🎉

3 / 23

- 지식
  - 지식은 어떤 주제나 분야에 대해 이론적으로 혹은 실제로 이해하는 것
  - 현재 알려진 사실들의 모음, 권력이 되기도 함
  - 지식을 소유한 사람을 전문가라 함
  - 전문가는 조직에서 가장 영향력있고, 핵심적인 사람
    - 특정 분야에 해박한 지식과 풍부한 경험을 쌓은 사람, 숙련된 사람
    - 전문가는 IF THEN의 구조로 이해
    - IF THEN의 규칙의 문법
      - 조건 IF 행동 THEN
      - 전건과 후건으로도 나눔
  - 규칙은 관계, 추천, 지시, 전략, 휴리스틱을 표현할 수 있다.

- 규칙은 관계, 추천, 지시, 전략, 휴리스틱을 표현할 수 있다.

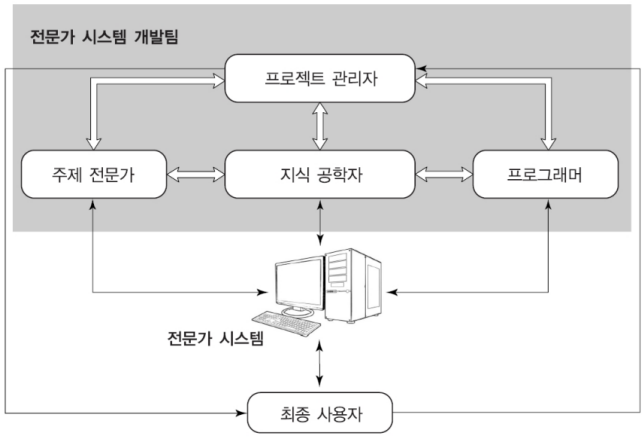
  - 【전략】
    - IF 차가 멈췄다
    - THEN ‘연료 탱크를 확인’한다
      1단계를 완료했다

    - 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

- 전문가 시스템 구성
  - 전문가 시스템
    - 전문가 수준에서 동작 할 수 있는 유능한 프로그램
    - 가장 인기 있는 전문가 시스템은 규칙기반 시스템이다.
  - 전문가 시스템 틀
    - 전문가 시스템 틀은 지식을 추가하지 않은 전문가 시스템
    - 모든 사용자는 지식을 규칙형식으로 추가하고, 문제를 풀기 위해 필요한 데이터를 제공해야함
  - 전문가 시스템 개발
    - 주제 전문가, 지식 공학자, 프로그래머, 프로젝트 관리자, 최종 사용자까지 총 5명으로 구성
    - 성공 여부는 각 구성원이 협업을 얼마나 잘하냐에 따라 달림

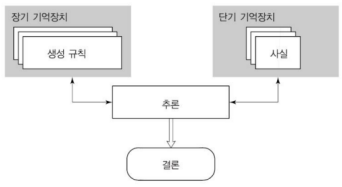
      - 개발팀의 기본 관계



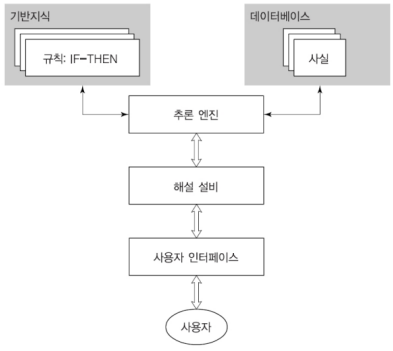
  - 전문가 시스템 개발의 주요 구성원
    - 주제 전문가
      - 특정 분야나 주제(Domain)에 대한 지식이 풍부하고, 관련 문제를 푸는데 능숙한 사람으로 주제에 대해 최고 전문 지식을 갖추어야 함
      - 주제 전문가의 전문 지식은 전문가 시스템에 저장

    - 전문가는 자신의 지식을 전달해야 하고, 시스템 개발에 참여해야 하며, 프로젝트에 많은 시간을 투자해야 함
    - 주제 전문가는 전문가 시스템 개발에서 가장 중요한 사람임
- 지식 공학자
    - 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, 사실과 규칙을 다루기 위해 어떤 추론 방법을 사용했는지 알아내고, 전문가 시스템에서 이를 어떻게 표현할지 결정
    - 그 후 지식 공학자는 기존의 개발 소프트웨어나 전문가 시스템 틀을 선택하고 지식을 표현할 프로그래밍 언어를 살펴봄
    - 마지막으로, 실제 작업 영역으로 옮겨 테스트하고 수정하여 통합, 지식 공학자는 전문가 시스템의 초기 설계 단계부터 최종 단계, 그리고 프로젝트가 완료 후에도 시스템을 유지하는 일에 참여함
- 프로그래머
    - 프로그래밍을 책임지며, 컴퓨터가 이해할 수 있는 용어로 기술하는 사람
    - LISP, Prolog, OPS5같은 AI언어를 프로그래밍할 줄 알아야 하고, 다른 형태의 여러 전문가 시스템 틀을 응용한 경험이 있어야하고, C, Pascal, FORTRAN, Basic 같은 전통적인 프로그래밍 언어도 알아야 함
    - 전문가 시스템 틀을 사용할 때는 지식 공학자가 전문가 시스템에 쉽게 코딩할 수 있어서, 프로그래머가 필요하지 않을 수 있다
    - 지식, 데이터 표현 구조, 제어 구조, 대화 구조 등을 개발해야 하고, 전문가 시스템을 테스트할 때에도 참여할 수 있다.
- 프로젝트 관리자
    - 전문가 시스템 개발팀의 리더로서, 프로젝트를 제대로 진행할 수 있도록 관리
    - 실행 가능한 모든일과 목표를 충족시키고, 다른 역할의 구성원과 상호작용
- 전문가 시스템 개발의 주요 구성원
    - 최종 사용자

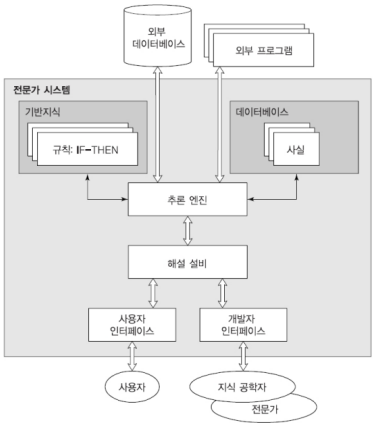
        - 개발한 전문가 시스템을 사용하는 사람을 말함
        - 사용자는 화성 토양의 분자 구조를 연구하는 분석 화학자 (Feigenbaum 외, 1971), 전염성 순환계질환을 진단해야 하는 경험이 부족한 의사(Shortliffe, 1976), 새로운 광물 매장지를 발견하고자 하는 탐사 지리학자(Duda 외,1979), 또는 응급 시에 조언을 받고 싶은 발전 시스템 기사일 수도 있다(Negnevitsky,1976).
- 규칙기반 전문가 시스템의 탄생
    - 1970년대 초반, 카네기멜론 대학교의 뉴웰과 사이먼은 현대의 규칙기반 전문가 시스템 기초가 된 생성 시스템 모델 제안
    - 생성 모델은 인간이 자신의 지식을 적용해서 제시된 문제를 관련 정보로 해결한 다는 아이디어에 기반
    - 생성 규칙은 장가 기억장치에 저장되고, 특정 정보나 사실은 단기 기억장치에 저장
    - 규칙 기반 전문가 시스템의 모델



    - 규칙 기반 전문가 시스템의 기본 구조



- 규칙 기반 전문가 시스템의 필수요소
  - 기반지식
    - 문제 해결에 필요한 특정 분야에 관한 지식
    - 지식을 규칙 집합으로 표현
    - 각각의 규칙은 관계, 추천, 지시, 전략, 휴리스틱을 명시하고 IF THEN의 구조를 띔
    - 규칙의 조건 부분을 만족 하면, 규칙이 점화 되었다고 하며, 취해야 할 행동 부분을 실행
  - 데이터베이스
    - 데이터베이스는 기반지식에 저장된 규칙의 IF와 비교할 때 사용하는 사실들의 집합을 저장하고 있음.
  - 추론 엔진
    - 전문가 시스템이 해를 구할 수 있도록 추론 역할을 담당하며, 기반지식에 주어진 규칙들을 데이터베이스에 있는 사실과 연결
- 규칙 기반 전문가 시스템의 필수 요소
  - 해설 설비
    - 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
    - 문제의 답을 찾고 싶어 하는 사용자와 전문가 시스템 간의 통신 수단
    - 통신은 가능한 의미 있고, 사용하기 편리해야 함
- 규칙기반 전문가 시스템의 완전한 구조



- 전문가 시스템의 기본적인 특성
  - 좁고 전문화된 분야에서 전문가 수준으로 동작하도록 설계됨.
  - 빠르고 느리고의 문제가 아닌, 결과가 잘못되면 사용자는 만족하지 않음.
  - 하지만 문제 해결 속도는 매우 중요함, 환자가 죽거나 핵발전소가 폭발한 경우와 같이 매우 급한 상황에서 해를 구하는데 많은 시간이 든다면, 판단이나 진단이 정확해도 무용지물이 됨
  - 전문가는 문제를 정확하게 이해하고, 실제 경험을 사용하여 일반인 보다 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
- 해설 능력
  - 전문가 시스템의 독특한 특징으로 이를 통해 전문가 시스템은 자신의 추론을 재검토 하고, 결론을 설명함
  - 실제 전문가 시스템에서 해설이란 문제를 푸는 동안 점화된 규칙을 추적하는 것
  - 기반 지식에 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
  - 전문가 시스템은 문제를 풀 때 심벌 추론을 채택함
  - 개념, 규칙 같은 다른 종류의 지식을 표현하는데 심벌 사용
  - 수치 데이터를 처리하는 기존의 프로그램과 달리 전문가 시스템은 지식을 처리하며, 질적인 데이터를 쉽게 다룰 수 있음
- 전통적 프로그램 VS 전문가 시스템
  - 전통적인 프로그램은 알고리즘, 즉 잘 정의된 단계적 연산을 이용하여 데이터를 처리함
  - 알고리즘은 항상 같은 순서로 같은 연산을 수행하며, 늘 정확한 해를 제공하므로 전통적인 프로그램은 실수를 하지 않음
  - 전문가 시스템은 미리 기술된 단계의 순서를 따르지 않기 때문에 정확하지 않은 추론을 허용하고 부완전하고 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 수 있다는 것을 알면서도 여전히 전문가를 신뢰하는 것과 같이 대부분의 경우에는 전문가 시스템이 제공한 해를 신뢰할 수 있지만, 실수도 있다는 걸 인지해야함
- 전통적인 프로그램이 전문가 시스템보다 좋은점
  - 항상 같은 정확한 해를 제공한다.
  - 이에 비해서, 전통적인 프로그램은 데이터가 완전하고 정확할 때만 문제를 다룰 수 있다.
  - 데이터가 불완전하거나 약간의 에러를 포함하고 있으면, 전통적인 프로그램은 아무런 해도 제공하지 못할 뿐만 아니라 틀린해를 제공
  - 반면 전문가 시스템은 사용할 정보가 불완전하거나, 모호한 상황에서도 동작할 수 있으며, 여전히 합리적인 결론에 도달할 수 있음.
- 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 프로그램을이해하고 다시 살펴보기가 어려움.
        - 전문가 시스템에서는 지식이 처리 메커니즘과 명확하게 분리되어 있기 때문에전문가 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.
    - 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. IF-THEN생성 규칙으로 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 수있음
        - 지식과 과정의 분리(Separation of knowledge from its processing)
            - 규칙기반 전문가 시스템의 구조는 기반지식과 추론 엔진을 효율적으로 분리하여, 똑
              같은 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 범위내에서 기반지식에 규칙을 약간 추가하면 가능

- 불완전하고 불확실한 지식 다루기(Dealing with incomplete and uncertainknowledge)
  - 규칙기반 전문가 시스템은 불완전하고 불확실한 지식을 표현하고 추론

- 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.

- 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.

- 모든 문제에 규칙기반 전문가 시스템을 적용할 수 있는가
  - 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.