중 각각의 단어, 구두점 등을 분리시키는 처리를 의미하는 것은?

- ① 형태 분석 ② 구조 분석
③ 어의 분석 ④ 문장 통합

[정답] ①

[해설] 구조 분석은 문장 구성 요소들의 문법적 구조를 분석하여 적합한 모형을 찾는 과정,어의 분석은 문장 요소들의 의미를 파악하는 과정, 문장 통합은 인접 문장의 의미에 따른 해석을 하는 과정이다.

68. 다음 중에서 학습과 관련된 추론방법은 어느 것인가?

- ① 유도법 ② 귀납법
③ 연역법 ④ 삼단논법

[정답] ②

[해설] 귀납법은 공통적인 내용을 표현하는 사실들로부터 일반적 규칙을 찾아내는 추론 방법으로, 이러한 과정은 학습과 밀접한 관계가 있다.

69. 신경회로망에 대한 올바른 설명은?

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

[정답] ③

[해설] 신경회로망은 단순한 기능을 하는 수많은 뉴런들이 상호간에 방대한 연결을 유지하며 대 단위 병렬처리를 하는 시스템이다. 정보는 뉴런들 사이의 신경 연결에 분산 저장되며, 학습기능을

통해 신경 연결의 연결 강도를 조정할 수 있다.

70. 퍼셉트론 모델에 대한 올바른 설명은?

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

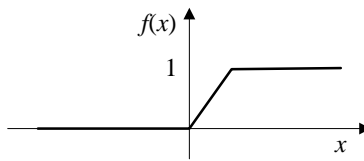
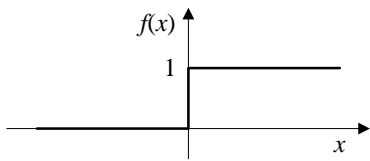
[정답] ②

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

71. 오차 역전도 모델에서 사용하는 활성화함수는?

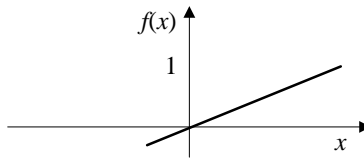
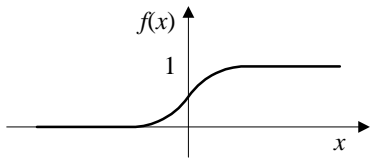
①

②



③

④



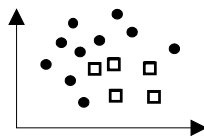
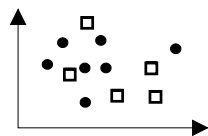
[정답] ③

[해설] 오차 역전도 모델에서는 sigmoid 함수를 사용한다. sigmoid 함수는 미분 가능하며, 자동 이득 조절 특성이 있는 함수이다.

72. 다음의 패턴 분포 중 단층 퍼셉트론으로 학습할 수 있는 것은?

①

②



③

④



[정답] ④

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

73. 오차 역전도 모델에 대한 설명으로 올바른 것은?

- ① 비지도 학습을 한다.
- ② XOR 문제를 학습할 수 없다.
- ③ 뉴런의 활성화함수는 sigmoid를 사용한다.
- ④ 단층 퍼셉트론에서만 사용될 수 있는 학습 모델이다.

[정답] ③

[해설] 오차 역전도 모델은 비선형 결정경계를 갖는 경우에도 학습할 수 있는 다층 신경망의 학습방법이다. 지도 학습을 사용하며, 활성화함수는 sigmoid를 사용한다.

74. 신경회로망에서 정보가 저장되는 형태는?

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

[정답] ③

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

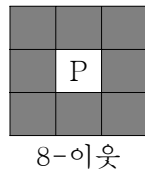
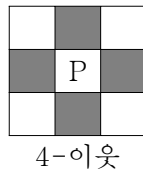
75. 다음 중 4-이웃 연결성을 적용할 때 픽셀 P의 이웃 픽셀들은?

- 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.



76. 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 $\frac{1}{8} \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$

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

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 라플라시안(Laplacian), ③과 ④는 소벨(Sobel) 연산자이다.

77. 다음 중 2차 미분에 의해 물체의 에지를 검출하는 것은?

① 소벨 연산자

② 라플라시안(Laplacian) 필터

③ 분할과 합병

④ 가우시안(Gaussian) 필터

[정답] ②

[해설] 소벨 연산자는 1차 미분으로 에지를 검출하는 연산자이며, 가우시안 필터는 평활화 필터이다.

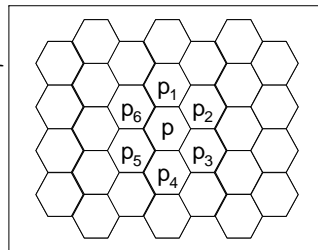
78. 그림과 같은 육각형 픽셀 구조의 장점은?

① 데이터 양이 적다.

② 어떠한 픽셀의 인접 픽셀들이 모두 거리가 같다.

③ 하드웨어 구현이 쉽다.

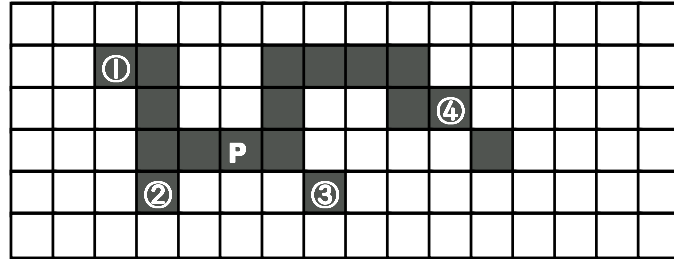
④ 픽셀 간 거리 계산이 쉽다.



[정답] ②

[해설] 육각형 픽셀 구조는 이웃 픽셀들이 모두 같은 거리를 갖는다. 설계의 편의상 일반적으로 사용하는 사각형 픽셀 구조는 인접한 픽셀들의 거리가 모두 같지 않다는 단점이 있다.

79. 사각형 픽셀 구조에서 4-이웃 연결성을 사용할 경우 픽셀 p로부터 연결되는 경로가 존재하는 픽셀이 아닌 것은?



[정답] ③

[해설] 4-이웃 연결성의 경우는 상, 하, 좌, 우 4개의 픽셀을 이웃 픽셀로 정의한 것이다. 그러한 관점에서 ③은 연결되지 않은 독립된 픽셀이다.

80. 다음 중 1차 미분으로 수평 방향의 그레이레벨 변화를 검출하는 마스크의 형태는?

① $\frac{1}{8} \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$

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

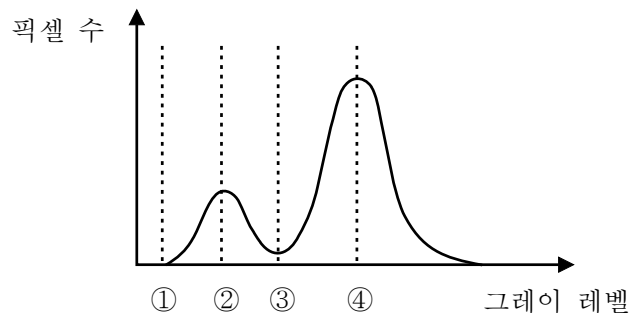
③ $\frac{1}{4} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$

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

[정답] ③

[해설] 중앙 3개의 픽셀을 중심으로 우측 3개의 픽셀 가중치 합을 좌측 3개의 픽셀 가중치 합으로 빼는 형태이므로 수평방향의 그레이레벨 변화를 검출할 수 있다.

81. 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.