Ch02_ 규칙기반 전문가 시스템

이 장에서 다룰 내용

- 01_지식이란 무엇인가?
- 02_지식 표현 기법으로써의 규칙
- 03_전문가 시스템 개발팀의 주요 구성원
- 04_규칙기반 전문가 시스템의 구조
- 05_전문가 시스템의 기본적인 특성
- 06_순방향 연결과 역방향 연결 추론 기법
- 07_MEDIA ADVISOR: 논증 규칙기반 전문가 시스템
- 08_충돌 해법
- 09_규칙기반 전문가 시스템의 장점과 단점
- 10_요약

01_지식이란 무엇인가?

❖ 지식이란?

- 지식이란 무엇인가
  - 지식은 어떤 주제나 분야에 대해 이론적으로 혹은 실제로 이해하는 것
  - 현재 알려진 사실들의 모음이기도 하며, 권력이 되기도 함.
  - 지식을 소유한 사람을 전문가(expert)라고 불리워짐.
  - 전문가는 조직에서 가장 영향력 있고 핵심적인 사람
  - 성공한 회사는 적어도 몇 명의 일급 전문가를 고용하며, 그들 없이 사업을 이끌기 힘듦.

- 일반적으로 어떤 사람을 전문가라고 하는가
  - 전문가는 특정 분야에 해박한 지식(사실과 규칙 둘 다에 관한)과 풍부한 경험을 쌓은 사람이지만, 분야의 범위는 제한적
  - e.g., 전기 기계 전문가는 변압기에 관해서는 보통 정도만 알고, 생명 보험 마케팅 전문가는 부동산 보험 원칙을 제한 적으로 앎.
  - 일반적으로 전문가는 다른 사람이 할 수 없는 일을 해낼 수 있는 숙련된 사람

- 전문가는 어떻게 사고하는가?
  - 인간의 정신 작용은 내면적이고, 알고리즘으로 표현하기에는 너무 복잡함.
  - 대부분의 전문가는 자신의 지식을 문제 풀이에 관한 규칙 형식으로 표현하는 데 능숙함.

01_지식이란 무엇인가?

❖ 지식이란?

- 전문가는 어떻게 사고하는가?
  - 당신은 외국인에게 신호등이 녹색일 때는 안전하게 길을 건널 수 있고
  - 신호등이 빨간색일 때는 길을 건너지 말아야 한다고 설명 할 때
  - 당신의 지식은 다음의 간단한 문장으로 정식화할 수 있다.

IF '신호등'이 녹색이다
THEN 길을 건넌다

IF '신호등'이 빨간색이다
THEN 멈춰 기다린다

  - 이 IF-THEN 형식으로 표현한 문장을 생성 규칙 또는 줄여서 규칙이라고 함.
  - AI에서 '규칙'은 지식을 표현하는 가장 일반적인 방법으로, IF 부분에 주어진 정보나 사실을 THEN 부분에 기술 한 어떤 행동과 연결하는 IF-THEN 구조로 정의
  - 규칙은 문제를 어떻게 풀 것인지에 대한 설명을 제공.
  - 규칙은 비교적 만들고 이해하기 쉬움.

02_지식 표현 기법으로써의 규칙

❖ 규칙의 문법

- 규칙은 두 부분으로 구성됨.
  - 전건(antecedent), 전제(promise) 또는 조건(condition)이라고 하는 IF 부분
  - 후건(consequent), 결론(conclusion) 또는 행동(action)이라고 하는 THEN 부분

- 규칙의 기본 문법

IF 〈전건〉
THEN 〈후건〉

- 하나의 규칙에는 AND나 OR 혹은 둘을 조합한 여러 전건이 있을 수 있음.
- 같은 규칙 안에서는 논리곱과 논리합을 섞어 쓰지 않는 편이 좋음.
- 규칙의 후건에도 여러 절이 있을 수 있음.

IF 〈전건 1〉
OR 〈전건 2〉
.
.
.
OR 〈전건 n〉
THEN 〈후건〉

IF 〈전건〉
THEN 〈후건 1〉
〈후건 2〉
.
.
.
〈후건 m〉

02_지식 표현 기법으로써의 규칙

❖ 객체,값 그리고 연산자를 이용한 문법

- 규칙의 전건에서는 객체(object), 혹은 언어 객체(linguistic object)에 값(value)이 들어감.
- 연산자(operator)는 객체를 판별하고 값을 대입한다. ' ~이다', '~ 아니다'와 같은 연산자는 심벌 값을 언어객체에 대입하려고 사용.
- 전문가 시스템은 수학 연산자를 사용해서 객체를 숫자로 정의하고 수치를 대입할 수도 있음.

IF	'고객의 나이' < 18
AND	'현금인출액' > 1000
THEN	'부모의 서명' 이 필요합니다

- 후건은 전건처럼 연산자로 연결된 객체와 값을 결합함.
- 연산자는 값을 언어 객체에 대입
  - IF '과세 소득' > 16283
  - THEN '의료보험 징수액' = '과세 소득' * 1.5 / 100

02_지식 표현 기법으로써의 규칙

❖ 규칙은 관계, 추천, 지시, 전략, 휴리스틱을 표현할 수 있다.

- 【관계】
  - IF '연료 탱크' 가 비었다
  - THEN 차가 멈췄다

- 【추천】
  - IF 가을이다
  - AND 하늘이 흐리다
  - AND 일기예보는 보슬비다
  - THEN 조언은 '우산을 가지고 다니는 것'이다

- 【지시】
  - IF 차가 멈췄다
  - AND '연료 탱크' 가 비었다
  - THEN '차에 연료를 공급'한다

02_지식 표현 기법으로써의 규칙

❖ 규칙은 관계, 추천, 지시, 전략, 휴리스틱을 표현할 수 있다.

- 【전략】
  - IF 차가 멈췄다
  - THEN '연료 탱크를 확인'한다
1단계를 완료했다

  - IF 1단계를 완료했다
  - AND '연료 탱크'가 가득 찼다
  - THEN '배터리를 확인'한다
2단계를 완료했다

- 【휴리스틱】
  - IF 시료는 액체이다
  - AND '시료의 pH'< 6
  - AND '시료의 냄새'가 시큼하다
  - THEN '시료의 성분'은 '아세트산'이다

03_전문가 시스템 개발팀의 주요 구성원

❖ 전문가 시스템 구성

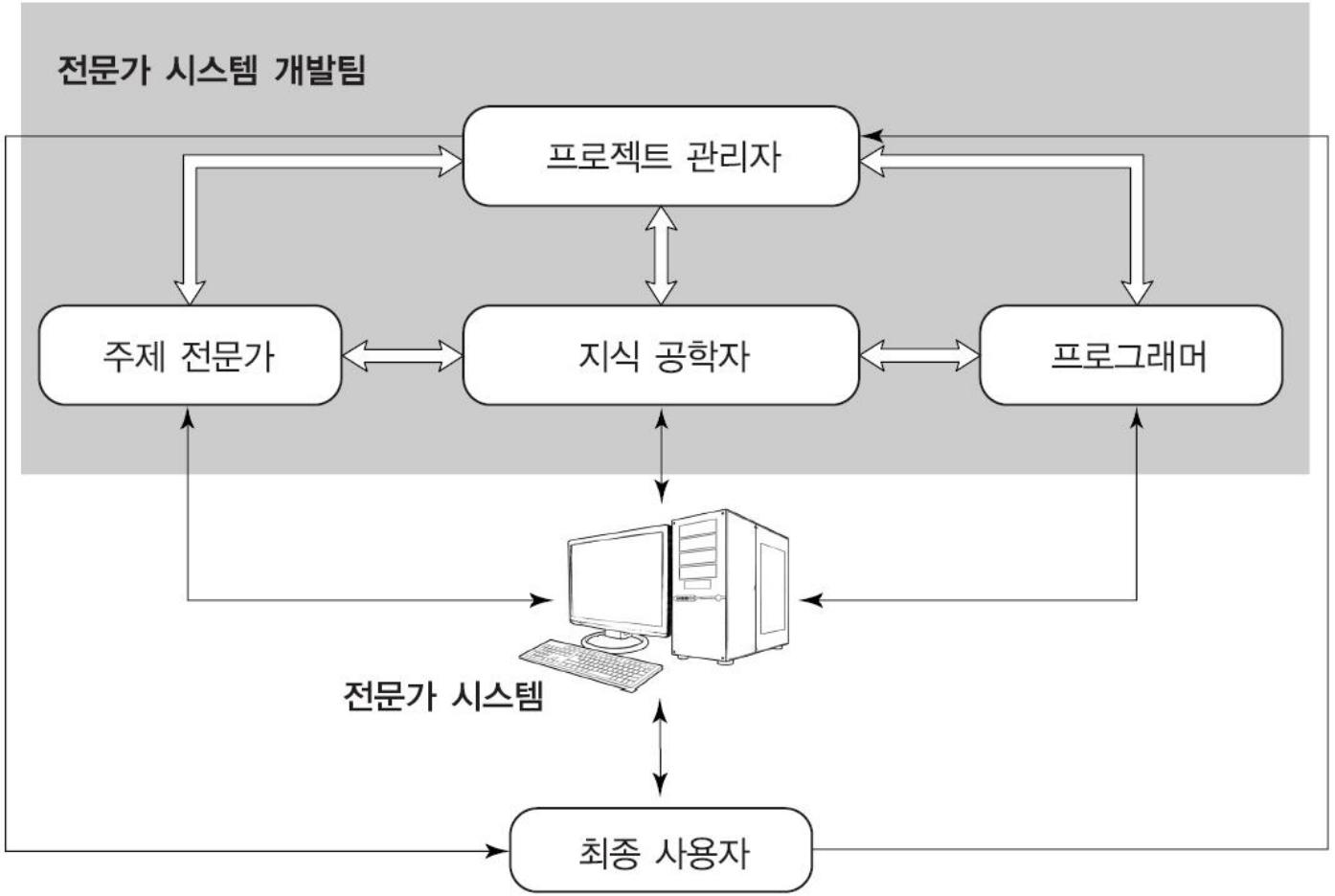
- 전문가 시스템
  - 전문가 시스템(expert systems)은 좁은 문제 영역에서 전문가 수준으로 동작할 수 있는 유능한 컴퓨터 프로그램
  - 가장 인기 있는 전문가 시스템은 규칙기반 시스템(rule-based system)

- 전문가 시스템 틀
  - 전문가 시스템 틀은 지식을 추가하지 않은 전문가 시스템
  - 모든 사용자는 지식을 규칙 형식으로 추가하고, 문제를 풀기 위해 필요한 데이터를 제공해야 함.

- 전문가 시스템을 개발하는 사람
  - 일반적으로, 전문가 시스템 개발팀은 **주제 전문가**, **지식 공학자**, **프로그래머**, **프로젝트 관리자**, **최종 사용자**까지 총 5명으로 구성.
  - 전문가 시스템의 성공 여부는 각 구성원이 얼마나 잘 협업하는지에 달림.

03_전문가 시스템 개발팀의 주요 구성원

❖ 개발팀의 기본 관계



[그림 2-1] 전문가 시스템 개발팀의 주요 구성원들

03_전문가 시스템 개발팀의 주요 구성원

❖ 전문가시스템 개발의 주요 구성원

- 도메인 엑스퍼트 (주제 전문가;domain expert)
  - 특정 분야나 주제(domain)에 대한 지식이 풍부하고, 관련 문제를 푸는 데 능숙한 사람으로 주제에 대해 최고의 전문 지식을 갖추고 있어야 함.
  - 주제 전문가의 전문 지식은 전문가 시스템에 저장.
  - 전문가는 자신의 지식을 전달할 수 있어야 하고, 전문가 시스템 개발에 기꺼이 참여해야 하며, 프로젝트에 많은 시간을 투자해야 함.
  - 주제 전문가는 전문가 시스템 개발팀에서 가장 중요한 사람임.

- 지식 공학자(knowledge engineer)
  - 전문가 시스템을 설계하고, 만들고, 테스트할 수 있는 사람으로 전문가 시스템을 만들기 위해 어떤 일을 해야 하는지 결정.
  - 지식 공학자는 특정 문제를 푸는 방법을 알아내기 위해 주제 전문가와 상담하는데 상담을 통해 전문가가 사실과 규칙을 다루기 위해 어떤 추론 방법을 사용했는지 알아내고, 전문가 시스템에서 이를 어떻게 표현할지 결정.
  - 그 후에 지식 공학자는 기존의 개발 소프트웨어나 전문가 시스템 틀을 선택하고 지식을 표현할 프로그래밍 언어를 살펴봄.(때로는 직접 표현)
  - 마지막으로, 전문가 시스템을 실제 작업 영역으로 옮겨 테스트하고 수정하여 통합. 따라서 지식 공학자는 전문가 시스템의 초기 설계 단계부터 최종 단계, 그리고 프로젝트가 완료 후에도 시스템을 유지하는 일에 참여함.

03_전문가 시스템 개발팀의 주요 구성원

❖ 전문가시스템 개발의 주요 구성원

- **프로그래머**(programmer)
  - 프로그래밍을 책임지며 지식을 컴퓨터가 이해할 수 있는 용어로 기술하는 사람.
  - 프로그래머는 LISP, Prolog, OPS5 같은 AI 언어를 프로그래밍할 줄 알아야 하고, 다른 형태의 여러 전문가 시스템 틀을 응용한 경험이 있어야 하고 C, Pascal, FORTRAN, Basic 같은 전통적인 프로그래밍 언어도 알고 있어함.
  - 전문가 시스템 틀을 사용할 때는 지식 공학자가 전문가 시스템에 지식을 쉽게 코딩할 수 있어서 프로그래머가 필요하지 않을 수 있음.
  - 그러나 그렇지 않을 때는 프로그래머가 지식, 데이터 표현 구조(기반지식과 데이터베이스), 제어 구조(추론 엔진), 대화 구조(사용자 인터페이스) 등을 개발해야 하고, 전문가 시스템을 테스트할 때에도 참여할 수 있다.

- **프로젝트 관리자**(project manager)
  - 전문가 시스템 개발팀의 리더로서, 프로젝트를 제대로 진행할 수 있도록 관리
  - 실행 가능한 모든 일과 목표를 충족시키고, 다른 역할의 구성원과 상호 작용.

03_전문가 시스템 개발팀의 주요 구성원

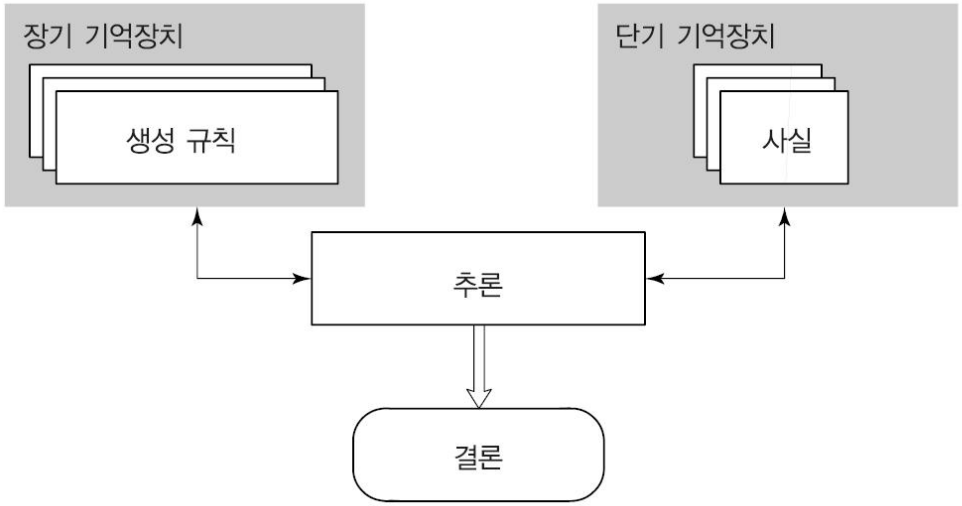
❖ 전문가시스템 개발의 주요 구성원

- 최종 사용자(end-user 또는 사용자(user))
  - 개발한 전문가 시스템을 사용하는 사람을 말함.
  - 사용자는 화성 토양의 분자 구조를 연구하는 분석 화학자(Feigenbaum 외, 1971), 전염성 순환계질환을 진단해야 하는 경험이 부족한 의사(Shortliffe, 1976), 새로운 광물 매장지를 발견하고자 하는 탐사 지리학자(Duda 외,1979), 또는 응급 시에 조언을 받고 싶은 발전 시스템 기사일 수도 있다(Negnevitsky,1976).

04_규칙기반 전문가 시스템의 구조

❖ 규칙기반 전문가 시스템의 탄생

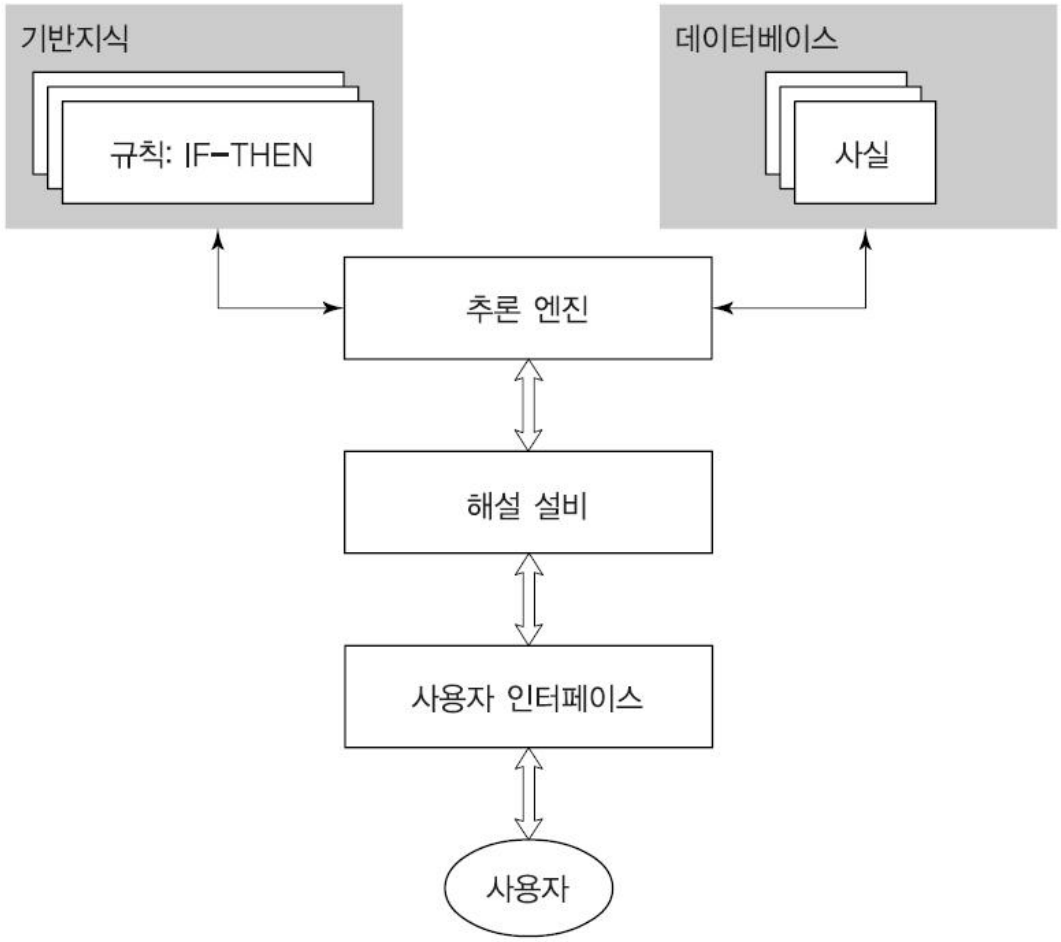
- 1970년대 초반, 카네기멜론 대학교의 뉴웰과 사이먼은 현대의 규칙기반 전문가 시스템의 기초가 된 생성 시스템 모델을 제안(Newell과 Simon, 1972).
- 생성 모델은 인간이 자신의 지식을 적용해서 제시된 문제를 관련 정보로 해결한다는 아이디어에 기반
- 생성 규칙(production rules)은 장기 기억장치(long-term memory)에 저장되고 특정 정보나 사실은 단기 기억장치(short-term memory)에 저장됨.
- 생성시스템 모델.



(a) 생성 시스템 모델

04_규칙기반 전문가 시스템의 구조

❖ 규칙기반 전문가 시스템의 기본 구조



(b) 규칙기반 전문가 시스템의 기본 구조

[그림 2-2] 생성 시스템과 규칙기반 전문가 시스템의 기본 구조

04_규칙기반 전문가 시스템의 구조

❖ 규칙기반 전문가 시스템의 필수요소

- 기반지식(날리지베이스 ; 지식 베이스; knowledge base)
  - 기반지식(knowledge base)는 문제 해결에 필요한 특정 분야에 관한 지식을 포함.
  - 규칙기반 전문가 시스템에서는 지식을 규칙 집합으로 표현
  - 각각의 규칙은 관계, 추천, 지시, 전략, 휴리스틱을 명시하고 IF(조건) THEN(취해야 할 행동)의 구조를 띔
  - 규칙의 조건 부분을 만족 하면 규칙이 점화되었다고 하며, 취해야 할 행동 부분을 실행

- 데이터베이스(database)
  - 데이터베이스(database)는 기반지식에 저장된 규칙의 IF(조건)와 비교할 때 사용하는 사실(fact)들의 집합을 저장하고 있음.

- 추론 엔진(inference engine)
  - 추론 엔진(inference engine)은 전문가 시스템이 해를 구할 수 있도록 추론 역할을 담당하며 기반지식에 주어진 규칙들을 데이터베이스에 있는 사실과 연결

04_규칙기반 전문가 시스템의 구조

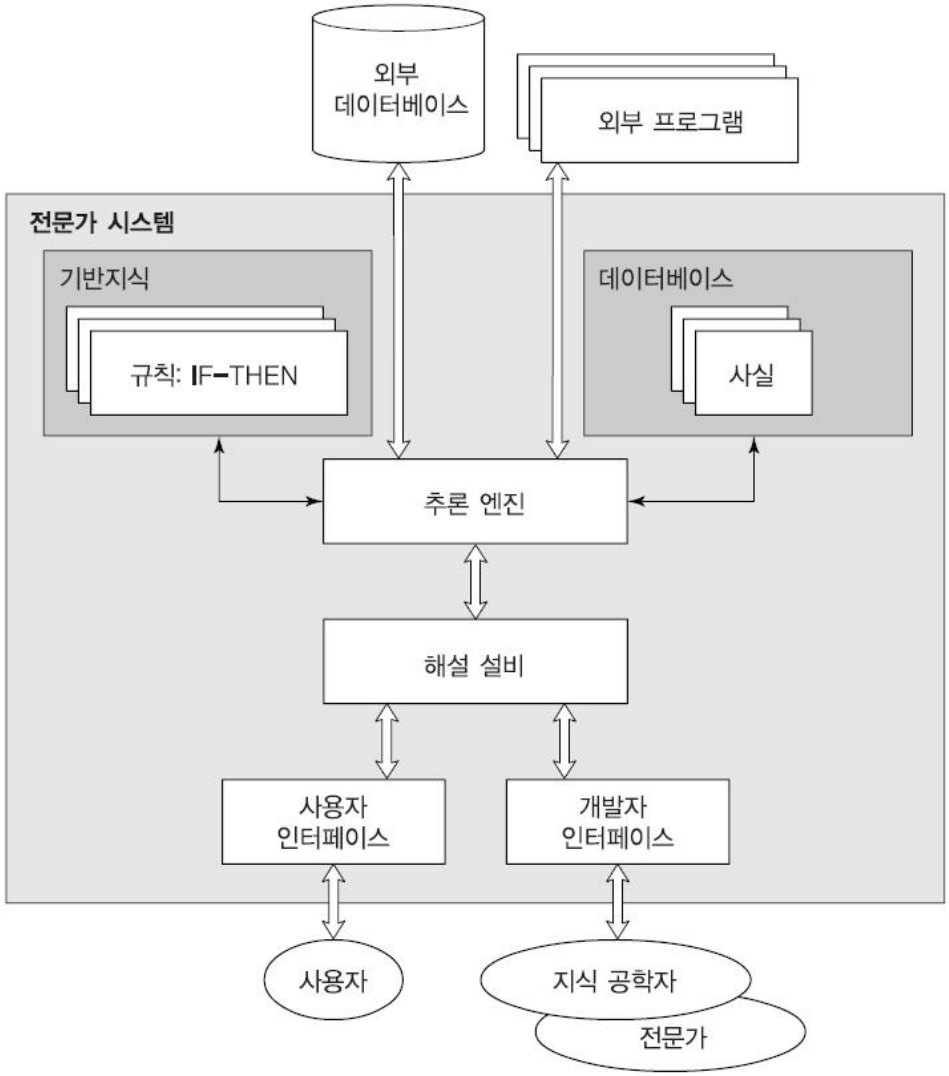
❖ 규칙기반 전문가 시스템의 필수요소

- **해설 설비**(explanation facilities)
  - **해설 설비**(explanation facilities) **는 사용자에게 전문가 시스템이 어떻게 특정 결론에 이르렀는지, 왜 특정 사실이 필요한지 설명함.**
  - **따라서 전문가 시스템은 자신의 추론 과정을 설명하고, 조언, 분석 또는 결론의 타당성을 밝힐 수 있어야 함.**

- **사용자 인터페이스**(user interface)
  - **문제의 답을 찾고 싶어 하는 사용자와 전문가 시스템 간의 통신 수단**
  - **통신은 가능한 한 의미 있고 사용하기 편리해야 함.**

04_규칙기반 전문가 시스템의 구조

❖ 규칙기반 전문가 시스템의 완전한 구조



04_규칙기반 전문가 시스템의 구조

❖ 부가적 구성요소들

- **외부 인터페이스**(external interface)
  - C, Pascal, FORTRAN, Basic과 같은 전통적인 프로그래밍 언어로 만든 외부 데이터 파일과 프로그램을 전문가 시스템이 사용할 수 있게 함.
- **개발자 인터페이스**(developer interface)
  - 주로 기반지식 편집기, 디버깅 보조도구, 입출력 설비들을 포함
- **텍스트 편집기**(text editor)
  - 규칙을 입력, 수정하고 규칙의 형식과 철자를 검사할 수 있는 간단한 텍스트 편집기 제공
- **부기 설비**(book-keeping facilities)
  - 지식 공학자나 전문가가 만든 변화를 감지하기 위해 포함됨.
  - 규칙이 바뀌면 후에 참조하기 위해 편집기는 자동으로 수정한 날짜와 사람의 이름을 저장
  - 다수의 지식 공학자와 전문가가 기반지식에 접근해 규칙을 수정할 때 중요한 기능

04_규칙기반 전문가 시스템의 구조

❖ 부가적 구성요소들

- **디버깅 보조도구**(debugging aid)
  - **주로 추적 설비**(tracing facilities)**와 중단 패키지**(breakpackages)**로 구성됨.**
  - **추적 설비는 프로그램 수행 중에 점화된 모든 규칙 목록을 제공**
  - **중단 패키지는 지식 공학자나 전문가가 현재 데이터베이스에 저장된 값을 검사하고 싶을 때 전문가시스템을 멈출 수 있도록 해 줌.**

- **실행시간 지식 습득**(run-time knowledge acquisition)**과**
  - **이를 통해 전문가 시스템은 실행 도중 데이터베이스에 필요한 정보가 없으면 해당 정보를 요청한다. 지식 공학자나 전문가가 요청한 정보를 입력하면 프로그램은 다시 실행됨.**

05_전문가 시스템의 기본적인 특성

❖ 전문가 시스템의 기본적인 특성

- 전문가 시스템은 좁고 전문화된 분야에서 전문가 수준으로 동작하도록 설계됨. 따라서 전문가시스템의 가장 중요한 특성은 고품질의 성능.
- 시스템이 문제를 얼마나 빨리 풀 수 있는지를 떠나 그 결과가 잘못되었다면 사용자는 만족하지 않음.
- 그렇다고 해도 문제해결 속도는 매우 중요함. 환자가 죽거나 핵발전소가 폭발한 경우와 같이 매우 급한 상황에서 해(solution)를 구하는데 많은 시간이 든다면, 판단이나 진단이 정확하다 해도 무용지물이 되기 때문.
- 전문가는 문제를 정확하게 이해하고 실제 경험을 사용하여 일반인보다 해를 좀 더 빨리 구함.
- 전문가는 문제를 해결하기 위해 어림짐작이나 휴리스틱을 사용.
- 인간을 모방하는 전문가 시스템은 추론을 잘 이끌어내도록 휴리스틱을 적용하여 해의 탐색 영역을 줄임.

05_전문가 시스템의 기본적인 특성

❖ 해설 능력(explanation capability)

- 전문가 시스템의 독특한 특징으로 이를 통해 전문가 시스템은 자신의 추론을 재검토하고, 결론을 설명함.
- 실제 전문가 시스템에서 해설이란 문제를 푸는 동안 점화된 규칙을 추적하는 것
- 기반지식에 저장된 각 규칙마다, 각각의 상위 레벨의 규칙마다 해당 분야의 적당한 기초적인 원칙을 텍스트로 표현하여 덧붙여 둘 수 있음.
- 전문가 시스템에 따라서 해설의 필요한 양이 달라짐.

❖ 심벌 추론(symbolic reasoning)

- 전문가 시스템은 문제를 풀 때 심벌 추론(symbolic reasoning)을 채택함.
- 개념, 규칙 같은 다른 종류의 지식을 표현하는 데 심벌을 사용
- 수치 데이터를 처리하는 기존의 프로그램과 달리 전문가 시스템은 지식을 처리하며 질적인 데이터를 쉽게 다룰 수 있음.

05_전문가 시스템의 기본적인 특성

❖ 전통적인 프로그램 VS 전문가 시스템

- 전통적인 프로그램은 알고리즘(algorithm), 즉 잘 정의된 단계적 연산을 이용하여 데이터를 처리함.
- 알고리즘은 항상 같은 순서로 같은 연산을 수행하며, 늘 정확한 해를 제공하므로 전통적인 프로그램은 실수를 하지 않음. (프로그래머는 실수를 한다).
- 전문가 시스템은 미리 기술된 단계의 순서를 따르지 않기 때문에 정확하지 않은 추론을 허용 하고 불완전하고 불확실하며 모호한 데이터를 다룰 수 있음.

❖ 전문가 시스템은 실수할 수 있는가

- 훌륭한 전문가라 할지라도 인간인 이상, 실수를 할 수도 있다!
- 전문가 수준으로 동작하는 전문가 시스템의 실수도 허용해야 함.
- 전문가들의 판단이 가끔 틀릴 수있다는 것을 알면서도 여전히 전문가를 신뢰하는 것과 같이, 대부분의 경우에는 전문가 시스템이 제공한 해를 신뢰할 수 있지만, 실수를 할 가능성도 있다는 점을 인지해야 함.

05_전문가 시스템의 기본적인 특성

❖ 전통적인 프로그램이 전문가 시스템에 비해 이점이 있다는 의미인가

- 전통적인 프로그램은 이론적으로 항상 같은 '정확한'해를 제공.
- 이에 비해서, 전통적인 프로그램은 데이터가 완전하고 정확할 때에만 문제를 다룰 수 있음.
- 데이터가 불완전하거나 약간의 에러를 포함하고 있으면 전통적인 프로그램은 아무런 해도 제공하지 못할 뿐만 아니라 틀린 해를 제공.
- 반면 전문가 시스템은 사용할 정보가 불완전하거나 모호한 상황에서도 동작할 수 있으며, 여전히 합리적인 결론에 도달할 수 있음.

05_전문가 시스템의 기본적인 특성

❖ 전문가 시스템의 장점

- 전문가 시스템을 전통적인 시스템과 구별할 수 있는 또 다른 중요한 특징은 지식이 처리 과정과 분리되어 있다(기반지식과 추론 엔진이 구분되어 있다)는 점.
- 전통적인 프로그램은 지식의 복합체이며, 지식을 처리하는 제어 구조. 이런 복합적인 구조는 코드가 바뀌면 지식과 처리구조에 영향을 미치므로 프로그램을 이해하고 다시 살펴보기가 어려움.
- 전문가 시스템에서는 지식이 처리 메커니즘과 명확하게 분리되어 있기 때문에 전문가 시스템을 만들고 유지하는 작업을 더 쉽게 만듦.
- 전문가 시스템 틀을 사용하면 지식 공학자나 전문가가 기반지식에 규칙을 간단하게 입력할 수 있음.
- 각각의 새로운 규칙은 새로운 지식을 추가시키며 전문가 시스템을 더 기능적으로 만들고, 그 후에 규칙을 바꾸거나 뺌으로써 시스템을 쉽게 수정할 수 있음.

05_전문가 시스템의 기본적인 특성

[표 2-1] 전문가 시스템과 전통적인 시스템, 전문가의 비교

전문가	전문가 시스템	전통적인 시스템
특정 영역의 문제를 푸는 데 어림짐작이나 휴리스틱 형태로 된 지식을 사용한다.	규칙의 형태로 표현된 지식을 처리하고 **특정 영역**의 문제를 풀기 위해 심벌 추론을 사용한다.	일반적인 수치 문제를 풀기 위해 데이터를 처리하고, 잘 정의된 연산자인 알고리즘을 사용한다.
지식은 인간의 머릿속에 수집된 형태로 존재한다.	**처리 과정과 명확히 분리된 지식**을 제공한다.	지식을 처리하기 위해 제어 구조를 지식과 분리하지는 않는다.
추론 과정을 설명할 수 있고, 세부사항을 제공할 수 있다.	문제 풀이 세션 동안 **점화된 규칙을 추적**하고 **어떻게** 특정 결론에 이르렀는지, **왜** 특정 데이터가 필요했는지 **설명**한다.	특정 결과가 어떻게 나왔는지, 왜 입력 데이터가 필요했는지 설명하지 않는다.
부정확한 추론을 사용하고, 불완전하고 불확실하며 모호한 정보를 다룰 수 있다.	**부정확한** 추론을 허용하며, 불완전하고 불확실하며 모호한 데이터를 다룰 수 있다.	데이터가 완전하고 정확할 때만 동작한다.
데이터가 불완전하거나 모호할 때 실수할 수 있다.	데이터가 불완전하거나 모호할 때 **실수할 수 있다.**	데이터가 불완전하거나 모호하면 해를 제공하지 않거나 잘못된 해를 내놓는다.
오랜 세월에 걸친 학습과 실제 훈련을 통해 문제 푸는 능력을 향상 시킨다. 이 과정은 느리고 비효율적이며 비용이 많이 든다.	기반지식에 새로운 규칙을 추가하고 오래된 규칙을 조정함으로써 문제 푸는 능력을 향상 시킨다. 새로운 지식이 생겼을 때 **수정하기 쉽다.**	프로그램 코드를 바꿈으로써 문제 푸는 능력을 향상시키는데, 이는 지식과 처리 과정 둘 다에 영향을 주므로 수정하기 어렵다.

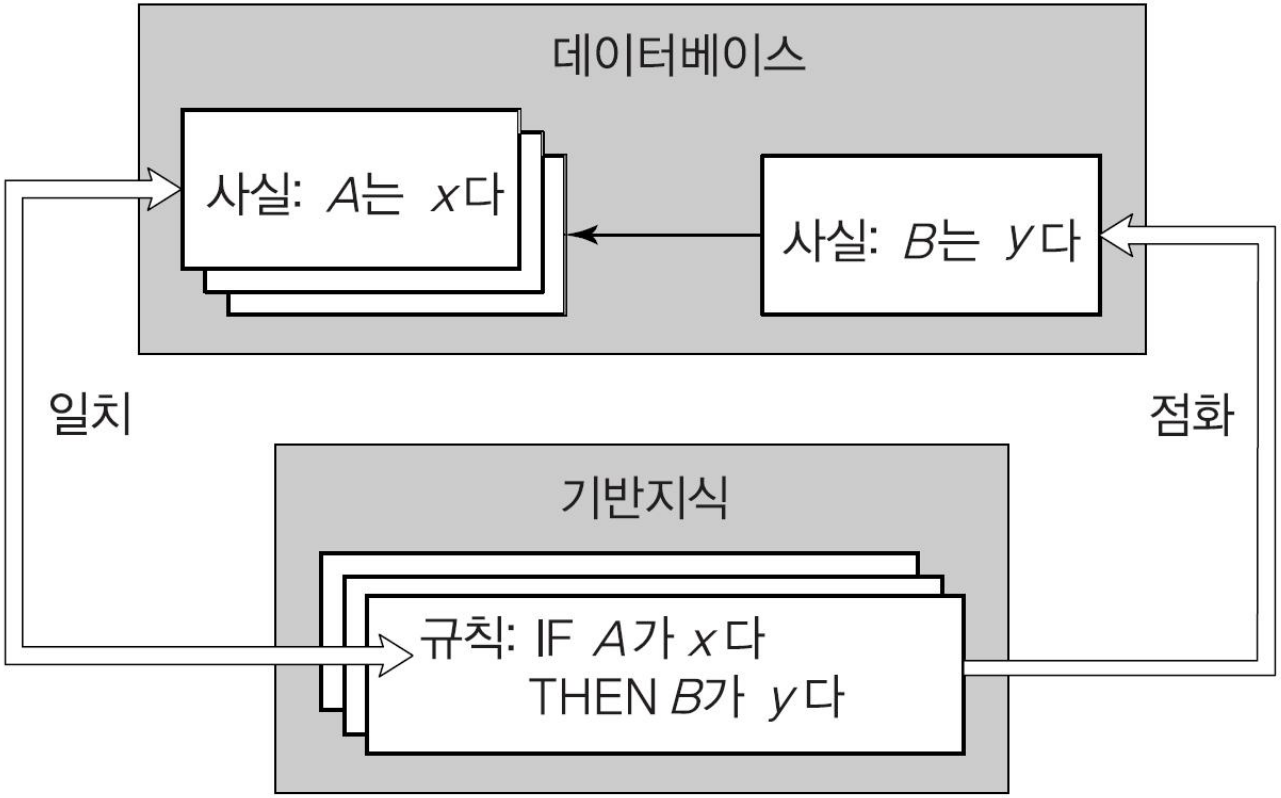
06_순방향 연결과 역방향 연결 추론 기법

❖ 규칙기반 전문가 시스템에서의 추론

- 규칙기반 전문가 시스템에서, 분야 지식은IF-THEN 생성 규칙의 집합이며 데이터는 현재 상황에 대한 지식의 집합으로 표현됨.
- 추론 엔진은 기반지식에 저장된 각각의 규칙을 데이터베이스에 있는 사실과 비교
- IF(조건) 부분이 사실과 일치하면 그 규칙은 점화되고,THEN(취해야 할 행동) 부분을 수행하고, 데이터베이스에 있는 문자와 기반지식은 상황 또는 개념을 표현하는 데 쓰임.
- 규칙의 IF 부분과 사실과의 일치는 추론 사슬(inference chains)을 생성함.
- 추론 사슬은 전문가 시스템이 결론에 이르기 위해 규칙을 어떻게 적용했는지를 나타냄.

06_순방향 연결과 역방향 연결 추론 기법

❖ 일치-점화 절차를 통한 추론 엔진 사이클

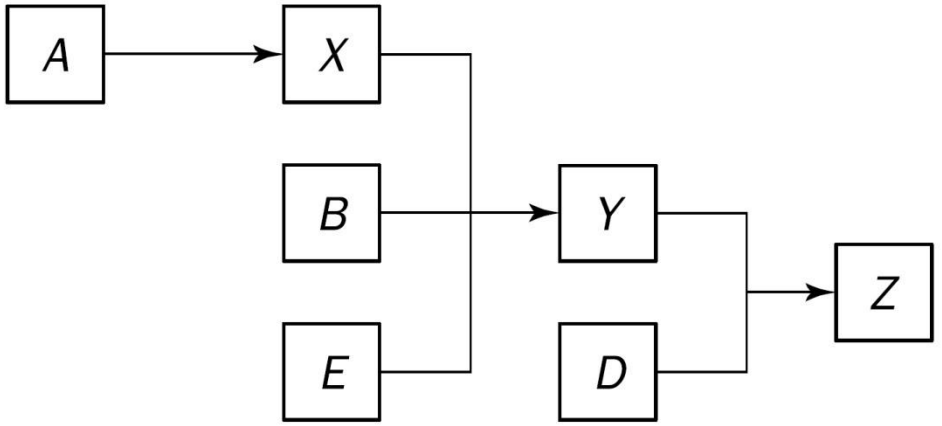


06_순방향 연결과 역방향 연결 추론 기법

- **규칙** 1: IF *Y가 참이다*
  AND *D가 참이다*
  THEN *Z가 참이다*

- **규칙** 2: IF *X가 참이다*
  AND *B가 참이다*
  AND *E가 참이다*
  THEN *Y가 참이다*

- **규칙** 3: IF *A가 참이다*
  THEN *X가 참이다*



[그림 2-5] 추론 사슬의 예

- **주어진 사실A로부터 새로운 사실X를 연역하기 위해서 규칙3이 점화**
- **그런 다음 초기에 알려진 사실B와 E 그리고 이미 알려진 사실X로부터 사실Y를 추론하기 위해 규칙 2를 수행함.**
- **마지막으로, 규칙 1은 결론 Z에 이르기 위해 초기에 알려진 사실 D와 방금 얻은 사실Y를 적용한다.**

06_순방향 연결과 역방향 연결 추론 기법

❖ 순방향 연결

- 순방향 연결은 데이터 지향(data-driven) 추론으로, 알려진 데이터에서 추론을 시작하여 순방향으로 진행해나감.
- 한 번에 가장 좋은 규칙 하나만 실행되며, 규칙이 점화되면 그 규칙은 데이터베이스에 새로운 사실을 추가함.
- 어떤 규칙이라도 한 번만 수행된다. 일치-점화 사이클은 더 이상 점화할 수 있는 규칙이 없으면 중단함.
- 다음과 같은 규칙을 생각해보자.
  - 규칙 1: $Y \& D \rightarrow Z$
  - 규칙 2: $X \& B \& E \rightarrow Y$
  - 규칙 3: $A \rightarrow X$
  - 규칙 4: $C \rightarrow L$
  - 규칙 5: $L \& M \rightarrow N$

- 여기서 화살표는 규칙의 IF와THEN 부분을 나타냄.

06_순방향 연결과 역방향 연결 추론 기법

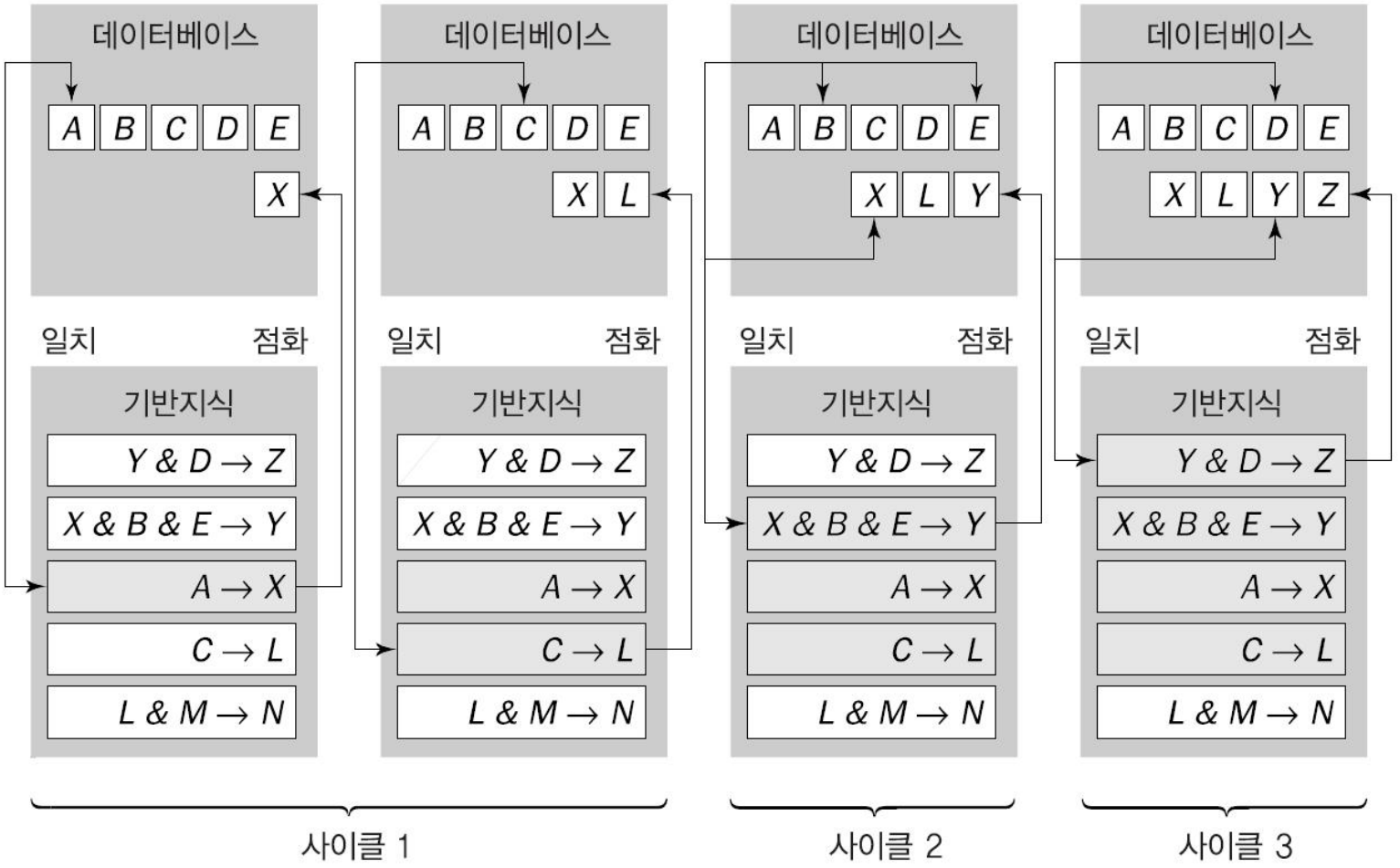
❖ 순방향 연결

- 첫 번째 싸이클
  - 규칙3: A→X와 규칙4:C→L 만이 데이터베이스에 있는 사실과 부합됨.
  - 규칙 3: A → X가 가장 위에 있으므로 먼저 점화됨.
  - 이 규칙의 IF 부분이 데이터베이스에 있는 사실 A와 일치하면, 규칙의 THEN 부분이 수행되고, 새로운 사실 X가 데이터베이스에 추가됨.
  - 그 다음 규칙4: C→L이 점화되고 사실L도 데이터베이스에 추가됨.

- 두 번째 싸이클
  - B, E, X가 이미 데이터베이스에 있으므로 규칙 2: X & B & E→Y가 점화되고, 그 결과로 사실Y가 추론되어 데이터베이스에 추가됨.
  - 이로써 규칙 1: Y & D→Z가 수행되고 사실 Z가 데이터베이스에 추가됨(사이클 3).
  - 규칙 5: L & M → N의 IF 부분은 데이터베이스의 모든 사실과 부합하지 않기 때문에 규칙 5가 점화되지 않고 일치-점화 사이클이 멈춤

06_순방향 연결과 역방향 연결 추론 기법

❖ 순방향 연결



06_순방향 연결과 역방향 연결 추론 기법

❖ 순방향 연결

- 순방향 연결은 정보를 모으고 난 후, 이를 바탕으로 추론 가능한 무언가를 추론함.
- 순방향 연결에서는 정해진 목표와 관련 없는 많은 규칙들이 수행될 수 있음.
- 앞의 예에서 사실Z를 결정하는 것이 목표라고 가정하면, 기반지식에는 다섯 개의 규칙이 있고, 그 중 네 개가 점화됨.
- 그러나 규칙 4: $C \rightarrow L$은 사실 Z와 관련이 없는데도 다른 것과 마찬가지로 점화됨.
- 기반 전문가 시스템에는 수백 개의 규칙이 있을 수 있고, 그 중 상당수는 유효한 새로운 사실을 이끌어내기 위해 점화되겠지만, 불행히도 목표와 무관할 수 있음.
- 그러므로 정해진 목표가 단지 하나의 특정한 사실을 추론하는 것이라면 순방향 연결 추론 기법은 효과적이지 못함.
- 그런 상황에는 역방향 연결이 좀 더 적당함.

06_순방향 연결과 역방향 연결 추론 기법

❖ 역방향 연결

- 역방향 연결은 목표 지향(goal-driven) 추론으로, 전문가 시스템이 목표(가정해)를 정하고, 추론엔진은 이를 증명하기 위해 증거 찾기를 시도한다. 먼저 원하는 해가 있는 규칙을 찾기 위해 기반지식을 탐색한다. 규칙은 THEN(취해야 할 행동) 부분에 목표가 반드시 있어야 한다. 목표로 한
- 규칙을 발견하고 IF(조건) 부분이 데이터베이스에 있는 데이터와 일치한다면, 규칙은 점화되고
- 목표는 증명된다. 그러나 이런 경우는 드물다. 따라서 추론 엔진은 규칙의 IF 부분을 증명하기 위
- 해 작업 중이던 규칙은 제쳐두고(스택(stack)에 놓어둔다) 새로운 목표와 하위 목표를 설정한다.
- 그리고 하위 목표를 증명할 수 있는 규칙을 찾아 기반지식을 다시 탐색한다. 추론 엔진은 현재의
- 하위 목표를 증명하기 위해 기반지식에서 더 이상 규칙을 발견할 수 없을 때까지 규칙을 스택에
- 쌓아 올리는 과정을 반복한다.

06_순방향 연결과 역방향 연결 추론 기법

❖ 역방향 연결의 추론

- 1단계
  - 추론 엔진은 사실Z를 추론함.
  - 기반지식을 탐색해서THEN 부분에 설정을 목표(여기서는 사실 Z)가 있는 규칙을 찾음.
  - 추론 엔진은 규칙 1: Y & D → Z를 찾아서 스택에 넣음.
  - 규칙1의 IF 부분은 사실Y와D를 포함하므로 이 사실을 반드시 검증해야 함.

- 2단계
  - 추론 엔진은 하위 목표와 사실 Y를 설정하고 그것을 검증함.
  - 먼저 데이터베이스를 검사하지만 사실 Y는 데이터 베이스에 없음.
  - 그러면 THEN 부분에 사실 Y가 있는 규칙을 찾기 위해 기반지식을 다시 검색
  - 추론 엔진은 규칙 2: X & B & E → Y를 발견하고 스택에 넣음.
  - 규칙 2의 IF 부분은 사실X,B, E로 이루어져 있고, 이 사실들 또한 검증해야 함.

- 3단계
  - 추론 엔진은 새로운 하위 목표로 사실X를 설정함.
  - 사실X를 찾아 데이터베이스를 검사하고, 검사가 실패할 때는X를 추론해내는 규칙을 찾음.
  - 추론 엔진은 규칙3: A → X를 찾아내고 스택에 넣는다. 이제 추론 엔진은 사실A를 검증해야 함.

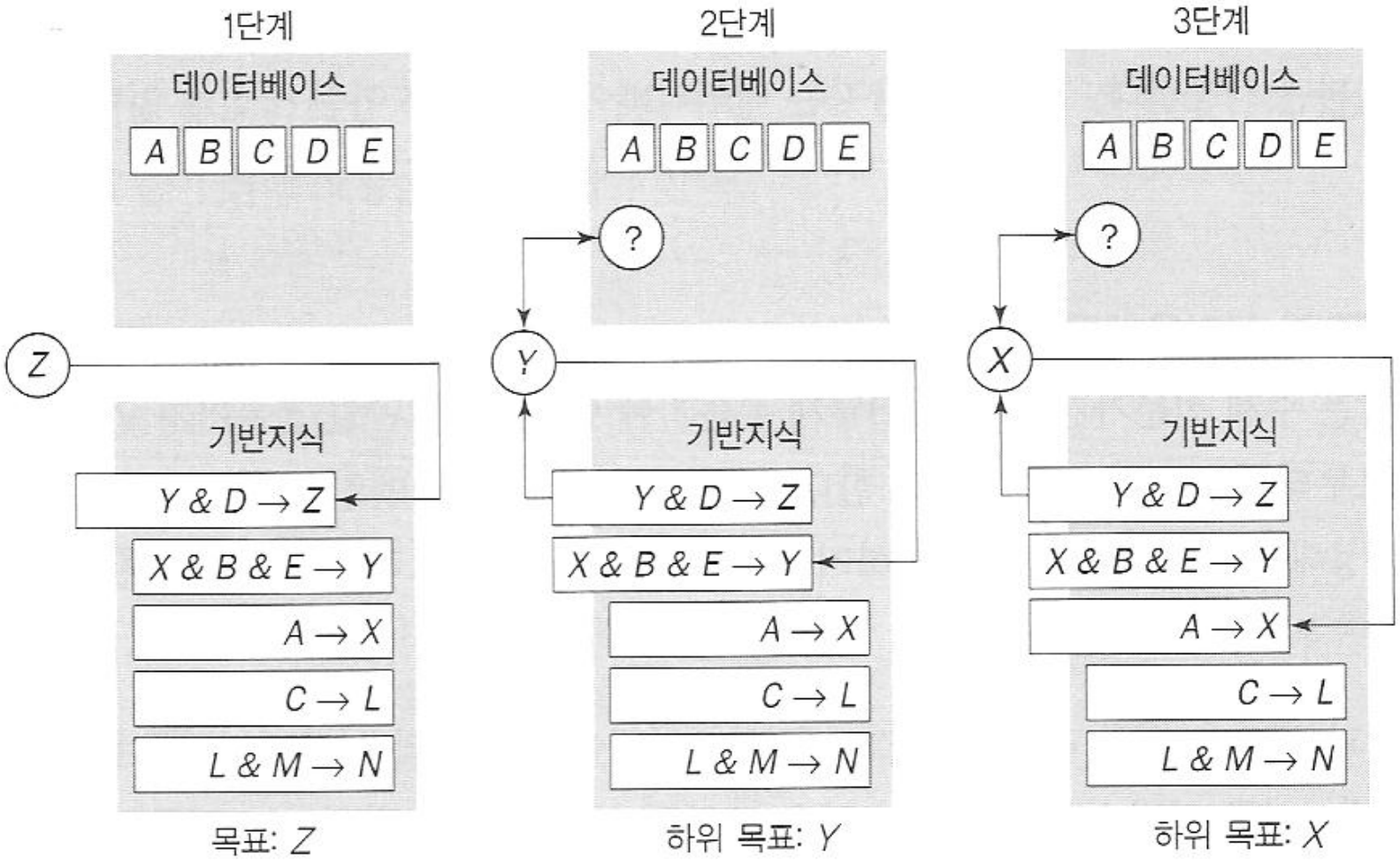
06_순방향 연결과 역방향 연결 추론 기법

❖ 역방향 연결의 추론

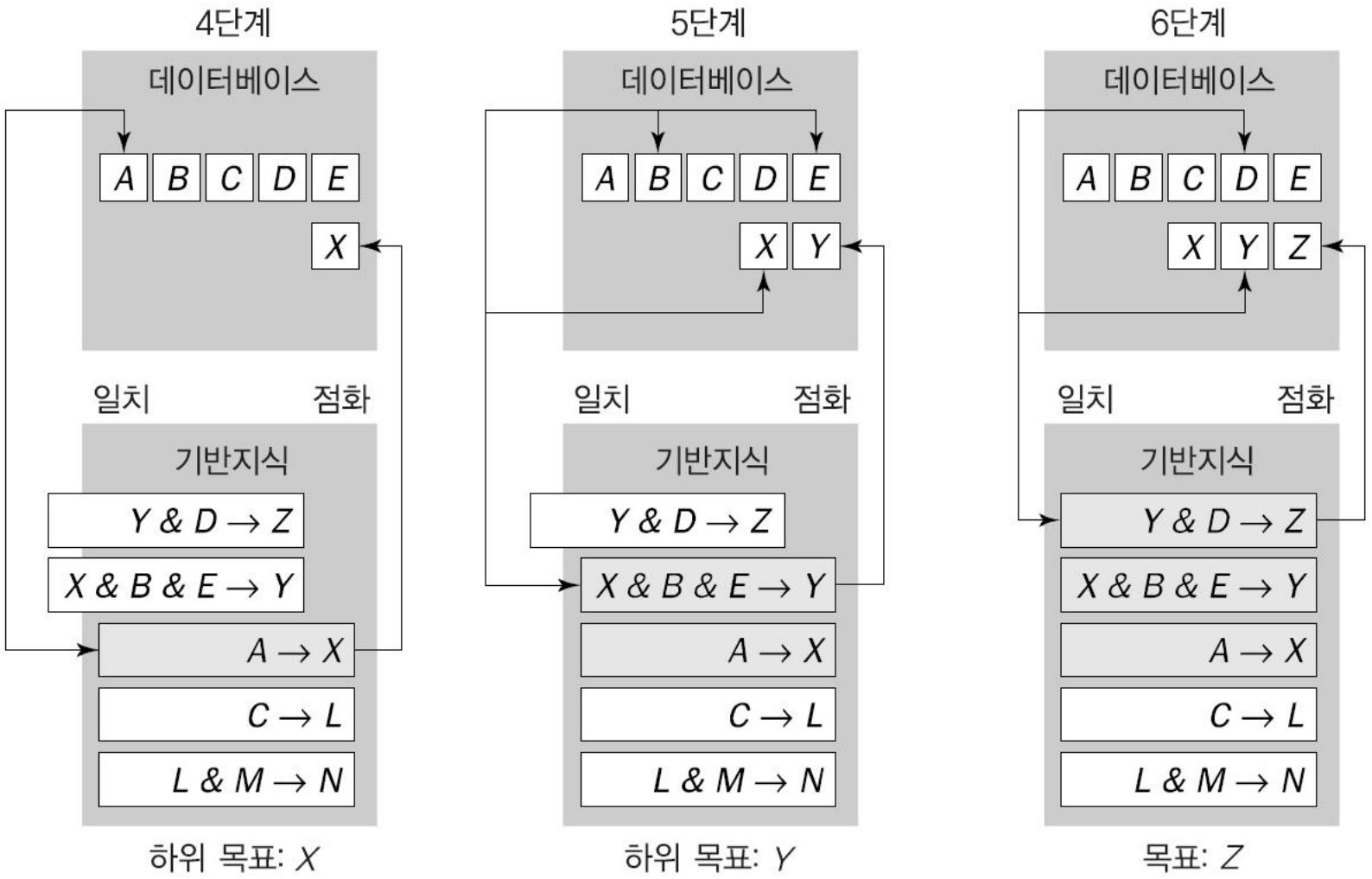
- 4단계
  - 추론 엔진은 데이터베이스에서 사실 A를 발견하고, 규칙 3: A → X가 점화되며 새로운 사실X를 추론함.
- 5단계
  - 추론 엔진은 하위 목표인 사실 Y로 돌아가 규칙 2: X & B & E → Y를 다시 수행함.
  - 사실X, B, E는 데이터베이스에 있으므로 규칙 2가 점화되며, 새로운 사실Y가 데이터베이스에 추가됨.
- 6단계
  - 시스템은 원래의 목표, 사실 Z를 이끌어내기 위해 규칙 1: Y & D → Z로 돌아감.
  - 규칙 1의 IF 부분은 데이터베이스에 있는 모든 사실과 일치하므로 규칙 1을 수행하고, 원래의 목표가 마침내 증명됨.
- 순방향 연결을 사용했을 때는 네 개의 규칙이 점화되지만 역방향 연결을 적용했을 때는 단지 세 개의 규칙만 점화됨.
- 이 간단한 예는 특정한 사실 (여기서는 사실Z) 하나를 추론해야 할 때 역방향 연결 추론 기법이 좀 더 효과적이라는 것을 보여줌.

06_순방향 연결과 역방향 연결 추론 기법

❖ 역방향 연결 - 1~3 단계



❖ 역방향 연결 (계속) - 4~6 단계



06_순방향 연결과 역방향 연결 추론 기법

- 순방향과 역방향 연결 중 무엇을 선택해야 할까
  - 도메인 엑스퍼트가 문제를 어떻게 푸는가에 달려 있음.
  - 만일 전문가가 먼저 정보를 수집한 후 그로부터 무엇인가를 추론하려 한다면, 순방향 연결 추론 엔진을 선택하면 됨.
  - 그러나 전문가가 가정 해에서 시작해 이를 증명하기 위해 어떤 사실을 찾으려 한다면 역방향 연결추론 엔진을 선택하면 됨.
  - 순방향 연결은 분석과 해석을 위한 전문가 시스템을 설계하기에 자연스러운 방법
  - E.g.,질량 스펙트럼 데이터를 기반으로 알려지지 않은 토양의 분자 구조를 결정하는 전문가 시스템인 DENDRAL(Feigenbaum 외, 1971)
  - 역방향 연결 전문가 시스템 대부분은 진단 목적으로 사용됨.
  - E.g., 전염성 혈액 질환을 진단하는 의학 전문가 시스템인 MYCIN(Shortliffe,1976)

- 순방향과 역방향 연결을 결합할 수 있는가
  - 많은 전문가 시스템 틀이 순방향과 역방향 연결 추론 엔진을 결합하여 사용하기 때문에 반드시 하나만 선택할 필요는 없음.
  - 그러나 기본적인 추론 메커니즘은 주로 역방향 연결
  - 새로운 사실이 들어왔을 때, 새로운 데이터를 최대로 활용하기 위해서만 순방향 연결을 채택

07_MEDIA ADVISOR: 논증 규칙기반 전문가 시스템

❖ MEDIA ADVISOR

- MEDIA ADVISOR라고 하는 결정 지원 시스템을 만드는 도구로 Leonardo 전문가 시스템 틀을 선택했음.
- 시스템은 직업 훈련을 받는 사람의 역할을 참조해서 훈련 프로그램 수단을 선택하는 데 조언을 해줌.
- 예를 들면, 직업 훈련을 받는 사람이 수력 발전 시스템을 정비하는 기계 기술자라면, 적당한 훈련 프로그램은 워크샵이 될 수 있음.
- 워크샵에서 수력 발전 시스템의 기초적인 구성요소가 어떻게 동작하는지, 수력학에서 발생하는 문제를 어떻게 조정하는지, 간단한 수력 시스템수리를 어떻게 하는지 배울 수 있을 것임.
- 반면, 훈련을 받는 사람이 보험 신청서를 평가하는 직원이라면 훈련 프로그램에 특정 문제에 관한 강의나 실제 작성한 신청서를 평가해 볼 수 있는 튜토리얼을 포함시킬 수 있음.
- 훈련을 받는 사람이 실수하기 쉬운 복잡한 업무에 대해서는, 훈련 프로그램이 진행 상황을 피드백해야 함.

07_MEDIA ADVISOR: 논증 규칙기반 전문가 시스템

❖ 기반지식

- 규칙 1
  - if 작업 환경이 논문이다
  - or 작업 환경이 매뉴얼이다
  - or 작업 환경이 문서다
  - or 작업 환경이 참고서다
  - then 자극_상황이 언어적이다
- 규칙 2
  - if 작업 환경이 그림이다
  - or 작업 환경이 일러스트다
  - or 작업 환경이 사진이다
  - or 작업 환경이 도표다
  - then 자극_상황이 시각적이다
- 규칙 3
  - if 작업 환경이 기계다
  - or 작업 환경이 건물이다
  - or 작업 환경이 도구다
  - then 자극_상황이 '물체'다
- 규칙 4
  - if 작업 환경이 숫자다
  - or 작업 환경이 수식이다
  - or 작업 환경이 '컴퓨터 프로그램'이다
  - then 자극_상황이 기호적이다
- 규칙 5
  - if 업무는 강의하는 것이다
  - or 업무는 조언하는 것이다
  - or 업무는 상담하는 것이다
  - then 자극_반응이 언어적이다
- 규칙 6
  - if 업무는 만드는 것이다
  - or 업무는 수리하는 것이다
  - or 업무는 조정하는 것이다
  - then 자극_반응이 '실무'다

07_MEDIA ADVISOR: 논증 규칙기반 전문가 시스템

- 규칙 7
  - if 업무는 쓰는 것이다
  - or 업무는 타이핑하는 것이다
  - or 업무는 그리는 것이다
  - then 자극_반응이 기록하는 것이다
- 규칙 8
  - if 업무는 평가하는 것이다
  - or 업무는 추론하는 것이다
  - or 업무는 조사하는 것이다
  - then 자극_반응이 분석적이다
- 규칙 9
  - if 자극_상황이 '물체' 다
  - and 자극_반응이 실무다
  - and 피드백이 필요하다
  - then 수단은 워크샵이다
- 규칙 10
  - if 자극_상황이 기호다
  - and 자극_반응이 분석적이다
  - and 피드백이 필요하다
  - then 수단은 '강의-튜토리얼' 이다
- 규칙 11
  - if 자극_상황이 시각적이다
  - and 자극_반응이 기록하는 것이다
  - and 피드백이 필요 없다
  - then 수단은 비디오카세트다
- 규칙 12
  - if 자극_상황이 시각적이다
  - and 자극_반응이 언어적이다
  - and 피드백이 필요하다
  - then 수단은 '강의-튜토리얼' 이다
- 규칙 13
  - if 자극_상황이 문어적이다
  - and 자극_반응이 분석적이다
  - and 피드백이 필요하다
  - then 수단은 '강의-튜토리얼' 이다
- 규칙 14
  - if 자극_상황이 문어적이다
  - and 자극_반응이 언어적이다
  - and 피드백이 필요하다
  - then 수단은 ' 역할 연기 연습' 이다

07_MEDIA ADVISOR: 논증 규칙기반 전문가 시스템

❖ 객체

- MEDIA ADVISOR는 언어 객체 여섯 개(작업 환경, 자극_상황, 역할, 자극_반응, 피드백, 수단)를 사용함.
- 각각의 객체에는 허용된 값(예를 들면, 작업 환경 객체에는 논문, 매뉴얼, 문서, 참고서, 그림, 일러스트, 사진, 도표, 기계, 건물, 도구, 숫자, 수식, 컴퓨터 프로그램의 값이 들어간다) 중 하나가 들어감.
- 하나의 객체와 값은 하나의 사실을 나타낸다(예를 들면, 작업 환경이 기계고 업무가 수리하는 것이다). 모든 사실은 데이터베이스에 저장됨.

객체	허용된 값	객체	허용된 값
작업 환경	논문	업무	강의하는 것
	매뉴얼		조언하는 것
	문서		상담하는 것
	참고서		만드는 것
	그림		수리하는 것
	일러스트		조정하는 것
	사진		쓰는 것
	도표		타이핑하는 것
	기계		그리는 것
	건물		평가하는 것
	도구		추론하는 것
	숫자		조사하는 것
	수식		
	컴퓨터 프로그램	자극_반응	언어적
			실무
			기록하는 것
			분석적
자극_상황	문어적		
	시각적		
	물건	피드백	필요하다
	기호적		필요 없다

07_MEDIA ADVISOR: 논증 규칙기반 전문가 시스템

❖ 선택지

- 규칙기반 전문가 시스템의 최종 목표는 입력 데이터에 기반하여 문제에 대한 해를 제공하는 것임.
- MEDIA ADVISOR에서의 해는 다음 네 가지 선택지에서 선택한 수단을 뜻함.
  - 수단은 워크샵이다
  - 수단은 '강의-튜토리얼' 이다
  - 수단은 비디오카세트다
  - 수단은 '역할 연기 연습' 이다

❖ 대화

- 다음 대화와 같이, 전문가 시스템에서는 문제를 풀기 위해 필요한 데이터(작업 환경, 업무, 피드백)를 사용자가 입력해야 함.
- 사용자에게서 받은 대답(대답은 화살표로 표시한다)에 기반해서, 전문가 시스템은 자신의 기반지식으로부터 규칙을 적용하여 자극_상황이 물체고 자극_반응이 실무라는 것을 추론해냄.
- 그런 다음 규칙9는 허용된 수단 값들 중 하나를 선택한다.

07_MEDIA ADVISOR: 논증 규칙기반 전문가 시스템

❖ 대화

- 훈련받는 사람은 업무에서 어떤 종류의 환경을 다루는가? ⇒ 기계
- 규칙 3
  - if 작업 환경이 기계다
  - or 작업 환경이 건물이다
  - or 작업 환경이 도구다
  - then 자극_상황이 '물체' 다

- 훈련받는 사람은 업무에서 행동이나 대응을 어떻게 해야 하는가? ⇒ 수리
- 규칙 6
  - if 업무는 만드는 것이다
  - or 업무는 수리하는 것이다
  - or 업무는 조정하는 것이다
  - then 자극_반응이 '실무'다

- 훈련받는 사람의 진행 상황에 대한 피드백이 필요한가? ⇒ 필요하다
- 규칙 9
  - if 자극_상황이 '물체' 다
  - and 자극_반응이 '실무' 다
  - and 피드백이 필요하다
  - then 수단은 워크샵이다
- 수단은 워크샵이다

07_MEDIA ADVISOR: 논증 규칙기반 전문가 시스템

❖ 추론 기법

- Leonardo는 사용 가능한 정보를 가장 효과적으로 다룰 수 있는 역방향 연결을 표준 추론 기법으로 사용하며, 필요에 따라 순방향 연결을 사용함.
- 사용자가 Leonardo의 역방향 또는 순방향 연결 기능을 실행하거나 정지하도록 설정할 수 있기 때문에, 추론 기법을 각각 적용할 수도 있음.
- 순방향 연결은 데이터 지향 추론이므로 먼저 데이터를 입력해야 함.
  - 작업 환경이 기계다

    '작업 환경'이 사용자 입력을 통해 '기계'로 설정되었다
  - 업무는 수리하는 것이다

    '업무'가 사용자 입력을 통해 '수리하는 것'으로 설정되었다
  - 피드백이 필요하다

    '피드백'이 사용자 입력을 통해 '필요하다'로 설정되었다

- 그러면 다음 과정이 진행될 것이다.
  - 규칙 30이 점화된다 '자극_상황' 이 규칙 3을 통해 '물체' 로 설정된다
  - 규칙 60이 점화된다 '자극_반응' 이 규칙 6을 통해 '실무' 로 설정된다
  - 규칙 9가 점화된다 '수단' 이 규칙 9를 통해 '워크샵' 으로 설정된다
  - 아무 규칙도 점화되지 않는다 중단한다

07_MEDIA ADVISOR: 논증 규칙기반 전문가 시스템

❖ 추론 기법

- 역방향 연결은 목표 지향 추론이므로, 우리는 먼저 가정해(목표)를 설정할 필요가 있으므로, 다음과 같은 목표를 설정해보자.

- '수단' 은 '워크샵' 이다
- 1단계
  - 규칙 9를 시도 '자극_상황' 객체를 찾아야 함
  - 규칙 9를 스택에 넣음 '자극_상황' 객체를 '물체' 로 놓음

- 2단계
  - 규칙 3을 시도 '작업 환경' 객체를 찾아야 함
  - 규칙 3을 스택에 넣음 '작업 환경' 객체를 '기계' 로 놓음
- 작업 환경을 물어본다
  - ⇒ 기계 '작업 환경' 이 사용자 입력을 통해 '기계' 로 설정됨
  - 규칙 3을 시도 '자극_상황' 이 규칙 3을 통해 '물체' 로 설정됨

- 3단계
  - 규칙 9를 시도 '자극_반응' 객체를 찾아야 함
  - 규칙 9를 스택에 넣음 '자극_반응' 객체를 '실무' 로 놓음

07_MEDIA ADVISOR: 논증 규칙기반 전문가 시스템

❖ 추론 기법

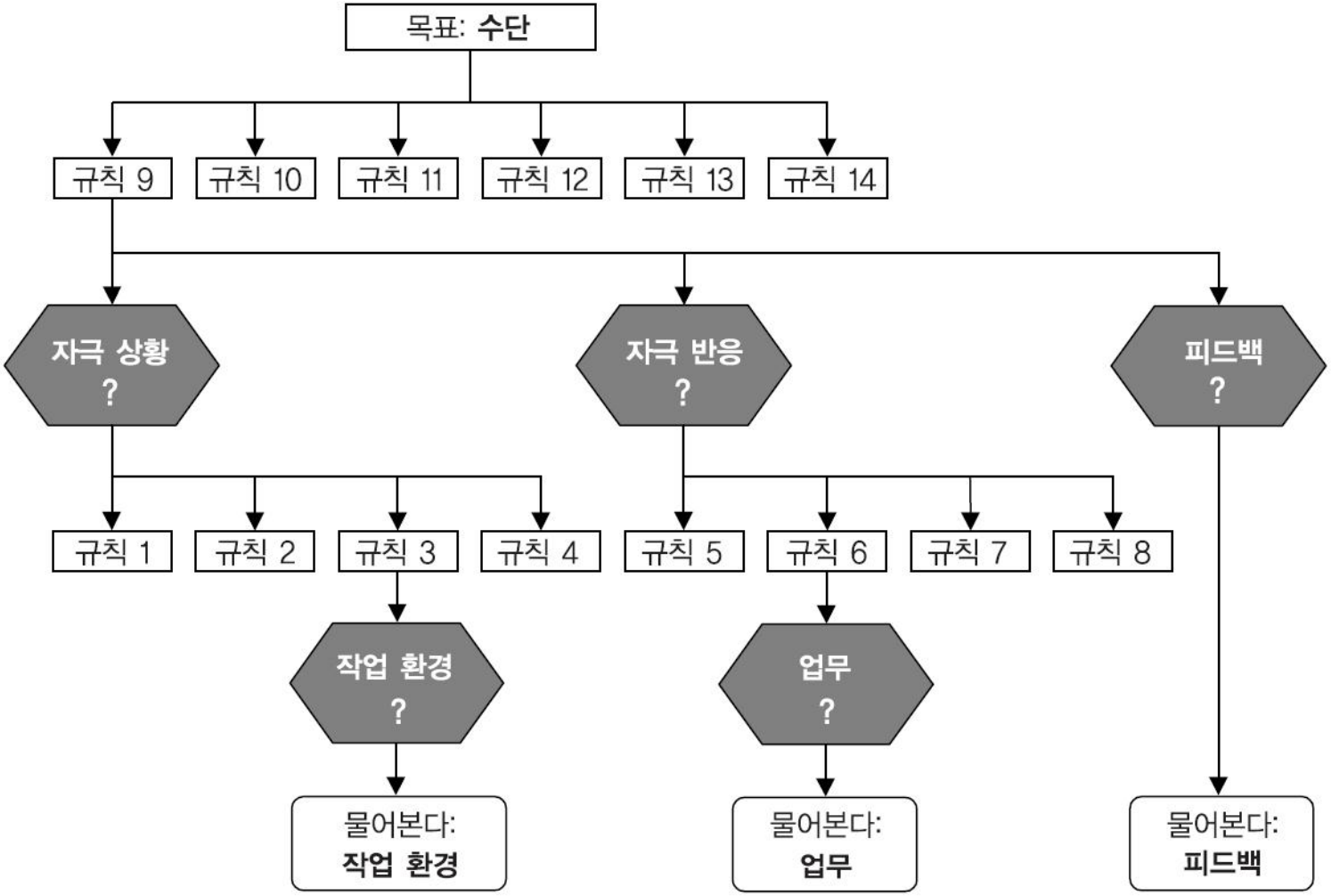
- 4단계
  - 규칙 6을 시도 '업무' 객체를 찾아야 함
  - 규칙 6을 스택에 넣음 '업무' 객체를 '만드는 것' 으로 놓음
  - 업무를 물어본다
  - ⇒ 수리하는 것 '업무' 가 사용자 입력을 통해 '수리하는 것' 으로 설정됨
  - 규칙 6을 시도 '자극_반응' 이 규칙 6을 통해 '실무' 로 설정됨

- 5단계
  - 규칙 9를 시도 '피드백' 객체를 찾아야 함
  - 규칙 9를 스택에 넣음 '피드백' 객체를 '필요하다' 로 놓음
  - 피드백을 물어본다
  - ⇒ 필요하다 '피드백' 이 사용자 입력을 통해 '필요하다' 로 설정됨
  - 규칙 9를 시도 '수단' 이 규칙 9를 통해 '워크샵 ' 으로 설정됨

- 수단은 워크샵이다

07_MEDIA ADVISOR: 논증 규칙기반 전문가 시스템

❖ 규칙기반 전문가 시스템 MEDIA ADVISOR에 대한 트리 형식의 도표



07_MEDIA ADVISOR: 논증 규칙기반 전문가 시스템

- MEDIA ADVISOR가 모든 모든 상황을 다루지는 못한다.
- 다음과 같은 대화가 있을 수 있다. 상황을 다룰 수 있는가
  - 위의 전문가 시스템이 제공한 선택지가 훈련받는 사람은 업무에서 어떤 종류의 환경을 다루는가?
    - ⇒ 일러스트
  - 훈련받는 사람이 업무에서 어떤 방식으로 행동하거나 대응해야 하는가?
    - ⇒ 그리기
  - 훈련 중 훈련받는 사람의 진행 상황에 대한 피드백이 필요한가?
    - ⇒ 필요하다
  - 이 데이터에 대해 어떠한 결론도 내릴 수 없습니다.
  - 즉, 현 상태의MEDIA ADVISOR는 이 상황을 다룰 수 없음.
  - 하지만 사용자가 원하는 일을 해낼 때까지 다른 규칙을 수용하여 전문가 시스템을 쉽게 확장시킬 수 있음.

08_충돌 해법

❖ 앞에서 길을 건너는 두 개의 간단한 규칙을 살펴봤다.

❖ 여기에 세 번째 규칙을 추가해 본다.

❖ 규칙 1
- IF '신호등' 이 녹색이다
- THEN 길을 건넌다

❖ 규칙 2
- IF '신호등' 이 빨간색이다
- THEN 멈춰 기다린다

❖ 규칙 3
- IF '신호등' 이 빨간색이다
- THEN 길을 건넌다

08_충돌 해법

- 어떤 일이 일어나는가
  - 추론 엔진은 규칙의 IF(조건) 부분을 데이터베이스에 있는 데이터와 비교하여 조건을 만족하면 그규칙을 점화 상태로 설정한다.
  - 어느 한 규칙의 점화는 다른 규칙을 활성화하는 데 영향을 주기 때문에 추론 엔진은 반드시 한 번에 하나의 규칙만 점화해야 한다.
  - 앞에 나온 길 건너기 예에서 규칙2와 규칙 3은 IF 부분이 동일하다. 그러므로 조건을 만족하는 두 규칙은 점화할 것이다.
  - 이런 규칙의 집합을 충돌 집합(conflict set)이라 한다.
  - 주어진 사이클에서 두 개 이상의 규칙이 점화되었을 때 그 중 어느 규칙이 점화할지 선택하는 방법을 충돌 해법(conflict resolution)이라 한다.
- 신호등이 빨간색이라면 어떤 규칙을 수행해야 하는가
  - 순방향 연결을 사용하는 경우에는 규칙 두 개가 점화됨.
  - 규칙 2가 가장 위에 있는 규칙이므로 먼저 점화되고, THEN 부분이 수행되어 언어 객체 취해야 할 행동은 멈춰 '기다린다'라는 값을 얻음.
  - 그러나 규칙3의 조건 부분 역시 데이터베이스에 있는 사실 '신호등' 이 '빨간색이다' 와 부합되므로 이 규칙 또한 점화됨.
  - 그 결과로 취해야 할 행동 객체는 새로운 값인 '길을 건넌다'를 얻음.
  - 이 예는 순방향 연결 추론 기법을 사용할 때 규칙의 순서가 아주 중요하다는 것을 보여즘.

08_충돌 해법

❖ 충돌을 어떻게 해결할 수 있는가

- 충돌을 해결할 수 있는 명백한 전략은 목표를 설정하고, 설정한 목표에 도달하면 규칙 수행을 멈추는 것
- 예를 들어, 위 문제의 목표는 언어 객체 취해야 할 행동의 값을 정하는 것임.
- 전문가 시스템이 취해야 할 행동의 값을 결정했다면 이는 목표에 도달한 것이므로 규칙 수행을 멈춤.
- 따라서 신호등이 빨간색이다라는 규칙2가 실행되면 멈춰 기다린다라는 값이 들어가고, 전문가시스템은 멈춤.
- 주어진 예에서 전문가 시스템은 옳은 결정을 내렸음.
- 그러나 규칙을 반대로 정렬했다면 결론은 잘못되었을 것
- 이는 기반지식 내에서 규칙의 순서가 매우 중요하다는 사실을 의미

08_충돌 해법

❖ 다른 충돌 해법이 있는가

- 우선 순위 전략
  - 가장 높은 우선순위를 가진 규칙을 점화
  - 간단한 응용 사례에서는 기반지식에 적당한 순서로 규칙을 배치함으로써 우선순위를 정함.
  - 그러나 어떤 응용 사례에서는 중요한 순서에 따라 데이터를 처리해야 함.
- 목표 1: 처방전
- 규칙 1: 뇌막염 처방전 1 (우선순위 100)
  - IF 감염이 뇌막염이다
  - AND 환자가 어린이다
  - THEN 처방전은 Number_1이다
  - AND 추천 약은 암피실린(ampicillin)이다
  - AND 추천 약은 겐타마이신(gentamicin)이다
  - AND 뇌막염 처방전 1을 보여준다
- 규칙 2: 뇌막염 처방전 2(우선순위 90)
  - IF 감염이 뇌막염이다
  - AND 환자가 어른이다
  - THEN 처방전은 Number_2이다
  - AND 추천 약은 페니실린(penicillin)이다
  - AND 뇌막염 처방전 2를 보여준다

08_충돌 해법

❖ 다른 충돌 해법이 있는가

- 최장 일치 전략(longest matching strategy)
  - 가장 특수한 규칙을 점화시킴.
  - 특수한 규칙일수록 일반적인 규칙보다 더 많은 정보를 처리할 것이라는 가정에 근거

- 규칙 1
  - IF 가을이다
  - AND 하늘이 흐리다
  - AND 일기예보에서는 비가 온다고 한다
  - THEN 조언은 '집에 머무르시오' 다
- 규칙 2
  - IF 가을이다
  - THEN 조언은 '우산을 가져가시오' 다

- 만일 계절이 가을이고 하늘이 흐리며 일기예보에서는 비가 온다고 했다면, 규칙1의 전건인 매칭부분이 규칙 2보다 좀 더 특수하기 때문에 규칙 1이 점화될 것
- 그러나 계절이 가을이라는 것만 알려져 있다면 규칙2가 수행될 것

참고 – 오컴의 면도날 (Occam's Razor, Ockham's Razor)

- 흔히 경제성의 원리라고도 함
- 오컴의 저서에 등장하는 원문
  - "Pluralitas non est ponenda sine neccesitate." (필요하지 않은 경우에까지 많은 것을 가정하면 안 된다)
  - "Frustra fit per plura quod potest fieri per pauciora." (보다 적은 수의 논리로 설명이 가능한 경우, 많은 수의 논리를 세우지 말라.)

- 예를 들어, 새까맣게 그을린 나무가 있다고 가정하자.
  - 이는 나무가 벼락에 맞았기 때문이거나,
  - 아니면 누군가가 어떤 장치를 이용해서 나무가 완전히 잿더미로 변하지 않도록 적절히 그을린 다음 자신이 그을렸다는 흔적을 완전히 없앤 것일 수도 있다.
- 이 상황을 판단할 다른 증거가 없는 경우 오컴의 면도날을 적용해 본다면, 나무가 그을린 것은 벼락에 맞았기 때문이라고 추론하는 것이 더 적합할 수 있다. 왜냐하면, 벼락에 맞았다는 쪽이 조건을 덜 필요로 하기 때문이다.

08_충돌 해법

❖ 다른 충돌 해법이 있는가

- 최근 입력 데이터 사용 전략
  - 데이터베이스에 가장 최근에 입력된 데이터(data most recently entered)를 사용하여 규칙을 점화
  - 이 방법은 데이터베이스에 있는 각각의 사실에 붙여진 시간 태그에 의존
  - 충돌 집합 중에서, 전문가 시스템은 전건에 가장 최근에 데이터베이스에 추가된 데이터가 들어있는 규칙을 먼저 점화

- 규칙 1
  - IF 일기예보에서는 비가 온다고 한다 [96/11/25 08:16 PM]
  - THEN 조언은 '우산을 가져가시오' 다
- 규칙 2
  - IF 비가 온다 [96/11/26 10:18 AM]
  - THEN 조언은 '집에 머무르시오' 다

- 두 규칙의 IF 부분이 데이터베이스에 있는 사실과 일치한다고 가정하자.
- 이때 비가 온다는 사실이 일기예보에서는 비가 온다고 한다라는 사실 다음에 입력되었기 때문에 규칙 2가 점화될 것
- 이 기법은 데이터베이스의 정보가 자주 업데이트되는 실시간 전문가 시스템에 특히 유용함.

08_충돌 해법

❖ 다른 충돌 해법이 있는가

- 앞에서 설명한 충돌 해법은 쉽게 구현할 수 있으며 대부분 만족스런 해를 제공함.
- 그러나 프로그램이 커지고 복잡해질수록 지식 공학자가 기반지식에 있는 규칙을 관리하고 감독하기가 점점 어려워짐.
- 그러므로 전문가 시스템 자신이 적어도 그 부담 중 일부를 맡고, 그 자신의 작동 방식을 이해해야 함.

❖ 메타지식(metaknowledge)

- 전문가 시스템의 성능을 개선하려면 전문가 시스템에 수록된 지식에 관한 지식, 즉 메타지식(metaknowledge)을 시스템에 제공해야 함.
- 메타지식은 지식에 관한 지식(knowledge about knowledge)으로 간단하게 정의할 수 있는데 이는 전문가 시스템 내에 있는 분야 지식을 사용하고 제어하는 데 필요한 지식
- 규칙기반 전문가 시스템에서는 메타지식을 메타규칙으로 표현
- 메타규칙은 전문가 시스템 업무와 관련된 규칙을 어떻게 사용할 것인지에 대한 전략을 결정

08_충돌 해법

❖ 메타지식의 근원은 무엇인가

- 지식 공학자는 주제 전문가의 지식을 전문가 시스템으로 전달하고 문제에 관련된 규칙을 어떻게 사용할지 배우며, 점차적으로 전문가 시스템의 전체적인 작동에 관한 지식과 새로운 지식의 형태를 머릿속에 만들어나감.
- 즉, 메타지식은 분야에 종속적이다.
- 메타규칙 1
  - 전문가가 제공한 규칙은 무경험자가 제공한 규칙보다 우선순위가 높다
- 메타규칙 2
  - 인명 구조를 다루는 규칙은 발전 시스템 장비의 부하를 처리하는 규칙보다 우선순위가 높다

❖ 전문가 시스템은 메타규칙을 이해하고 사용할 수 있는가

- 어떤 전문가 시스템은 메타규칙을 위한 추론 엔진을 따로 제공
- 그러나 대부분의 전문가 시스템은 규칙과 메타규칙을 구분하지 못함.
- 따라서 메타규칙은 기반지식 내에서 우선순위를 높게 설정해야 함.
- 메타규칙이 점화되면, 다른 규칙의 우선순위를 바꿀 수도 있는 중요한 정 보가 데이터베이스에 저장

09_규칙기반 전문가 시스템의 장점과 단점

❖ 지식 공학자는 왜 규칙기반 전문가 시스템을 선호할까

- **자연스러운 지식 표현**(Natural knowledge representation)
  - 전문가는 문제 풀이 과정을 '이러이러한 상황에서, 나는 이러이러한 것을 한다' 와 같은 식으로 설명한다. 이런 표현은 IF-THEN생성 규칙으로 아주 자연스럽게 표현할 수 있음.
- **통일된 구조**(Uniform structure)
  - 생성 규칙은 통일된 IF-THEN 구조를 가지고 있다. 각각의 규칙은 독립적인 지식 조각임.
  - 생성 규칙의 문법이 체계적이기 때문에 별다른 설명이 없이 규칙을 쉽게 이해할 수 있음
- **지식과 과정의 분리**(Separation of knowledge from its processing)
  - 규칙기반 전문가 시스템의 구조는 기반지식과 추론 엔진을 효율적으로 분리하여, 똑같은 전문가 시스템 틀로 서로 다른 응용 시스템을 개발할 수 있음.
  - 또한 전문가 시스템을 우아하고 손쉽게 확장할 수 있도록 해줌.
  - 시스템을 더욱 영리하게 만들려면 지식 공학자가 제어 구조에 방해가 되지 않는 범위 내에서 기반지식에 규칙을 약간 추가하면 가능
- **불완전하고 불확실한 지식 다루기**(Dealing with incomplete and uncertain knowledge):
  - 규칙기반 전문가 시스템은 불완전하고 불확실한 지식을 표현하고 추론할 수 있음.

09_규칙기반 전문가 시스템의 장점과 단점

❖ 지식 공학자는 왜 규칙기반 전문가 시스템을 선호할까

- **불완전하고 불확실한 지식 다루기**(Dealing with incomplete and uncertain knowledge):
  - 규칙기반 전문가 시스템은 불완전하고 불확실한 지식을 표현하고 추론할 수 있음.
  - 다음과 같은 규칙을 보자.

  - IF 가을이다
  - AND 하늘이 흐리다
  - AND 바람이 약하다

  - THEN 일기예보는 맑음이다 {0.1}
    일기예보는 보슬비다 {1.0}
    일기예보는 비다 {0.9}

  - 위 규칙은 '가을이고 보슬비가 올 것 같다면 아마 오늘 중으로 비가 내릴 것이다' 와 같이 불확실성을 나타내는 데 사용될 수 있음
  - 규칙의 불확실성은 {0.1}와 같이 확신도(certainty factors)라고 하는 숫자로 표현한다. 전문가 시스템은 확신의 정도나 규칙이 참이라는 신뢰의 수준을 나타내는 데 확신도를 사용함.

09_규칙기반 전문가 시스템의 장점과 단점

❖ 모든 문제에 규칙기반 전문가 시스템을 적용할 수 있는가

- **규칙 간의 불분명한 관계**(Opaque relations between rules)
  - 개별적인 생성 규칙이 상대적으로 간단하고 이해하기 쉽다고 하더라도, 많은 규칙으로 이루어진 규칙 집합 안에서는 규칙의 논리적인 상호관계가 확실하지 않을 수 있음.
  - 그리고 규칙기반 시스템에서는 개별적인 규칙이 전체 전략에 어떻게 기여하는가를 관찰하기 어려움.
  - 이 문제는 규칙기반 전문가 시스템에서 계층적인 지식 표현이 부족하기 때문에 생긴다.

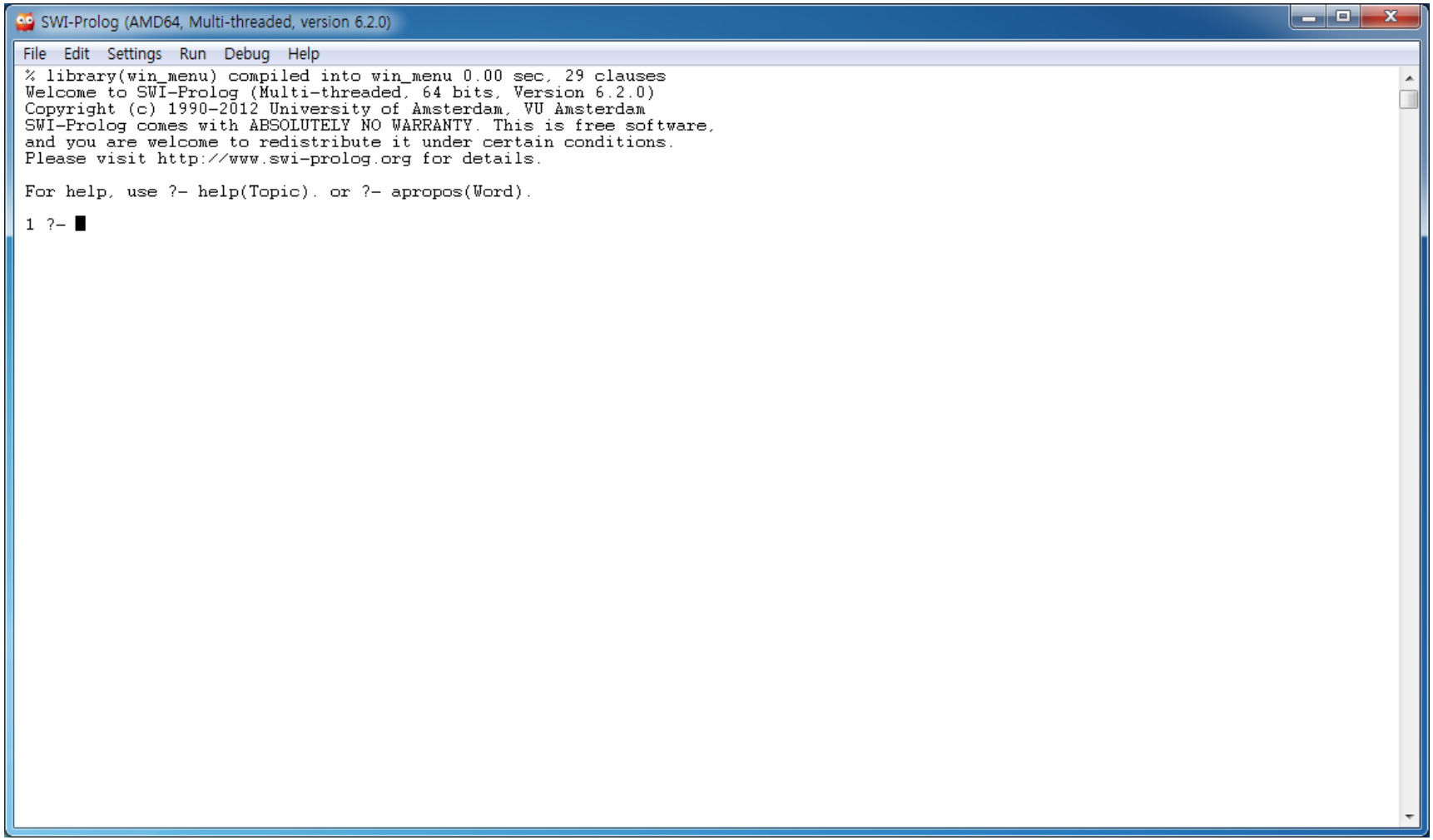
- **비효율적인 탐색 전략**(Ineffective search strategy)
  - 추론 엔진은 각각의 사이클 동안 모든 생성 규칙을 철저히 탐색
  - 많은 규칙으로 이루어진 규칙 집합(100개 이상의 규칙)을 포함하는전문가 시스템은 실행 속도가 느려질 수도 있으므로 큰 규모의 규칙기반 시스템은 실시간 응용사례에는 부적합

- **학습할 수 없음**(Inability to learn)
  - 일반적으로, 규칙기반 전문가 시스템에는 경험을 통해 배우는 능력이 없음.
  - 언제 '규칙을 변경할' 지 아는 인간 전문가와 달리, 전문가 시스템은 자동으로 자신의 기반지식을 수정하거나 원래 있던 규칙을 조정하거나 새 규칙을 추가하지 못함.
  - 따라서 지식 공학자가 계속 시스템을 수정하고 유지해야 함.

전문가 시스템 개발용 소프트웨어

- Java Expert System Shell (JESS) – Rete inference algorithm
- Prolog – SWI-Prolog
  - http://www.swi-prolog.org/

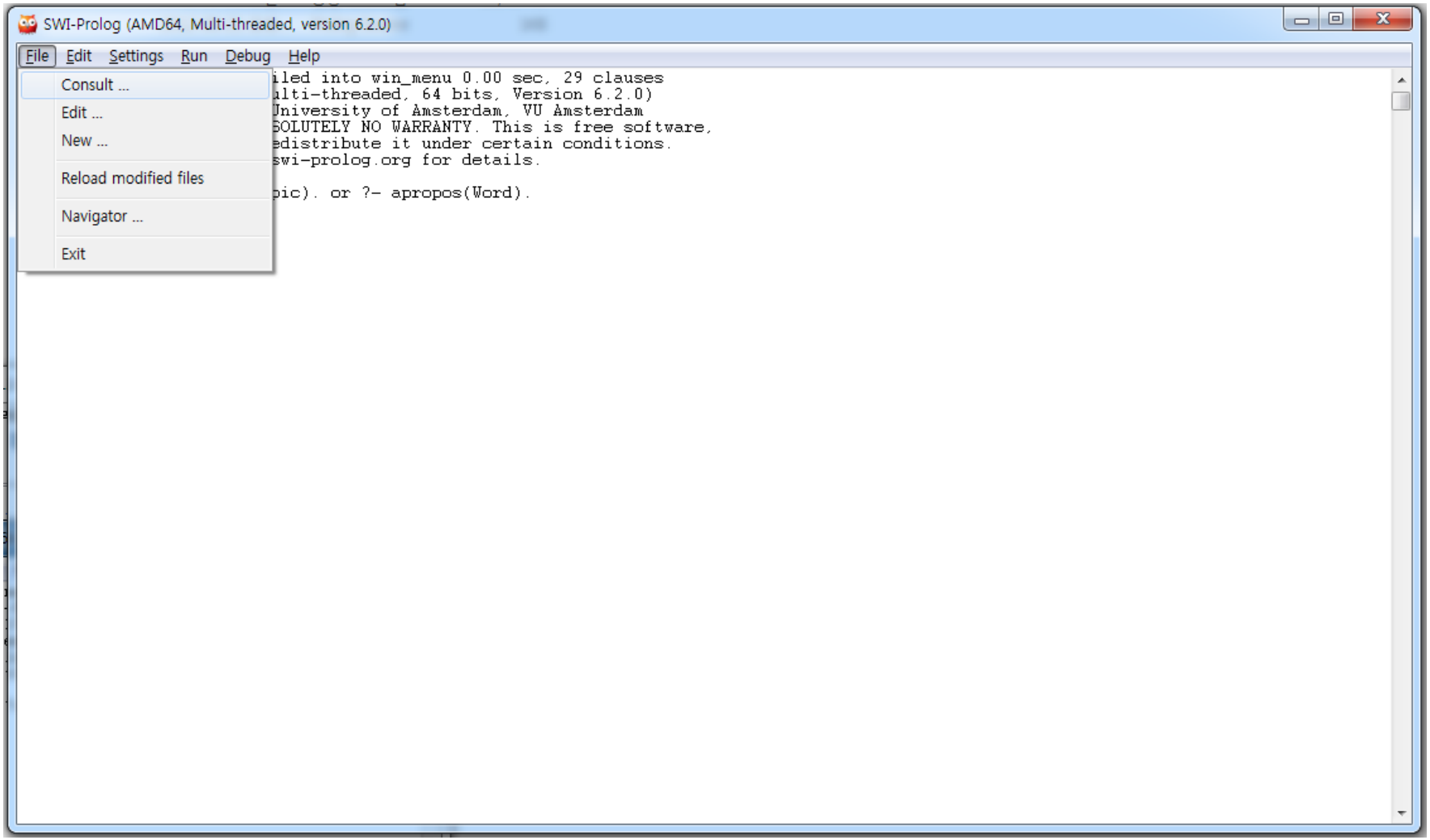
SWI-Prolog 실행



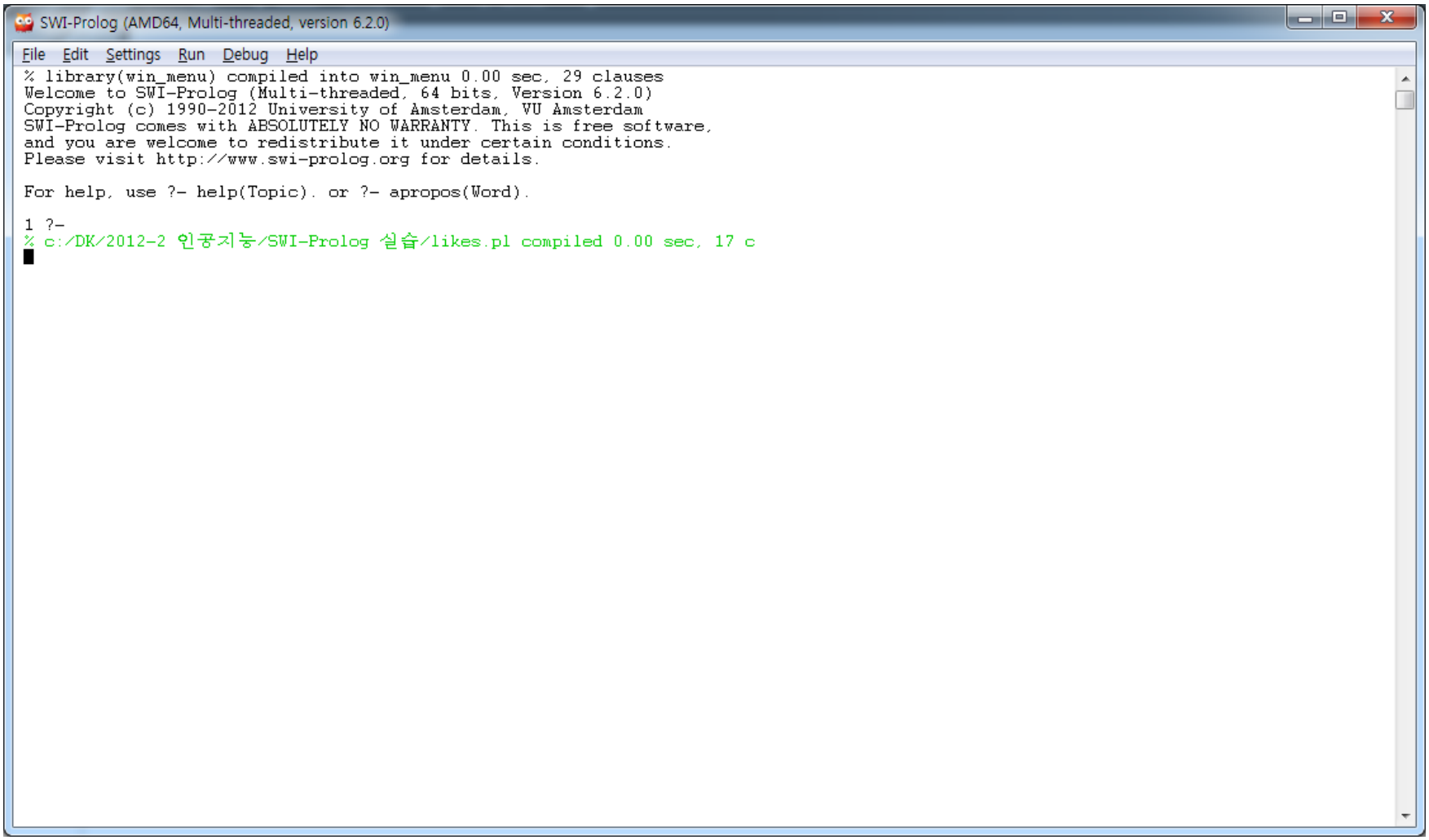
예제 likes.pl

```
likes(sam,Food) :-      indian(Food),      mild(Food).
likes(sam,Food) :-      chinese(Food).
likes(sam,Food) :-      italian(Food).
likes(sam,chips).

indian(curry).
indian(dahl).
indian(tandoori).
indian(kurma).

mild(dahl).
mild(tandoori).
mild(kurma).

chinese(chow_mein).
chinese(chop_suey).
chinese(sweet_and_sour).

italian(pizza).
italian(spaghetti).
```

File -> Consult 로 프롤로그 파일을 읽어들임



likes.pl 을 읽어들인 모습



likes.pl 실행 결과

```
1 ?-
|    likes(sam,dahl).
true .

2 ?- likes(sam,chow_mein).
true .

3 ?- likes(sam,pizza).
true .

4 ?- likes(sam,curry).
false.

5 ?-
```

교재의 예제

```
z :- y, d.
y :- x,   b, e.
x :- a.
l :- c.
n :- l, m.

a.
b.
c.
d.
e.
```

교재의 예제 실행 결과

```
1 ?- m.
ERROR: toplevel: Undefined procedure: m/0 (DWIM could not correct goal)
2 ?- n.
ERROR: n/0: Undefined procedure: m/0
  Exception: (7) m ? creep
3 ?- a.
true.

4 ?- b.
true.

5 ?- z.
true.

6 ?- y.
true.

7 ?- x.
true.

8 ?- l.
true.

9 ?- n.
ERROR: n/0: Undefined procedure: m/0
  Exception: (7) m ? creep
10 ?- m.
ERROR: toplevel: Undefined procedure: m/0 (DWIM could not correct goal)
11 ?-
|    c.
true.

12 ?- d.
true.

13 ?- e.
true.
```

10_요약

❖ 규칙기반 전문가 시스템 요약

- 지식은 어떤 주제나 분야에 대해 이론적 혹은 실제로 이해한 것을 말하고, 이는 현재 알려진 것의 총집합임.

- 전문가는 특정 분야에 관해 사실과 규칙의 형태로 표현할 수 있는 깊은 지식과 다양한 실제 경험을 갖춘 사람이고, 다른 사람이 할 수 없는 일을 할 수 있다.

- 전문가는 보통 자신의 지식을 생성 규칙의 형태로 표현하는데,생성 규칙은 IF(전건) THEN(후건) 문으로 나타낸다. 생성 규칙은 지식을 표현할 때 가장 많이 쓰이는 유형으로 규칙은 관계, 추천, 지시, 전략, 휴리스틱을 나타낼 수 있음

- 특정 분야의 문제에 관해서 전문가 수준으로 동작할 수 있는 컴퓨터 프로그램을 전문가 시스템이라 한다. 가장 널리 쓰이는 전문가 시스템은 규칙기반 전문가 시스템이다.

- 전문가 시스템 개발팀에는 주제 전문가, 지식 공학자, 프로그래머, 프로젝트 관리자, 최종 사용자가 있어야 한다.
  - 지식 공학자는 전문가 시스템을 설계하여 만들고 테스트하며
  - 주제 전문가에게서 지식을 얻어서 추론 방법을 정하고 개발 소프트웨어를 고른다.

10_요약

❖ 규칙기반 전문가 시스템 요약

- 규칙기반 전문가 시스템의 기본 구성요소에는 기반지식, 데이터베이스, 추론 엔진, 해설 설비,사용자 인터페이스 등 다섯 가지가 있다.
  - 기반지식은 규칙의 집합으로 표현한 분야 지식을 포함한다.
  - 데이터베이스는 사실 집합을 저장하고 있는데, 사실은 기반지식에 저장된 규칙의 IF 부분과 일치하는지 비교할 때 쓰인다.
  - 추론 엔진은 규칙을 사실과 연결하고 전문가 시스템이 해를 구할 때까지 추론을 수행한다.
  - 해설 설비는 사용자에게 전문가 시스템이 어떻게 특정 결론에 이르렀으며, 왜 특정 사실이 필요한지 설명한다.
  - 사용자 인터페이스는 사용자와 전문가 시스템 간의 통신 수단이다.

- 전문가 시스템
  - 전문가 시스템은 기반지식과 추론 엔진을 분리하여 지식과 처리 과정을 분리하였다. 이를 통해시스템을 쉽게 만들고 효율적으로 유지할 수 있다.
  - 전문가 시스템은 문제 풀이 세션 중에 점화된 규칙을 추적하여 제한적인 해설을 제공한다.

10_요약

- 전문가 시스템
  - 전통적인 프로그램과 달리 전문가 시스템은 불완전하고 불확실한 데이터를 다룰 수 있고 부정확한 추론을 허용한다. 그러나 사람과 마찬가지로 전문가 시스템도 정보가 불완전하고 불분명할 경우에는 실수할 수 있다.

- 직접적인 탐색과 추론 방법에는 순방향 연결과 역방향 연결 추론 기법이 있다.
  - 순방향 연결은 데이터 지향 추론이다. 이는 알려진 데이터부터 시작해 더 이상 점화되는 규칙이 없을 때까지 순방향으로 진행한다.
  - 역방향 연결은 목표 지향 추론으로, 전문가 시스템이 목표(가정해)를 정하고, 그 추론 엔진은 이를 증명하기 위해 증거 찾기를 시도한다.

- 주어진 사이클에서 규칙을 하나 이상 점화할 수 있다면 추론 엔진은 어떤 규칙을 점화할 것인지 반드시 결정해야 한다. 어떤 규칙을 선택할지 결정하는 방법을 충돌 해법이라 한다.

- 규칙기반 전문가 시스템
  - 규칙기반 전문가 시스템은 자연스러운 지식 표현, 통일된 구조, 처리 과정과 지식의 분리, 불완전하고 불확실한 지식을 다룬다는 것이 장점이다.
  - 규칙기반 전문가 시스템의 단점은 규칙들 간의 모호한 관계, 비효율적인 탐색 전략, 학습이 불가능하다는 점이다.

Thank You !