

1주차 강의 - (03.03-목)

신나는 오리엔테이션

- 간단한 전체 대면 신나죠~?
- 교재 : 엑셀로 배우는 인공지능 (교수님이 번역한 책><)
- 중간/기말 : 필기(40%, 45%)
- 출석 : 5%
- 팀프로젝트 : 10%

- * 팀프로젝트는 1~3인으로 구성 가능
- * 출석점수는 없다. 출미만 되지 않도록!

무엇을 배울까?

- 인공지능의 기초 개념과 동작 원리
- 인공지능 분야로 진출을 희망한다면 들었을 때 도움이 된다.
- 엑셀을 이용해서 진행해볼 것이다.

중간광고 타임

- 2022년 데이터 분석 기반 인공지능 교육
- 정보산업진흥원 주최
- 인공지능 기반 기초 코딩 교육(160 시간)
- 1차교육 : 03.07 ~ 05.20
- 기초통계학, 인공지능 기초, 비지도 학습 및 클러스터링, 유전자 알고리즘 기초, 인공지능 기반 영상처리, 기계 학습 기초
- 2차교육 : 07.04 ~ 09.05
- 기계 학습 관련 통계 및 확률, 강화 학습, 인공지능경망 기초 및 유전자 알고리즘, 확률 이론과 클러스터링 알고리즘, 데이터 분석 기반 인공지능 알고리즘(FCM, 퍼지, 흠필드), 기계학습 알고리즘 (의사 결정 트리, 베이지안 확률/네트워크)

- * 한 달 반 동안 160시간
- * 여러 번 진행하는 것 같아도 다들 잘 안 들어서 다 비슷한 내용
- * 동의대 410호에서 진행

두런두런 이야기

- 유전자 알고리즘이란?
 - : 입력값의 상태에 따라 나온 결과를 가중치의 곱에 따라 유전을 하며 발전하는 프로그램
 - > <https://www.youtube.com/watch?v=8V2sX9BhAW8>
- 스네이크 게임의 코드를 실행시켜서 보여주심
 - > 몸에 부딪히는 충돌 처리에 대한 학습을 위해 포인트를 먹었을 때 늘어나는 몸의 길이를 길게 함
 - > 보통 프로그램 2개를 돌린다 (진행상황 보이는 거, 안 보이는 거)
 - > 니트 알고리즘 이용
- 교수님의 (특강) 최종 목표는 스네이크 게임

1주차 강의 - (03.08-화)

강의 목적

- 인공지능 개요, 역사
 - 인공지능 최신 이슈
 - 인공지능 핵심 요소 기술
 - 각 인공지능 적합 적용 분야
- => 본인의 연구분야에 해결 문제에 적합한 인공지능 선택 가능

Q1 인공지능이 인간을 능가할 수 있을까?

- 인공지능 : Artificial Intelligence
- 정의 : 인간의 지적 성질을 인공적으로 재현시키려는 것
- 인공지능 연구란 : 속도/규모/정확성에서 인간 지능의 한계를 가진 문제를 해결하기 위한 H/W, S/W 개발

인공지능이 극도로 발전해도 인간의 능력을 뛰어넘지 못할 것이다.

=> 이런 오해를 불식시킬 수 있도록 연구 동향을 알아보자

- * 40~50년은 그럴 것이라고 보고 있다고 한다.
- * 현재는 3차 인공지능 붐
- * 영화추천 : Bicentennial man, A.I., 아이로봇, 이글아이, 엑스마키나

Q1 인공지능이 인간을 능가할 수 있을까?

- 우리 주위에도 컴퓨터는 넘치고 인공지능 탑재 가전은 많다
 - > ex. 의류 종류 파악 세탁기, 인간 움직임을 따르는 조명 등
 - 가장 친근한 우리 인공지능~! 스마트폰
 - 이런 똑똑이를 IoT(사물인터넷, Internet Of Things)
- => 데이터의 관련성과 의미를 파악하기 위해 인공지능이 필요

- * 우리 라비다가 운영자가 카카오 취업해서 죽었어요ㅠㅠ
- * 교수님은 음악을 바로 찾아주는 시리의 알고리즘이 궁금했어요
- * 어떻게 그렇게 짧은 시간듣고 빨리 찾아내지?
- * 예시 제품 Google nest temperature sensor, Fitbit

기계 발달과 그 의미

- 산업혁명
 - > 산업혁명의 기계화에도 “기계가 인간을 몰아냈는가”에 대한 논의X
 - > 산업혁명 초기에 직업 상실 문제가 있었되, 곧 사회적 동의 획득
 - > 중장비 : 팔의 움직임 모방 but 특수 목적에 최적화 된 인간과 같은 구조일 필요 X

=> “중장비가 인간 팔보다 쓸모있다”고 해서 비관하는 사람은 없음
- 인공지능
 - > 인공지능 : 인간의 뇌 활동을 모방 but 일부 활동에 특화
 - > 손발 보강과 머리를 보강하는 것은 다르지 않는가?
 - > 기계한테 팔씨름 지는 것이 큰 일인가?

누구의 책임인가

인공지능이 탑재된 자동운전

Q1. 사고가 나면 누가 책임지는가?

- 1) 제조사
- 2) 인공지능?ㅋㅋㅋㅋ
- 3) 운전자
- 4) 운행회사

=> 기술적 발전은 빠르지만 법률적인 규정이 따라가지 못 한다

+ 의료 로봇이 실수를 범한다면 누구의 책임?

+ 제조회사, 병원, 의사, 환자?

=> 아무리 인공지능이 발달해도 보장능력이 있는 인간이 책임질 것

* 일본은 기술적으로는 아니라도 법률적 준비를 잘 하고 있대요

* 혈 갈변 염색 삼푸 법적인 문제로 판매 불가능 울 엄마 사줄했는데

일의 질적 변화

- 인공지능에 의해 직업이 사라지지X, 일하는 방식변화를 가져올 것
- 사라지거나 반으로 주는 것은 있겠지만 목적/책임 관점에서 유지
 - > 기계가 책임 질 수는 없기 때문
- 목적/책임 관점으로 보면 반드시 인간이 해야하는 일이 있을 것

=> 비관적인 생각X, 질적 변화를 생각하고 미래의 직업을 생각하길



* 스웨덴의 아인라이드라는 무인 자율주행 트럭이 이미 운행 중

* 운전석이 없어서 효율적인 공간 생성 가능

2주차 강의 - (03.10-목)

빅데이터 분석

- 정의 : 대량의 데이터로부터 규칙성을 발견해내는 것
- DB의 데이터 분석 : 사전 지식을 갖고 계획을 갖고 분석
- 빅데이터 분석 : 이게 데이터인가? 여기서 뭘 꺼낼 수 있나? 정도
- 데이터의 요인 중 중요 요소를 추출하고 분류요인을 규칙화 하는 것
- 클러스터링 : 분류 기준을 인간이 지정했기 때문에 정확한 분석이 보장되지 않음

- 기존 학습 기법

- > 영상 인식 : 개/고양이 인식을 위해 구분 데이터를 사람이 제공
- > 데이터의 특징을 사람이 직접 일일이 알려 준 상태로 경향 분석
- => 번거롭고 시간이 걸림 == 정밀도와 편리성에서의 한계
- => 딥러닝(deep learning)이 새로운 전개 맞이

딥러닝(Deep learning)

- 목적 : 주어진 데이터의 특징을 자동 추출
- 인간이 분류하지X 특징 추출 및 특징에 따른 데이터 정리가 가능
- 주 사용처 : 빅데이터에 안성맞춤
 - > 학습 요인을 주지 않아도 자율적으로 타당 요인 도출
 - > 대량의 미지 데이터에서도 잠재 법칙 발견
 - > 새로운 현상에 해당 법칙을 적용해 최적의 해 제시

=> “빅데이터에서의 예측”이라는 성과가 나옴

- 구글의 고양이 인식



-->



--> cat!

> 고양이 특징을 인간이 부여하지 않음

> 딥러닝이 자동으로 생성해 새로운 영상의 동물을 고양이로 판단

> Purely Unsupervised Feature Learning in Image

기술적 특이점

- 1) 미국 서해안의 경찰은 과거 범죄 데이터를 이용해 순찰구역을 결정
 - 2) 한 가수는 과거 히트곡 분석으로 적합한 곡을 찾아내어 활동
 - 3) 싸이의 강남 스타일이 유행 후 해당 풍을 따라한 daddy 발매
- + 하루의 행동 스케줄, 연애 상대 등 다양하게 예측에 활용 중

=> 2045년에는 모든 측면에서 인공지능이 인간의 지능을 능가할 것

- 인공지능 예측에 따라 인간이 행동하게 될 것

- 이를 거스르는 것이 사회 질서를 어지럽히는 것

=> 인간이 서투르게 개입하지 않는 것이 나은 시대



- 기술적 특이점(technological singularity) :
인공지능이 인간의 뇌를 넘어서는 시점

> 미래학자 레이 커즈와일이 정의한 말

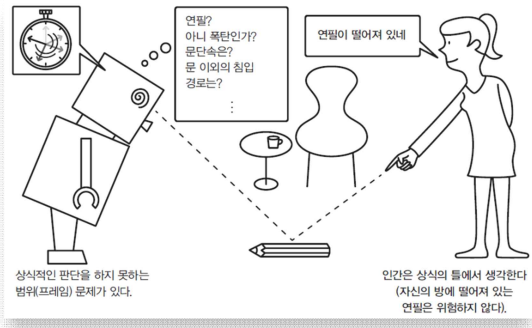
> 실제 음성을 표현하기 위한 칩개발로 유명

+ 영창 전자피아노 칩은 다 이 사람 꺼



2022.03.11. 오후 6시 - 이세돌 2집 거울볼 발매!!

프레임 문제



- 문제 대응법 : 기존의 **상식**을 이용
- 컴퓨터가 **상식**을 학습 하기 위해서
 - > 거의 무한에 가까운 방대한 사실들과 현상을 파악
 - > 컴퓨터가 얻는 지식인 법률, 의학 이런 정보에서는 뛰어남
 - > 일반적인 상황에는 대처가 좋지만 학습하지 못 한 상황엔 오류
 - > 컴퓨터는 상식을 넣을 수 없다.
- **사람은 상식**이 있기에 대처한다
 - > 인간은 그럼 무한한 상식을 가졌는가?
 - > 그럼에도 인간은 자연스럽게 상식적 판단을 해 낸다.
- 인공지능을 논할 때 옛날부터 지적받은 성가신 과제
- 자율적으로 행동해도 계속 남을 과제
- 한계가 있어도 도움 되는 걸로 충분하다고 볼 수 있다.

=> 인공지능은 **프로그래밍된 유한의 상식 범위 내에 문제 해결 가능**

인공지능의 목적

- 인공지능은 인간의 지적 활동을 공학적으로 재현해 한계를 넘길 것
- 인간 두뇌를 만드는게 목적은 아니나 더욱 지적인 활동 가능성

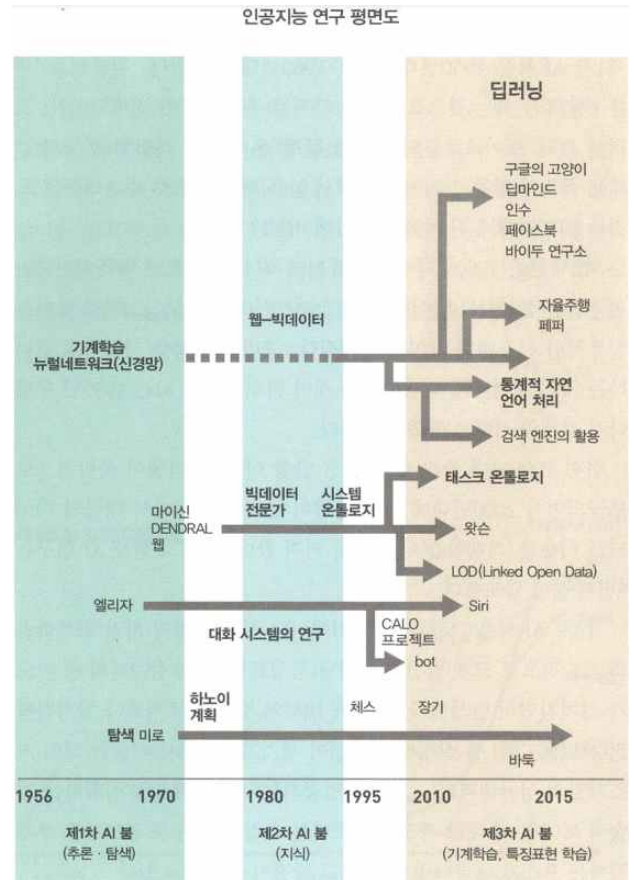
=> 인간의 뇌에 가까운 인공지능은 의미가 있는가?

- 오히려 사람처럼, 동요/실수/망각의 일이 벌어질 수 있음
- 인간답지만 기계에 기대하는 특성을 잃을 수 있음
 - > 정확성, 속도, 재현성, 양적 한계가 없는 측면
 - > 재현성 : 항상 동일 결과를 도출하는 것

=> 뇌에 가까운 인공지능이라는 표현은 의미X

=> 인간의 지적 활동을 어떤 측면에서 보충할까 하는 측면

인공지능 연구 발전 과정



- 인공지능의 발전은 컴퓨터의 발전에 영향을 받음

- 알고리즘의 시대 -> 지식의 시대 ->

- 1차 인공지능 붐

- > 알고리즘 개발에 집중적으로 진행
- > 알고리즘을 잘 짜면 사람의 일을 대체할 수 있을 것
- => 알고리즘의 한계를 느낌

- 2차 인공지능 붐

- > 사람에게만 있지만 프로그램에는 없는 것을 넣자 : 지식, 정보
- > 인간의 지식을 어떻게 숫자, 문자로만 표현할까
- > 지식 공학자의 등장
- => 계속 데이터를 지정하고 정리하는 것의 번거로움

- 3차 인공지능 붐

- > 사람이 가르치지 않아도 스스로 결과를 도출

* 현재(2022년)는 3차 인공지능 붐이 시작되고 4~5년 이후

인공지능 붐

- 2019년 오바마 대통령, 뇌신경과학 프로젝트 발표
 - > BRAIN(Brain Research through Advancing Innovative Neurotechnologies) Initiative
 - > 선충의 신경세포, 파리, 쥐로 확대해 영장류까지 확대
 - > 최종적으로는 인간 뇌지도도를 만드는 것
- 2013년 유럽의 인간 뇌 프로젝트(Human Brain Project)
- 일본의 동경대 입시 합격 로봇 프로젝트

저번 시간

● 체험해 봅시다: MC 문제

- M(선교사)과 C(식인종)의 인원수 및 배의 정원을 변경하여 다양한 조합에서 상태 전이가 이루어지는 모습을 보면, **계산 문제와는 다르게 문제의 규모에 따라 복잡성이 증가하는 것은 아니라는 것을 알 수 있음.**
- 상태 전이를 머릿속에서만 생각하면 혼란스럽지만 하나씩 상태를 조사해 나가면 해가 있는 경우에는 반드시 성공하므로 이 개념은 복잡해 보이는 문제에 유용.

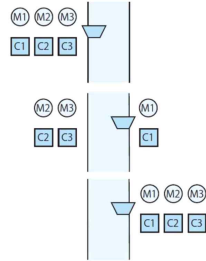


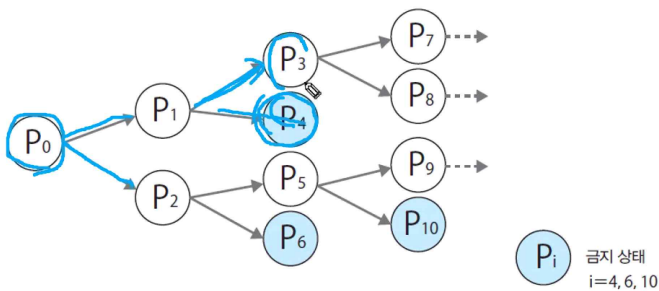
그림 5-1 MC 문제

탐색 트리(Search Tree)

- 상태 전이를 효율적으로 수행하기 위해서
 - ▶ 전략이 필요
 - ▶ 각 상태에서 주어지는 연산자를 어떻게 적용할지 기준 필요

탐색 트리(Search Tree)

- 시간 축에 따라 전개되는 상태 전이 모습은 초기 상태를 루트로 하는 트리구조로 표현 가능
- 문제 해결 : 탐색 트리를 루트부터 말단 노드까지 **가장 효율적으로** 쫓아가는 것



- 여러 가지 상태를 시도하고, 최종 적인 경로 도출
- 복잡한 문제는 트리가 엄청 커지는데, 어떻게 빠르게 도출하지?

MC 문제

- 수식으로는 잘 표현 못 하지만 동적인 문제의 대표적 예
- 재귀적으로 풀 수 없음
 - ▶ 동일한 방식을 반복하는 문제가 아님
- 문제의 규모를 M, C 인원수는 3명, 배의 정원은 2명으로 해 모델화

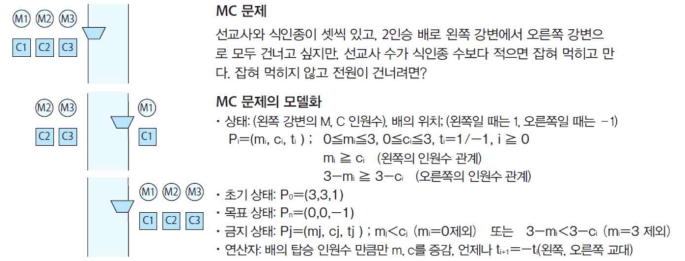


그림 5-4 MC 문제 모델화

- MC 문제 모델화

- 상태 : 가지의 속성을 갖는 순서쌍, 튜플, 리스트로 표현한다
 - 이진 개발자가 정해야한다.
 - 항상 왼쪽의 수만 신경쓴다.
 - $P_i = (m, c, t)$; 배의 위치; (왼쪽이면 1, 오른쪽이면 -1)
 - 규칙 (제약조건) : $m \geq c, 3-m \geq 3-c$

상태를 {왼쪽 강변의 M, C 인원수, 배의 위치(1/-1)}로 나타내고 수식화하면 다음과 같음.

$$P_i = (m_i, c_i, t_i)$$

$$0 \leq m_i \leq 3, 0 \leq c_i \leq 3, t_i = 1(\text{왼쪽}) \text{ 또는 } -1(\text{오른쪽})$$

$$m_i \geq c_i \quad (\text{왼쪽 강변에서의 사람 수 제약})$$

$$3 - m_i \geq 3 - c_i \quad (\text{오른쪽 강변에서의 사람 수 제약})$$

초기 상태 $P_0 = (3, 3, 1)$, 목표 상태 $P_n = (0, 0, -1)$

연산자는 배의 탑승 인원수 {M의 인원수, C의 인원수}만으로 상태(왼쪽)의 인원수가 증감.

- 초기 상태 : $P_0 = (3, 3, 1)$
- 목표 상태 : $P_n = (0, 0, -1)$
- 금지상태 = $P_j = (m_j, c_j, t_j) : m_i < c_i$ ($m_i=0$ 제외) / $3-m_i < 3-c_i$ ($m_i=3$ 제외)
- 연산자 : 배의 탑승 인원수 만큼만 m, c를 증감.
 언제나 $t_{i+1} = -t_i$ (왼쪽, 오른쪽 교대)

- * m : 선교사
- * c : 식인종

MC 문제 : 수식

$$\alpha: M \text{이 한 명만 이동} \quad m_{i+1} = m_i - t_i, t_{i+1} = -t_i$$

$$t_i = 1 \text{일 때는 } 1 \leq m_i \leq 3, t_i = -1 \text{일 때는 } 0 \leq m_i \leq 2$$

$$\beta: C \text{가 한 명만 이동} \quad c_{i+1} = c_i - t_i, t_{i+1} = -t_i$$

$$t_i = 1 \text{일 때는 } 1 \leq c_i \leq 3, t_i = -1 \text{일 때는 } 0 \leq c_i \leq 2$$

$$\gamma: M \text{이 두 명 이동} \quad m_{i+1} = m_i - 2t_i, t_{i+1} = -t_i$$

$$t_i = 1 \text{일 때는 } 2 \leq m_i \leq 3, t_i = -1 \text{일 때는 } 0 \leq m_i \leq 1$$

$$\delta: C \text{가 두 명 이동} \quad c_{i+1} = c_i - 2t_i, t_{i+1} = -t_i$$

$$t_i = 1 \text{일 때는 } 2 \leq c_i \leq 3, t_i = -1 \text{일 때는 } 0 \leq c_i \leq 1$$

$$\psi: M, C \text{가 한 명씩 이동} \quad m_{i+1} = m_i - t_i, c_{i+1} = c_i - t_i, t_{i+1} = -t_i$$

$$t_i = 1 \text{일 때는 } 1 \leq m_i \leq 3 \text{ \& } 1 \leq c_i \leq 3$$

$$t_i = -1 \text{일 때는 } 0 \leq m_i \leq 2 \text{ \& } 0 \leq c_i \leq 2$$

- a

- ▶ 배가 왼쪽에 있을 때는 선교사가 한 명 이상 있어야 한다.
- ▶ 배가 오른쪽에 있을 때는 식인종은 0~2명 있어야 한다.

MC 문제 : 결론

- 정의

- > 패턴 : {m}, {c}, {mm}, {cc}, {mc}
- > 위의 한 가지 패턴으로 인원 수 증감

- 조건

- ▶ 강변의 인원수 이하만 탈 수 있다.
- ▶ 최소 한 사람은 타야함
- ▶ 왼쪽 강변 인원에 주목하면 앞 페이지 5가지 연산 가능

- 상태 정의

- > 왼쪽 강변에서 $m \geq c$ 이면,
 - > 오른쪽 강변에서 반대로 $3-m \leq 3-c$ 가 될 것 같으므로
 - > 양쪽 강변에서 선교사 쪽 이 많다는 것이 불가사의?
 - > 다음과 같은 금지 상태를 생각하면 납득할 수 있음.
- 1) 금지 상태 $P_j = (m_j, c_j, t_j)$
 - 2) $m_j < c_j$ (단, $m_j = 0$ 제외) 또는 $3 - m_j < 3 - c_j$ (단, $m_j = 3$ 제외)

- 이 예에 대한 탐색 트리는 금지 상태를 제외하고 이전으로 돌아 가는 중복 상태도 제거하면, 적용 가능한 연산자가 거의 한 가지로 결정되므로 가지가 넓게 퍼지지 않고 쉽게 구축할 수 있음.
- 그림은 트리 형태가 되진 않았지만 왼쪽 끝을 루트로 해서 왼쪽에서 오른쪽 방향으로 가지가 펼쳐지는 것으로 생각하기 바람.
- 이 경우에는 금지 상태를 제외하면 항상 한 가지이므로 이 상태에 관한 트리를 전개함.

$I=0$ L	1 R	2 L	3 R	4 L	5 R	6 L	7 R	8 L	9 R	10 L	11 R
(3,3,1)	(3,2,-1)	(3,3,1)	(3,1,-1)	(3,2,1)	(3,0,-1)	(3,1,1)	(2,1,-1)	(2,2,1)	(0,2,-1)	(2,1,1)	(0,1,-1)
	(3,1,-1)	(3,2,1)	(3,0,-1)	(3,1,1)	(2,1,-1)	(2,2,1)	(2,0,-1)	(1,3,1)	(0,1,-1)	(1,1,1)	(0,0,-1)
	(2,3,-1)	(2,3,1)	(2,2,-1)		(2,0,-1)	(2,1,1)	(1,2,-1)	(1,2,1)		(0,3,1)	
	(2,2,-1)		(2,1,-1)		(1,1,-1)	(1,3,1)	(1,1,-1)	(0,3,1)		(0,2,1)	
	(1,3,-1)		(1,2,-1)			(1,2,1)	(0,2,-1)			(0,2,1)	

※ 배는 교대로 왼쪽(L)이나 오른쪽(R)에 있다. ■가 상태 전이, ■는 금지 상태, ■는 중복(이전으로 돌아감)

그림 5-5 MC 문제의 탐색 트리

연산자

$$a : m_{i+1} = m_i - t_i \quad (M \text{ 1명 이동})$$

$$\beta : c_{i+1} = c_i - t_i \quad (C \text{ 1명 이동})$$

$$y : m_{i+1} = m_i - 2t_i \quad (M \text{ 2명 이동})$$

$$\delta : c_{i+1} = c_i - 2t_i \quad (C \text{ 2명 이동})$$

$$\psi : m_{i+1} = m_i - t_i \quad \& \quad c_{i+1} = c_i - t_i$$

(M, C 1명씩 이동)

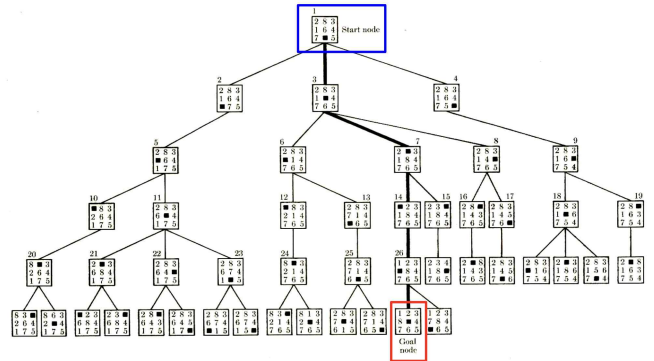
언제나 $t_{i+1} = -t_i$ (좌우 교대)

* https://github.com/oMFDoo/LearnAIwithExcel/blob/master/Ex6_MC%20%EBC%88%EC%A0%9C.xlsx

실행하고 놀았다

강의 왼쪽 강변에서 오른쪽 강변으로 같은 수의 M, C를 이용한다.	0	1	2	3	4	5	6
단, 왼쪽 강변, 오른쪽 강변, 배의 어디에서도 M 인접수가 0 이 인접된다.	1	10	11				
1. [인접수] 2에 M, C의 인접수 3(2-6, default:3), (강좌) 칸에 배의 정원(2-6, default:2)을 입			20	21	22		
2. [Step] 칸이 1인 경우는 1번 이동할 때마다 멈춘다. 이하면 현재 실행(default:0)			30	31	32	33	
3. 초기화 버튼을 누르면 M, C 인접수와 배의 위치가 초기 상태로 표시된다.			40	41	42		
4. 실행 버튼을 누르면, 연산자(오른쪽 표) 자동적으로 적용되어 상태 전이가 표시된다.			50	51			
5. [회시] 칸에 무사히 이동하면 회시 숫자, 무사히 이동하면 회시 'NG'가 표시된다.			60	61			

가장 효율적인 경로를 어떻게 선택할까?



- <https://cse.buffalo.edu/~rapaport/572/S02/8-puzzle.html>
- 시계방향으로 숫자를 정렬하는 최적의 방법은 무엇일까?

탐색법

- 문제 해결에 있어서 상태 공간 내에서의 초기 상태에서 목표 상태를 이르는 경로를 결정하는 방법
- 이 때까지는 제약과 금지상태만 고려했지만 이제는 아래 것도 고려

도달보증	탐색이 순환하지 않고 어디선가 멈추는 것. 멈춘 곳이 해인지 아닌지는 관계 없음.
최적해	목표 상태에 도달하는 것.
국소해	목표 상태 이외의 곳에 도달하는 것.
최적경로	경로비용을 고려할 때 목표상태에 이르는 경로의 전체 비용이 최소가 되는 경로.
누적비용	각 상태에 도달하기까지의 경로 비용(실제 값)
장래비용	각 상태에서 목표 상태까지의 경로 비용(추측 값)

- 최적해
 - > 목표상태가 제일 좋은 방안인 것
- 국소해
 - > vV 같은 형태가 있으면, 제일 낮은 결과를 도출
 - > 미분 개념을 이용한다
 - > 시작지점에 따라 결과 도출이 다를 수 있다.
 - > 왼쪽에서 시작하면 파란 구간을 국소해, 오른쪽이면 초록색을 도출
 - > 따라서 랜덤 위치로 탐색을 한다.
- * 많이 나와요 이 경우가
- 최적경로
 - > 경로의 비용이 다른 경우 비용을 고려해 비용이 최고인 경로 도출

- **장래비용**
 - > 목표 상태까지의 경로 비용을 추측하는 것
 - > 예측된 장래 비용이 작은 순으로 탐색
 - > 현재 바라보는 대상에 대한 가능성을 파악하는 것
- ex) 8-puzzle : 지금 이 상황은 한 10번은 움직여야겠다.
- > 문제점 극복 : 깊이가 10개 만 되어도 뒤질게 10만개 이상 모두 뒤질 수 없기 때문에 장래비용을 도입한 것
- * 이게 참 중요하건데

탐색법 전략에 따른 분류

1) 맹목적 탐색

- 적용 가능한 연산자를 무작위로 선택
- 중복을 무시
- 해가 있어도 반드시 도달 보장 못함
- 난수로 진행방향 결정, 일상의 실제 문제에서 유사
- * 영국 박물관 알고리즘과 유사 : 너무 많아서 발길 닿는대로 간다

2) 체계적 탐색

- 모든 상태공간을 중복없이 조사.
- 해가 있으면 반드시 도달
- 너무 비효율적
- ex) DFS, BFS, 분기한정법

3) 휴리스틱 탐색

- 경험 법칙에 따른 상태공간 축소 수행
- 효율적 탐색 도모
- * heuristic : 경험적인

=> 어떤 평가기준을 정하고 평가값이 좋은 쪽을 먼저 고려
=> 평가기준 : 평가함수 도입

- 평가함수

: 과거의 실제 비용(누적 비용)과 장래 예측 비용(장래 비용)의 합계

> $f(n) = e(n) + h(n)$

> $e(n)$: 초기 상태부터 상태 n 까지의 **최소라고 생각하는** 실제 비용

> $h(n)$: 상태 n 부터 목표상태까지 예상되는 최소 비용,
알 수 없는 경우에는 한 단계 앞의 비용을 사용

* h 휴리스틱의 약어

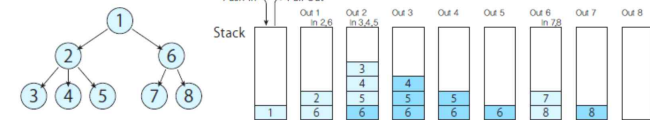
체계적 탐색

- 전체 경로로 빠짐없이 모두 조사 한다
- 시간이 너무 걸려 비현실적이지만, 가장 좋은 경로를 구할 수 있음
- 깊이우선 탐색
- 너비우선 탐색
- 분기한정법

체계적 탐색 : 깊이우선 탐색

: 스택을 사용한 전체 경로 탐색

깊이우선 탐색

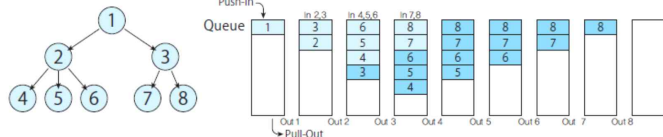


- 비용은 생각하지 않으므로 평가함수 없음

체계적 탐색 : 너비우선 탐색

: 큐를 사용한 전체 경로 탐색

너비우선 탐색

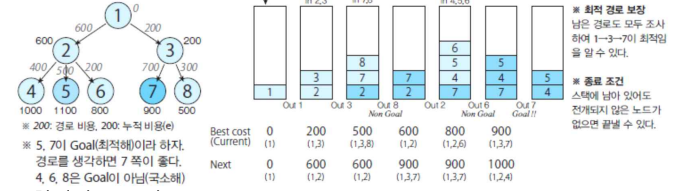


- 비용은 생각하지 않으므로 평가함수 없음

체계적 탐색 : 분기 한정법

: 전체경로 탐색, 누적 비용이 적은 방향으로 진행

분기 한정법



- 최적경로 보장

- 평가함수 : $f(n) = e(n)$

- 위 사진에서 : 1-3-8-2-6-7로 탐색하게 됨

* 시험문제에서 분기 한정으로 되지면 어떤 순서로 도출되나 물어요~

3주차 강의 - (03.24-목)

각 탐색의 장/단점

- 깊이 우선

- > 진행하는 방향이 맞으면 빠름
- > 밑으로 더 진행하지 못하게 되면 되돌림이 필요

- 너비우선

- > 전체적으로 되돌림이 없음
- > 모두 전개하면서 진행하기 때문에 속도가 느림

- 분기 한정

- : 비용이 적은 경로를 선택을 위해 비용을 정의 후 매번 평가
- > 비용의 정의는 평가 함수에 따라 이루어지지만, 누적 비용만 사용.
- > $f(n) = e(n) + h(n)$
- > 최적 경로를 구하는 가장 확실한 방법
- > 여기저기 건너뛰어 돌아다니므로 현실적이지는 못함



여백은 이세들이 채웠대구~!

휴리스틱 탐색

- : 누적비용 외에 **장래비용도 예측해** 탐색 효율을 높임
- > 장기, 체스와 같은건 분기 써도 너무 볼 거 많아서 등장
- > 구체적 비용 예측법 X, 경험치/기대치/보수 등이 있단걸 전제

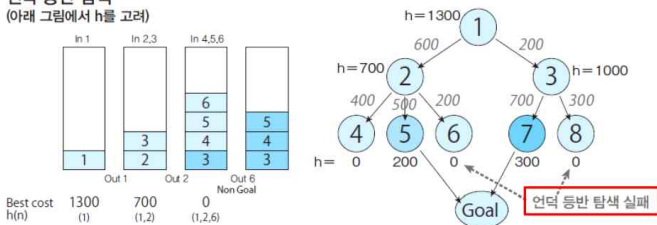
- 종류 : 언덕 등반 탐색, 최고 우선 탐색, A알고리즘, A*알고리즘

언덕 등반 탐색	$f(n) = h(n)$
	<ul style="list-style-type: none"> 자기보다 높은 위치로 가는 느낌 올라갈거라 생각해서 갔더니 내려가는 길이라도 돌아가지 않음 장래 비용이 적은 방향으로 진행 되돌림 없음 최적해 보장 없음
최고 우선 탐색	$f(n) = h(n)$
	<ul style="list-style-type: none"> 언덕등반의 개선편 장래 비용이 적은 방향으로 진행 다른 상태도 확인하며 진행 최적해 보장
A 알고리즘	$f(n) = e(n) + h(n)$
	<ul style="list-style-type: none"> 과거의 경로를 기억 다른 상태도 확인 최적 경로 보장
A* 알고리즘	$f(n) = e(n) + h(n)$
	<ul style="list-style-type: none"> 장래 비용 제약 설정 => A알고리즘 안정화 최적경로 보장

- * 이것도 잘 알아두세요~
- * 언덕 등반과 최고 우선은 묶어서 기억하세요~
- * A알고리즘 보다는 A*알고리즘을 알아두세요~

언덕 등반 탐색

언덕 등반 탐색
(아래 그림에서 h를 고려)



특징

- 평가 함수로 장래 비용만 사용
- 과거 경로 기억하지 않음
- 등산하는 것처럼 정상만을 보고 오르는 것과 같음
- 머신러닝에서 대부분 사용하는 기법
- 실제 사람이 보통 쓰는 방법

단점

- 장래비용이 잘 되어있으면 잘 찾는다.
- BUT 장래비용은 추측값. 경로를 그려쳐서 국소해에 빠질 가능성O

장점

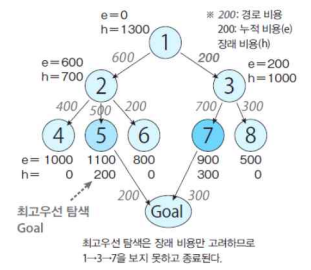
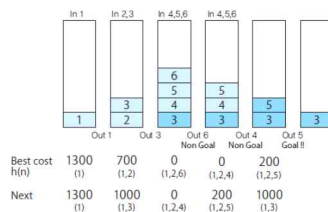
- 한정된 시간, 판단 기준 내에서 진행하기에는 유효한 방법

탐색

- h : 장래비용, 작으면 좋다, 어떻게 정하는진 지금 몰라도 됨.
- 숫자 : 경로 비용
- 경로 : 1 - 2 - 4(답이 아니네)
- 이렇게 없으면 답 못 찾고 끝난다~

최고 우선 탐색 (Best First Search)

최고우선 탐색



등장배경

- 국소해에 빠졌을 때 다른 상태를 다시 조사하도록 개선

동작

- 장래 비용만 사용
- 과거 경로 기억 X
- 개선점 : 아직 조사하지 않은 상태도 포함해 평가 함수를 비교

개선점

- 장래 비용보다 조사하지 않은 상태에 대한 장래 비용 쪽이 적으면
- 현재 경로를 일단 중단, 장래 비용이 가장 적은 다른 상태에서 진행

장점

- 장래 비용만을 고려하고 있음에도 불구하고,
- 어떤 상태에서도 적어도 그 시점에서는 가장 좋은 경로를 선택
- 국소해에 빠져도 조사하지 않은 상태 중에서 장래 비용이 가장 적은
- 상태에서 다시 시작하는 것이 가능하므로 반드시 목표 상태에 도달

탐색

- 경로 : 1(2, 3) -> 2(4, 5, 6) -> 4(6과 순서 바뀌어도 됨, 답X) -> 6(볼 수 있는 거 중 제일 값 작음, 답X) -> 5 -> goal!!

단점

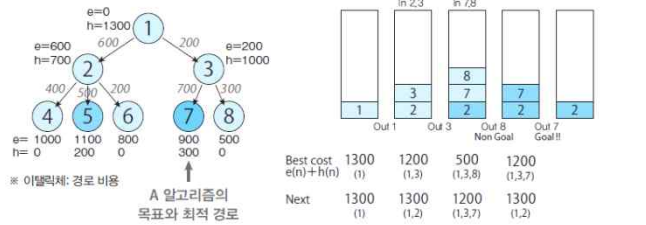
- 너무 많이 내려가서 누적비용이 너무 커지는 경우가 발생
- 실제로 1-3-7 갔으면 바로 답나왔는데 둘러쌌음.
- 최적해이 but 최적 경로라는 보장 X
- => 비용도 최소로 하고 싶음
- => A 알고리즘의 등장, 누적비용 적용



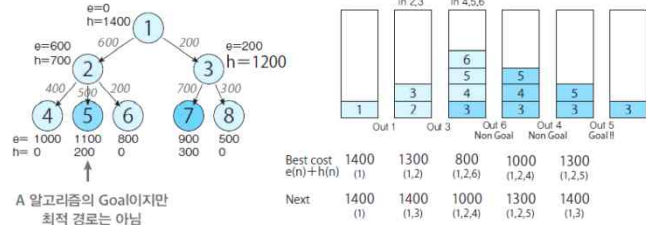
르르망 Scientist 노래 최고다~!

A 알고리즘(A Algorithm)

$f(n)=e(n)+h(n)$ 단, $e(n)$: 노드 n 까지의 누적 비용, $h(n)$: 노드 n 부터 목표까지의 예측 비용



$h(n)$ 을 과대하게 설정하면? 예를 들어, ⑤에서 $h=1200$ 이라고 하면?



- 평가함수 : 누적 비용과 장래 비용의 합계를 사용.

동작

- 과거 경로를 기억
- 각 상태에서 조사하지 않은 모든 상태들도 포함해 평가 함수를 비교
- 가장 값이 적은 방향으로 진행

장점

- 항상 가장 좋은 경로를 선택
- 반드시 목표 상태에 최적 경로로 도달

단점

- 여기저기서 경로를 중단하고 바꿔 가며 진행한다
- 장래 비용을 추가한 것 때문에 분기 한정법으로 수행했던 것처럼, 모든 경로를 건너뛰어 다니는 것은 아니므로 현실적

경로

- 가정 : 7이 제일 좋은 경로
- e : 경로 비용의 누적
- h : 장래 비용

- 1번 경로 : 1 -> 3($e + h = 1200$) -> 8($e+h = 500$ 답X) -> 7(되돌아가 2, 7 중 $e+h$ 가 더 작음) -> Goal!!

- h 값을 과다하게 설정
- 2번 경로 : 1 -> 2($e+h = 1300$) -> 6(4, 5, 6중에 제일 작음) -> 5(4도 답이 아니었음, 3,5랑 비교) -> Goal!!

h 값을 과다하게 설정한 경우

- > 결국 그럴듯한 값만 탐색하게 됨
- > e 의 의미가 퇴색됨. => e 가 영향을 거의 못 주기 때문
- > 결국 또 깊이깊이 내려가버림
- > e 와 h 값이 차가 너무 크지는 않도록 설정

* 큐도 스택도 아니기 때문에 오른쪽 스택은 무시

A* 알고리즘

: A 알고리즘의 장래 비용에 제약 조건을 설정

$$f(n) = e(n) + h(n)$$

$$h(n) \leq h^*(n)$$

$$h^*(n) : n \text{부터 장래의 진정한 최소 비용}$$

- 제약조건 설정 시 영향

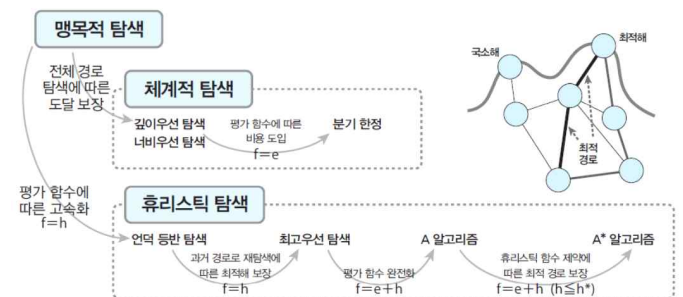
- > 어떤 상태도 거기부터 앞으로의 진정한 비용 이하의 상태가 선택
- > 목표 상태에 도달 시 진정한 비용을 유지하는 것이 가능
- > 목표 상태 도달 시 : 누적 비용과 같은 값으로 확정값

=> 최적해(목표 상태 도달) 및 최적 경로를 완전히 보장할 수 있음.

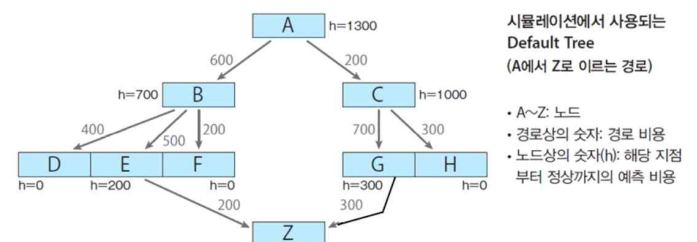
이런 느낌

- 진정한 비용을 어떻게 설정하지
- $h(n)$ 을 좀 보수적으로 잡자
- 답인 거 같긴 한데 과하게 내려왔으니까 둘러보고 오자
- * 진정한 비용 : 과장되지 않은 상태로 진짜 답이 될 듯한 비용

탐색법 정리



복습해보아요!



- 깊이우선 : A-B-D-E-Z-F-C-G-H
- 경로비용 : A-C-H-B-F-G-D-E-Z(e 가 1200인 Z), 도중 답있다면 끝
- 언덕등반 : A-B-DorF
- 최고우선 : A-B-D-F-E-Z
- A* : A-C-H-G-Z

* z의 h 를 0으로 가정

독자가

● 선공

○ 후공

컴퓨터 전략

○ 없음

○ 1수 앞

○ 2α

○ 3α

● 3β

횟수

Max

13

13

7기점수

초기화

속행

점수

74

독자

76

평가값

-2

Winner

게임

1

2

3

4

5

6

7

8

9

10

11

12

13

독자

12

11

10

9

8

2

3

4

5

6

7

13

1

<- 독자의 카드 일체 칸

나머지

1

2

3

4

5

6

7

8

9

10

11

12

13

<- 독자의 남은 카드

컴퓨터

2

3

4

5

6

7

8

9

10

11

1

12

13

<- 컴퓨터의 카드

나머지

1

2

3

4

5

6

7

8

9

10

11

12

13

<- 컴퓨터의 남은 카드

점수 이력

독자

5

14

21

26

29

33

37

41

45

49

53

65

76

<- 독자의 점수 이력

컴퓨터

10

18

24

28

30

35

40

45

50

55

61

62

74

<- 컴퓨터의 점수 이력

- 선공 최초로 이김><
- αβ 전략 시뮬레이션을 실행
- 상호간의 수와 상태를 아는 것을 전제로 함.

게임 전략

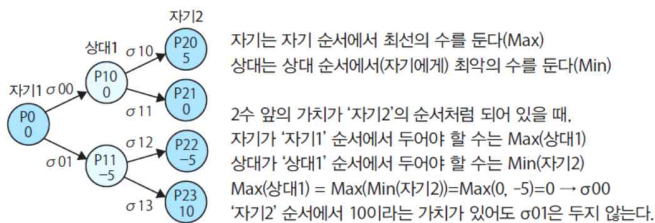
- 턴제로 진행되어 입장이 반대가 되는 개념의 게임
- 대전형 게임에서의 효과를 크게 본 기법
- Min-Max 전략
- αβ 전략

Min-Max 전략

: 내 순서에서는 최대, 남 순서에는 최소의 평가값 상태를 교대로 취함

- 게임 트리의 상태 전이

- > $f = e * h$: 여기서는 h가 더 중요
- > 자기 순서에는 선택 가지중 최선(평가값 최대)인 상태 취함
- > 상대 순서는 선택 가지 중 상대의 최선(나에게겐 최악)의 상태 취함



- 내가 둘 수 있는 수가 2개다.
- 위로 두면 평가 함수 값이 0, 아래면 평가 함수 값이 -5
> 위가 더 좋아보인다.
- 상대<->나가 바뀐다면
> 상대는 P20과 P21중에 P21을 고를 것
> 평가함수는 내 입장에서 좋은 것을 고른 거기 때문에

- 단점

- > 여러 수 앞까지의 게임 트리를 생각하기만 해도,
선택 가지 수의 급증 차수로 가지가 확장되므로 탐색 공간이 방대
- > 모든 가지를 조사해 가장 좋은 수를 취하는 건 어렵

αβ 전략

- 배경 : 조사하는 가지의 개수를 줄이는 것을 고려
- 조정 : 관심이 있는건 Min, Max 값뿐. 그 이외의 평가값 상태는 버림

- 가지치기(pruning) : 다음 상태를 평가하지 않고 버리는 방법

> α 가지치기 :

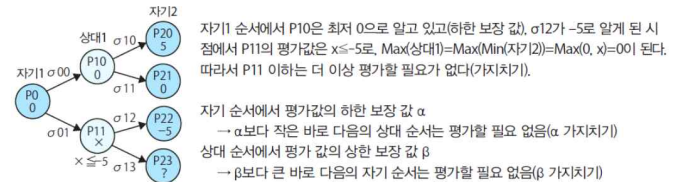
자기 순서에 하한 보장 값 α보다 작은 다음 상대 순서 노드를 버림

> β 가지치기 :

상대 순서에 상한 보장 값 β보다 큰 다음 자기 순서 노드를 버림

> 내 입장이냐 상대입장이냐? -> 잘 구분 안 하고 뭉쳐서 부름

> 내다본 수의 단계가 많을수록 가지치기는 효과적

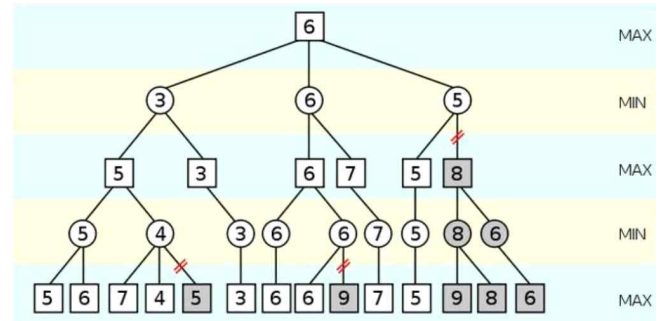


- P10을 보았다 상대 입장에서는 자기가 유리한 P21을 할거다

> 평가값의 최악은 0이다.

- P11을 보았다 상대 입장에서는 자기가 유리한 P22를 할거다

- > 벌써부터 -5다. 더 펼쳐봤자 일단 이전 탐색결과인 0보다 나쁘다.
- > 만약 P23이 100이든 1000이든 상대는 -5인 P22를 선택할 것
- > P11은 평가값이 -5이하로 평가값이 0인 P10보다 나쁘다.
- > 가지치기



- 이거 널 겁니다.
- 완전 다른 형태로 새로운 유형으로 널 겁니다~

- DFS와 유사

1) 6-3-5-5-5or6 => 상대는 5가 유리하기에 5를 둘것임

> 현재 최악은 5

2) 6-3-5-4-7or4or5 => 상대는 4가 유리하기에 4를 둘것임

> 현재 최악은 4

> 이전(5) 보다 최악(4)인 4를 발견하였기에 나머지는 안 굴

> 나머지 노드는 가지치기

3) 6-3-3-3-3 => 상대는 3이 유리하기에 3을 둘 것임

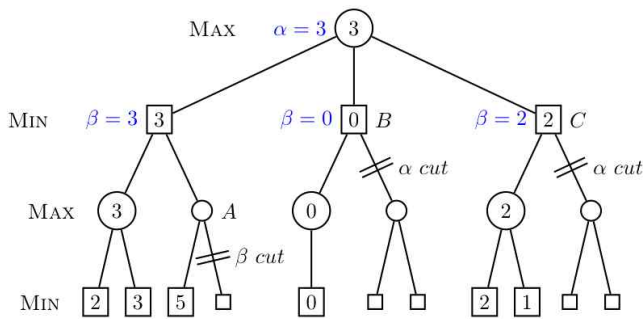
> 따라서 5 VS 3 (깊이 3) 이기에 나는 5쪽을 택해야함

...

- 단점

- > 가지치기가 실패하는 경우 존재
- > 수가 진행되고 미리 예상한 평가값이 잘못된 경우 돌아가기 불가
- > 전체 국면을 미리 다 파악 못하면 가지치기 실패는 피할 수 없음

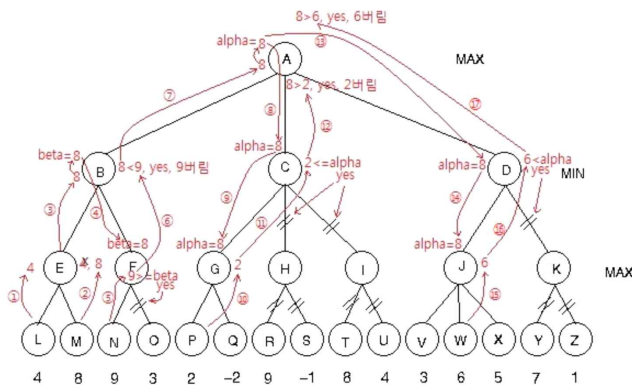
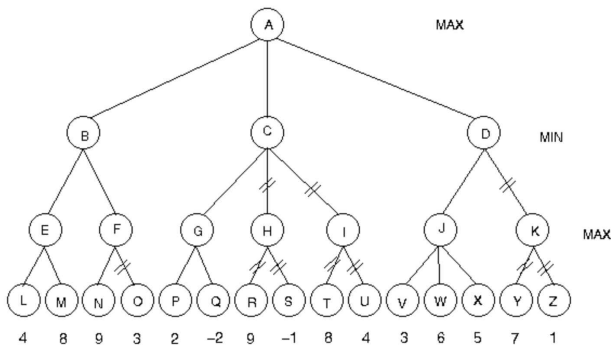
αβ 전략 : Alpha-Beta pruning



- ○ : 내 입장
- □ : 상대입장
- α : max에서 가지는 값
- β : min에서 가지는 값

- 각 단계에서 나의 입장인 max 상대입장인 min을 번갈아가며 선택
- max 노드에서 β컷 : 값이 β보다 크거나 같을 때 (Value >= β)
- min 노드에서 α컷 : 값이 α보다 작거나 같을 때 (Value <= α)

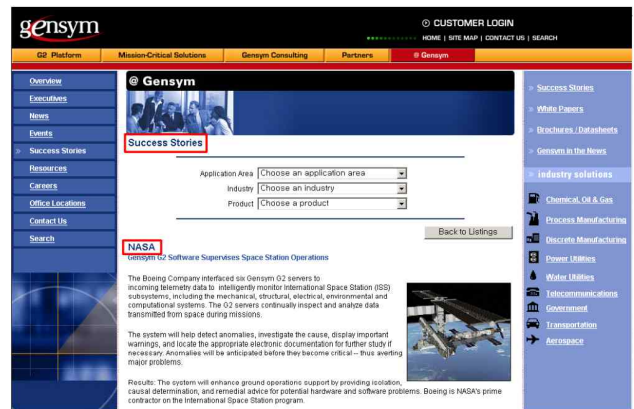
- * 시험에 나오니까 한 번 더 볼게요
- * 연습용 그림(<https://micropilot.tistory.com/2973>)



인간의 지식을 컴퓨터에서 표현

- 상식을 컴퓨터에게 어떻게 표현할까?
- ex) 이순신 장군님 폰번호 알아? => 당연히 안되는 걸 사람은 안다.

인간의 지식을 컴퓨터에서 표현 : Gensym



- success stories : 여기의 소프트웨어로 성공한 스토리
- > 도요타 : 생성 공정을 g2 소프트웨어로 가동
- > ISS(국제우주정거장) : NASA의 ISS 전체 운영 시스템에 적용

* 전시스라고 읽어오~

실습하기: 질병 진단 전문가 시스템

Ex.10 질병 진단 전문가 시스템(전향 추론) Copyright © 2015 ASAI Noboru

목표: 규칙과 문진표를 1을 단위로 독립되게 배워 넣으면 또 다른 시스템이 되는 것을 실험할 수 있음.

사용 방법: ① 초기화 버튼을 누른 후 문진표에 대한(임의의) 문진표를 하고 ② 진단 버튼을 누른 후 증명명 리스트에 결과 표시

주의: 문진표의 1열에 있는 번호와 생성 규칙의 행 번호는 대응 관계다. 변경할 때에는 이 대응 관계를 유지해야 한다. Step 0

문진표	예	아니오	모름	병명 리스트
1 열이 있습니까?	예	아니오	모름	1 감기
2 38°이상의 고열입니까?	예	아니오	모름	2 인플루엔자
3 38°미만의 저열입니까?	예	아니오	모름	3 중이염
4 두통이 있습니까?	예	아니오	모름	4 인후염
5 머리가 속속신 아릅니까?	예	아니오	모름	5 뇌졸중
6 머리가 갑자기 심하게 아릅니까?	예	아니오	모름	6 편두통
7 오한이 납니까?	예	아니오	모름	7 패혈
8 공맥잡(나른함)이 있습니까?	예	아니오	모름	8 기관지염
9 기침을 할니까?	예	아니오	모름	9 간염
10 가래가 나오니까?	예	아니오	모름	10 폐렴/결핵

- 자신의 증상에 '1'이라고 적어서 체크
- '진단' 버튼을 누르면 진단을 해줌

* https://github.com/loMFOO/LearnAIwithExcel/blob/master/Ex10_10_%EC%A7%88%EB%93%91%20%EC%A7%84%EB%88%A8%20E5.xlsm

지식 표현

- 대표적인 지식 표현

1) 생성 시스템

- > 지식을 사물의 인과관계로 생각
- > IF-THAN 규칙으로 표현

2) 의미망

- > 지식을 사물의 관계로 봄
- > 속성이 달린 네트워크로 표현
- > 네트워크 그래프의 형태를 띤다.

3) 프레임 모델

- > 지식을 속성이 있는 사물로 본다.
- > 사물을 프레임으로 표현
- > 프레임 : 클래스 같은 거
- ex) 삼각형 프레임 { 변이 3개, 3각의 합은 180 ... }

- * 술어표현, 절차표현도 존재 but 위의 3개만 보겠다.
- * 이런 모델들은 인공지능 보단 소프트웨어의 발전을 이끌었다.
- * 인공지능 위인? : 마빈 민스키, 허버트 사이먼, 앨런 뉴얼

6주차 강의 - (04.12-화)

시험시험하자

- 21일 강의 시간에 배운 만큼 시험
- 지식 표현을 어떻게 할 것인가?

생성 시스템

- 지식 : 사물/현상 a, b에 대해 'a면 b'같은 인과 관계로 봄
- a : 조건이나 원인, 조건부
- b : 결과나 행동, 결론부
- 생성 규칙 : 아래와 같은 표현
 - > 하나씩 독립적으로 주어짐, 선언적 - 연결성이 없다.
 - > 추가, 갱신이 용이
 - > 결론부에 복잡한 처리도 서술 가능한 유연성이 높은 표현 방법
 - > 모순이 없도록 조심

IF a THEN b 또는 $a \rightarrow b$

- 지식 베이스 : 다수의 생성 규칙 집합으로 구성
- 사실 : 지식과 달리, 실제 환경으로부터 얻어지는 관측 데이터
- 지식의 사용
 - 1) 사실에 일치하는 조건부를 가진 생성 규칙을 찾는다.
 - 2) 그것의 결론부를 실행
- * 계속 변화하는 환경에 적응하지 못 함

생성 시스템

- : 지식 표현으로 생성 규칙을 이용하고 처리하는 기능을 갖춘 시스템
- 일반적인 구성 : 추론 기관의 역할
 - 1) 조건과 사실을 조회
 - 2) 경합 해소
 - 3) 행동 & 사실 갱신
- 경합해소 : 1)d에서 조건이 사실과 일치한 생성 규칙이 여러 개 있을 때 실행해야, 할 행동을 하나 선택하는 것

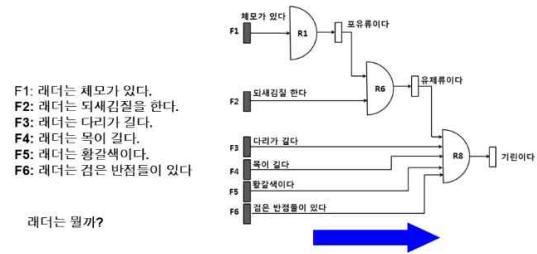
경합해소 기법

- 1) First Match: 처음 발견된 규칙을 선택
 - > 빠른 처리가 특징
- 2) Rule Priority: 규칙에 우선순위를 부여 후 순위가 높은 것을 선택
- 3) 최신 사실 우선: 최근에 처리된 사실과 일치하는 규칙을 선택
 - > 지식이 들어온 날을 기록하고 제일 최근 것이 좋다.
- 4) 구체 서술 우선: 가장 복잡한 조건을 갖는 규칙을 선택
 - > if의 조건이 많은 것이 좋은 것

* 1), 2)번은 많이 사용한다.

생성 시스템 : 추론 방향

1) 전향 추론(Forward Reasoning)



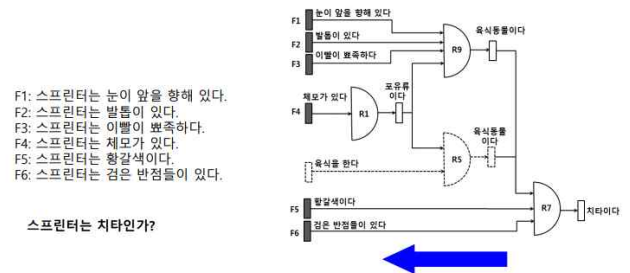
: 특정 사실로부터 출발하여 결론을 얻음.

> 데이터 구동(Data-Driven)형이라고도 함.

> 단점 : 지식이 너무 많으면 시간이 너무 많다

ex) P2 IF (고열이 있다) THEN (감기) or (인플루엔자)

2) 후향 추론(Backward Reasoning)



: 가설로부터 출발하여 특정 사실에 도달하게 되면 그 가설을 결론

> 목표 구동(Goal-Driven)형이라고도 함.

> 전향 추론의 보완

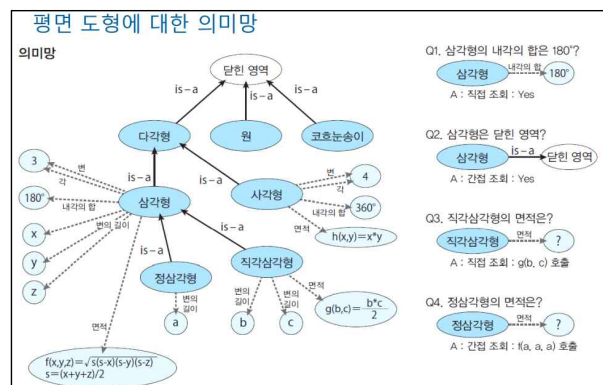
> 결론으로부터 조건을 탐색함

ex) IF (감기) THEN (몸이 나른하다) and (열이 있다) and (머리가 아프다) and (기침이 나온다) and ...

3) 쌍방향 추론

: 전향 추론으로 가설을 축소하고 후향 추론으로 가설을 검증하는 등 양방향의 특성을 모두 살림.

의미망(Semantic Network)



- 현상들 간의 관계를 네트워크로 표현

- 뇌의 기억 모델을 지식 표현에 적용

- 연결여부 뿐만 아니라 연결 이유와 관계의 의미를 부여하기도 함

- 단점 : 표현은 할 수 있지만, 조건 추적 알고리즘이 복잡

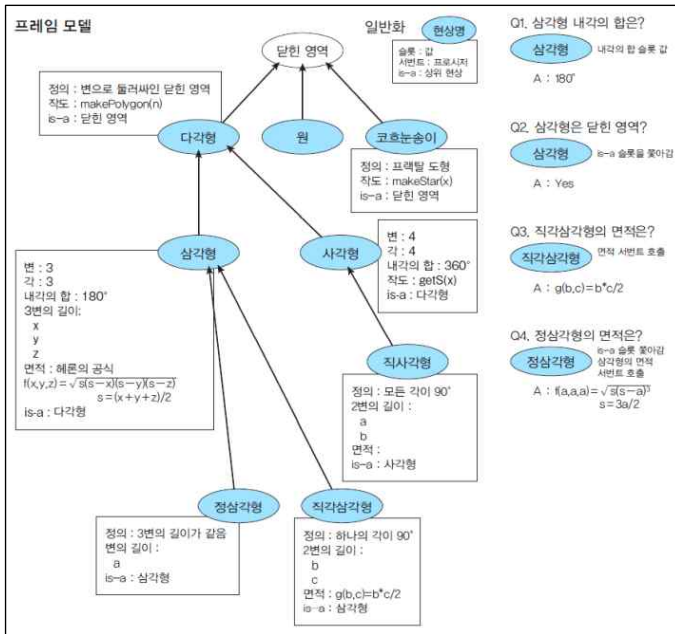
> 현상도 그 성질도 모두 동격으로 취급했기 때문

=> 의미망 : 위의 지식 표현 또는 이것을 처리하는 구조를 포함

* 한 때 많이 쓰다가 시들해진 기법

* 소프트웨어 발전에는 많은 영향을 주었다.

프레임 모델(Frame Model)



- 현상의 성질은 관련 속성들로 정리하여 현상 속에 모두 서술
- 속성을 한데 포함한 현상의 표현 방법
- 하나의 구조를 만들고 그 내부의 속성이 있다는 개념
- 클래스와 구조체와 유사

전문가 시스템(Expert System)

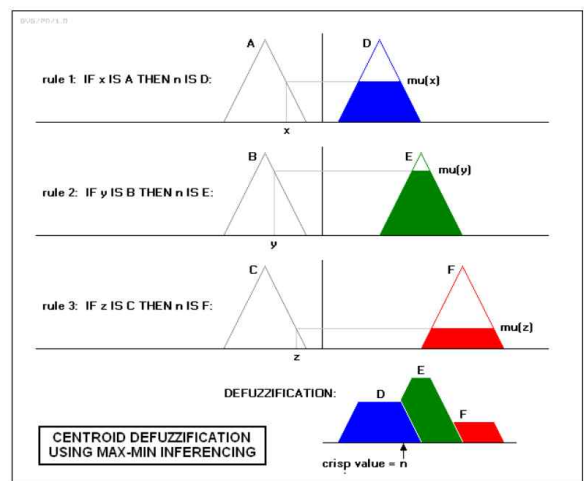
- 정의 : 지식 표현으로 전문가 지식을 컴퓨터로 처리하려는 시스템
- 역할
 - 1) 전문가 부족을 보완
 - 2) 지식 계승 으로서의 역할
 - 3) 위험한 작업 등을 대신하는 등
- 역사적인 전문가 시스템 : DENDRAL, MACSYMA, MYCIN
 - > 각 지질, 방정식, 마이신 항생제 처방을 위한 의료 역할
- MYCIN
 - > 전문가 시스템 구축 툴의 개념을 확립 => 추론 엔진만을 판매
 - > 지식 베이스 내용만 바꾸면 다양한 전문가 시스템을 구축할 수 있도록 개선
- EMYCIN (Empty MYCIN) : 의학기술을 뺀 마이신
 - > 이걸로부터 전문가 시스템 구축 툴의 발전 시작
 - > 1980년대 AI 붐이라고 할 수 있을 정도의 활황
 - > 각 기업이 상용화에 경쟁을 벌였음.
- * 90년대 이후로 쇠퇴해갔음
- 쇠퇴
 - > 인간의 상식까지 포함한 판단 기준에서 보면 도움이 되지 않음
 - > 법률 관계, 설비 스케줄링, 의료 등 비교적 확립된 분야 실용적
 - > 이용 범위를 잘 제한한다면 여전히 매우 유효한 개념

전문가 시스템의 유형

- 진단형 : 관측된 현상으로 원인을 추정.
ex) 의료 진단, 고장 진단 등
- 설계형 : 주어진 제약 조건 중 최적해를 제시.
ex) 반도체 칩 내의 배선, 건축 설계 등
- 제어형 : 센서 등의 관측 데이터로부터 최적의 제어를 수행.
ex) 화학 플랜트, 용광로, 지하철 등
- 상담형 : 요구를 만족하는 최적해를 제시.
ex) 법률 상담 등
- 교육형 : 학습자의 이해도에 따른 최적 지도를 수행.
ex) 지능적 CAI 등

7주차 강의 - (04.14-목)

퍼지 : 인간의 애매함을 컴퓨터로 처리하기



- 지식표현과 전문가 시스템을 연구할 때의 “애매한 지식”의 극복 대안
- 가능성/확률 ;
 - > 확률 : 사건이 발생할 수있는 정도
 - > 가능성 : 일어나거나 행해지는 능력
 - > 비가 올 가능성이 30%, 기준에 확실한 “확률”로 연구
- 확신도(CF) : 인간의 불확실성을 표현하는 척도
- 불확실은 다양한 유형이 존재
- 제어에 다수적으로 이용됨

- * Fuzzy : 보들보들한, 명확하지 않은
- * DST : 덤스터 삼프 이론 교수님 박사 논문
- * 표지에 있는 내용은 시험에도 꼭 나올 내용

퍼지 : 적용사례, 구현



- 1) 영국 판매용 삼성 세탁기
- 2) 막대기나 큐브를 쓰러지지 않게 세우고 있는 기계
- 3) 후면 주차가 가능한 공간을 찾아 주차하는 알고리즘

- * 일본에서 활발히 연구되어서 연구자에 일본인 많음

대회소개

- 임베디스 소프트웨어 경진대회
- 교육도 시키면서 대회도 진행
- 몇 달간 진행

일반 집합과 퍼지 집합

- 일반집합

$$A = \{x | x \text{에 대한 조건}\}$$

$$\begin{aligned} \text{예} \quad \text{짝수의 집합} \quad B &= \{x | \text{mod}(x, 2) = 0\} \\ \text{홀수의 집합} \quad C &= \{x | \text{mod}(x, 2) = 1\} \end{aligned}$$

- > 일반 집합 : 어떤 조건들을 만족하는 것들의 모임
- > 집합 요소 : 포함, 포함되지 않음 2가지 상황으로 정의

$$A = \{x | X_A(x) = 1\}$$

$$X_A(x) \rightarrow \{0, 1\}^{*1} \quad X_A(x) = 1 (x \in A) \quad \text{or} \quad X_A(x) = 0 (x \notin A)$$

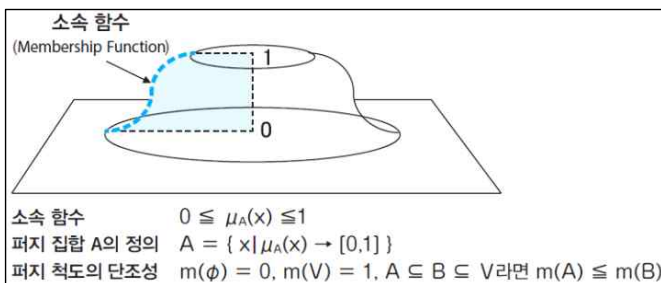
- > 현상의 애매성과 주관 표현을 다루기 위해 집합 정의를
- > $X_A(x)$ 는 인수 x 가 집합 A 의 요소이면 1, 아니면 0 반환
- > 크리스프(crisp) 집합 : 경계가 확실히 구분되어있는 집합

* x 로 쓰였지만 실제로 κ (카이)다.

- 퍼지집합

- > 등장 : 속함을 소수로 어느정도 속하는 느낌으로 내면 안되나?
- > 분류 : 1에 가까우면 집합 안쪽에 가깝, 0에 가까우면 바깥쪽
- > 퍼지(Fuzzy) 집합 : 경계가 모호한 집합

퍼지 집합(Fuzzy Set)의 개념



- 경계 폭 : 완전한 안쪽과 완전한 바깥쪽이 존재
- 경계 폭 값 : 0~1사이 값을 가진 소속함수로 보완
- 소속 함수 값이 높은 것은 정보의 정확도? 신뢰도?가 높은 느낌

참고 확률과의 차이점

주사위 던지기에서 2의 배수 또는 3의 배수가 나올 확률은 다음과 같다.
 $m(2 \text{의 배수}) + m(3 \text{의 배수}) - m(2 \text{와 } 3 \text{의 배수}) = 1/2 + 1/3 - 1/6 = 2/3$

퍼지의 경우에는 2의 배수나 3의 배수 중 어느 쪽이 될까?
 가능성이 높은 쪽(max)으로 한다면 2의 배수가 될 가능성이 높다.

- 퍼지집합 구분 : 집합이름 위 혹은 아래에 '~'을 넣어 구분한다.

* M : 소속함수를 나타내는 무

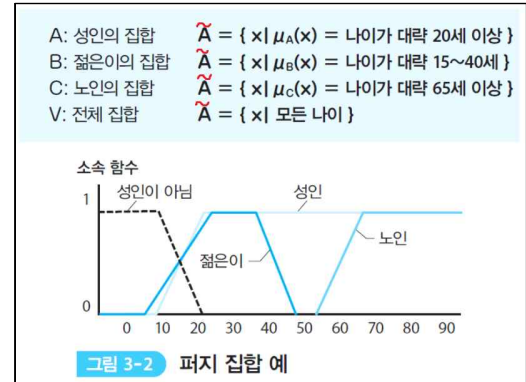
소속 함수(Membership Function)

- 역할 : 경계가 모호한 상태의 집합의 소속도를 나타냄
- 범위 : 집합의 안쪽 1, 바깥쪽 0

$$A = \{x | \mu_A(x) = y, y > 0\}$$

$$\mu_A(x) \rightarrow [0, 1]^{*2} \quad 0 < \mu_A(x) \leq 1 (x \in A) \quad \text{or} \quad \mu_A(x) = 0 (x \notin A)$$

퍼지 집합 : 예



- 퍼지 집합이 아니면 '대략 20세' 이런 거 안됨
- > 생일 하루 남았는데 성인이야? 아니야?
- 성인을 10살부터 보고 20살 넘어서 다 성인
- 젊은이를 7살정도 보고 40 후반까지 젊다
- => 양 끝은 소속이 되기에 소속 함수 값이 낮다.

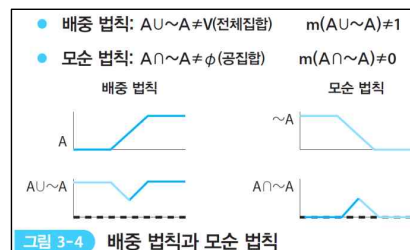
Q.성인이 아닌 집합D가 있다 생각하면 어떻게 볼까? (여집합)

- > D: 성인이 아닌 집합 $D = \{x | \mu_D(x) \neq \text{나이가 대략 20세 이상}\}$
- > A, D를 합하면 V가 되므로, $m(A \cup D) = 1$ 이 될 것 같지만 아님
- > 정의 : $\mu_D(x) = 1 - \mu_A(x)$
- > 이거 아님 : 1- A여집합 != A집합

퍼지 집합 : 연산

- 되는 연산
- ✓ 교환 법칙: $A \cup B = B \cup A, A \cap B = B \cap A$
- ✓ 결합 법칙: $A \cup (B \cap C) = (A \cup B) \cap C, A \cap (B \cup C) = (A \cap B) \cup C$
- ✓ 분배 법칙: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C), A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
- ✓ 이중 부정: $A = \sim \sim A$
- ✓ 드모르간 법칙: $\sim(A \cup B) = \sim A \cap \sim B, \sim(A \cap B) = \sim A \cup \sim B$

- 안되는 연산 : 배중법칙, 모순법칙



- > 배중 : A집합 \cup A여집합 != 전체집합
- > 모순 : A집합 \cap A여집합 != 공집합

* 쌍대성 원리 : $\cap \Leftrightarrow \cup$, 공집합 \Leftrightarrow 전체집합을 둘 다 바꾸면 식성립

퍼지 추론

- 일반 추론

- > 삼단 논법처럼 진위 판정을 위한 이진논리 추론은 현실 적용 어렵
- > 현실 문제 : ~에 가깝다 ~처럼 보인다 같은 범위

=> **맘다니(Mamdani)추론** : 위 범위를 고려한 추론이 가능한 기술

- * 정말 중요한 퍼지 추론... 해봅시다~!
- * 퍼지 추론까지 시험범위입니다~
- * 많은 변형 추론 기법이 있지만 저걸 배우는 걸로

8주차 강의 - (04.19-화)

퍼지 추론의 개념

- $P \rightarrow Q$: P이면, Q이다. : 긍정식(모던스 포넨스)

P	\rightarrow	Q
T	T	T
T	F	F
F	T	T
F	F	F

- 가정이 틀리면 다 틀려...? : 부정식(모던스 포넨스)

- > P : 날개가 있으면 난다.
- > 날개가 있을 수도 없을 수도 있으니까. P는 T거나 F다.
- > P : 관찰된 사실 / Q : 검증된 지식
- > 비가오면, 우산을 쓴다.
- > P이면 Q인 것이 밝혀졌을 때, P가 참이면, Q 참일 수 밖에 없다.

- $\neg Q \rightarrow P$: Q의 역이 참이라면, P는 참이다. : 모던스 토렌스

- > 우산을 안썼다. 비가 안온다.

퍼지 추론의 개념

- 이진 추론의 긍정식을 다음과 같이 확장한 형태

$((p \rightarrow q) \& p') \rightarrow q'$ (p에 가까우면 q에 가깝다)

- 퍼지 규칙 : $p \rightarrow q$

- > 조금더 애매한 P, ex. 비가 좀 오면

- p, q : 퍼지 집합, 애매한 집합
- p' : 현실 관측값 or 퍼지 집합값
- q' : 결론, 수치값으로 도출

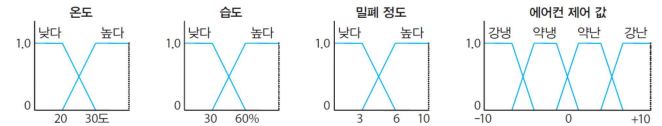
퍼지 추론의 예

에어컨 제어 퍼지 규칙

- R1: IF(온도와 습도가 모두 높고, 방의 밀폐 정도가 높다) THEN 약냉
- R2: IF(온도와 습도가 모두 높고, 방의 밀폐 정도가 낮다) THEN 강냉
- R3: IF(온도가 높고 습도가 낮을 때는 밀폐 정도에 관계없다) THEN 약냉
- R4: IF(온도가 낮고 습도가 높을 때는 밀폐 정도에 관계없다) THEN 약난
- R5: IF(온도와 습도가 모두 낮고, 방의 밀폐 정도가 높다) THEN 약난
- R6: IF(온도와 습도가 모두 낮고, 방의 밀폐 정도가 낮다) THEN 강난

- '몇 도' 같이 정확한 규칙을 두지 않는다.

소속 함수

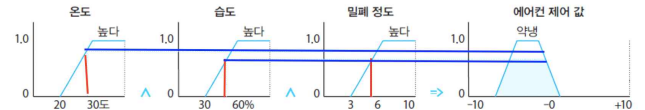


- x축 : 온도, 습도, 밀폐도, 에어컨 제어 값
- y축 : 소속함수. 수치는 제작자의 마음

가정 상황

온도 28도, 습도 50%, 밀폐 정도 5의 경우

R1: 온도와 습도가 모두 높고, 방의 밀폐 정도가 높을 때에는 '약냉'



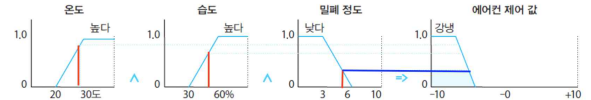
- 각각 한 온도 0.8, 습도 0.7, 밀폐도 0.7 정도 된다.

=> 확신도가 낮아진다.

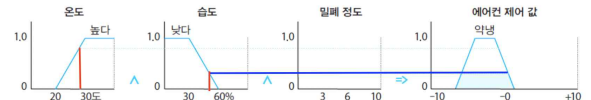
- 퍼지의 \wedge : 참^참 == 참X, 전체 요소의 minimum 값을 취한다.
- 퍼지의 and로 Q를 잘라버린다. : 그림의 파란칠 (0.7만 약냉한다.)

* \wedge : and 조건

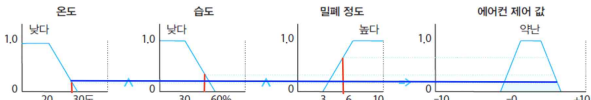
R2: 온도와 습도가 모두 높고, 방의 밀폐 정도가 낮을 때에는 '강냉'



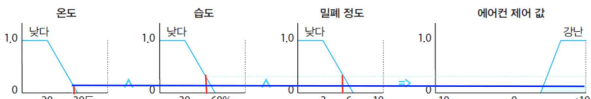
R3: 온도가 높고 습도가 낮을 때는 밀폐 정도에 관계없이 '약냉'



R5: 온도와 습도가 모두 낮고, 방의 밀폐 정도가 높을 때에는 '약난'



R6: 온도와 습도가 모두 낮고, 방의 밀폐 정도가 낮을 때에는 '강난'



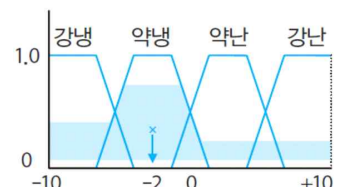
- 1) 각 규칙의 최소 척도를 구한다.
- 2) 수평절단을 수행해 소속함수들의 합집합을 구한다.

> 우측 그림 : 합집합

> 약냉이 가장 큼을 알 수 있다.

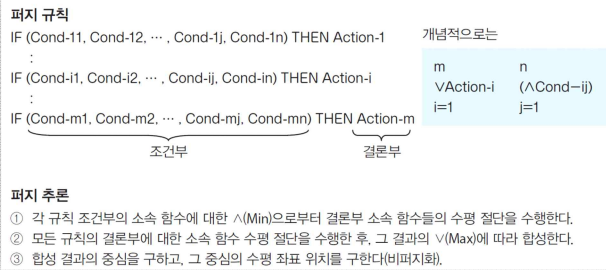
> 약냉은 -3도지만 -2로 냈다

> 무게 중심법을 이용



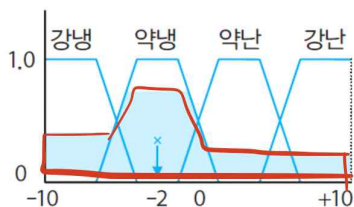
3) 최종적인 결론, 비퍼지화에 따라 '에어컨 제어값을 2도 낮춤' 도출

퍼지 추론 순서



- 1) 퍼지 규칙 정의 : IF (조건부) THEN (결론부)
- 2) 규칙에 나타나는 개념(매대한 용어)의 소속함수 정의
- 3) 규칙의 조건부에 대한 각 개념의 관측값에 대해,
각 개념의 교집합을 구함(개념 척도의 최소값)
- 4) 결론부 개념의 소속함수를 조건부 척도의 최소값으로 수평절단
- 5) 2)의 각 규칙에 대해 3)4)를 수행해 소속함수 결과의 합집합 도출
 > 결과를 나타내는 새로운 소속함수가 됨
- 6) 5)의 함수의 무계중심으로 비퍼지화(defuzzification) 수행

무게 중심법



- 퍼지에서 결론적으로 온도를 도출한 방법 중 하나
- 도출법 : 위 영역에서 무게중심을 잡을 수 있는 지점
- CoG : Center of Gravity
- 수식 : 적분을 이용한다. 구분구적법!?!? 셋상에 여기서 써?

프로그램 돌려보기 : Ex3_퍼지 추론

Ex.3 퍼지 추론에 의한 에어컨 제어

Copyright © 2015 ASAI Noboru

소속 함수

온도

온도 °C

낮다

높다

습도

습도 %

낮다

높다

필패 정도

필패정도

낮다

높다

에어컨 제어

제어값

강판

냉각수

핵심

장난