

인공지능 4분반

Homework 2



인공지능 4분반
소프트웨어학과
6조

권혁태, 김형균, 윤예원, 이장원, 정다훈

1 지식베이스가 다음과 같이 정의될때 주어진 Prolog의 실행 결과를 기술하시오.

Knowledge Base
likes(mary, food). likes(mary, wine). likes(john, wine). likes(john, mary).

P1.1. 주어진 query의 결과를 기술하시오.

Query
?- likes(mary, food). ?- likes(john, wine). ?- likes(john, food), likes(john, mary)

P1.2. 다음 query을 설계하고 실행 결과를 보이시오.

1. John likes anything that Mary likes.
2. John like anyone who likes wine.

P1.1.

?- likes(mary, food)

지식베이스에 **likes(mary, food).** 가 그대로 존재하기 때문에
결과는 true 가 나온다.

실행결과

2 **?- likes(mary, food).**
true.

?- likes(john, wine).

지식베이스에 **likes(john, wine).** 도 그대로 존재하기 때문에
결과는 true 가 나온다.

실행결과

```
3 ?- likes(john, wine).  
true.
```

?- likes(john, food), likes(john, mary)

두 개의 목표가 AND로 묶인 쿼리이므로 다음과 같은 과정을 거친다..

1. `likes(john, food)` 을 먼저 찾는다.

지식베이스에 그런 사실이 없음 → 실패.

2. AND(кома) 왼쪽이 실패하면 오른쪽 `likes(john, mary)`가 참이어도
전체 쿼리는 실패한다. 따라서 결과는 **false** 가 나온다.

실행결과

```
4 ?- likes(john, food), likes(john, mary).  
false.
```

P1.2

1) John likes anything that Mary likes.

Mary가 좋아하는 대상 X에 대해 John도 그 X를 좋아하는지를 확인해야 하므로 다음 질의를 사용한다.

```
?- likes(mary, X), likes(john, X).
```

`likes(mary, X)`로부터 X는 food, wine 두 가지가 나오며, 각각에 대해 `likes(john, X)`를 확인한다.

food는 John이 좋아하지 않으므로 실패하고, wine은 John이 좋아하므로 성공한다.

결과:

```
X = wine.
```

실행결과

```
5 ?- likes(mary, X), likes(john, X).  
X = wine.
```

2) John likes anyone who likes wine.

와인을 좋아하는 사람 Y를 찾고, John이 그 사람 Y를 좋아하는지를 확인해야 하므로 다음 질의를 사용한다.

```
?- likes(Y, wine), likes(john, Y).
```

`likes(Y, wine)`로부터 Y = mary, john 두 값이 나오며, 각각에 대해 `likes(john, Y)`를 확인한다.

john은 John이 좋아하지 않으므로 실패하고, mary는 John이 좋아하므로 성공한다.

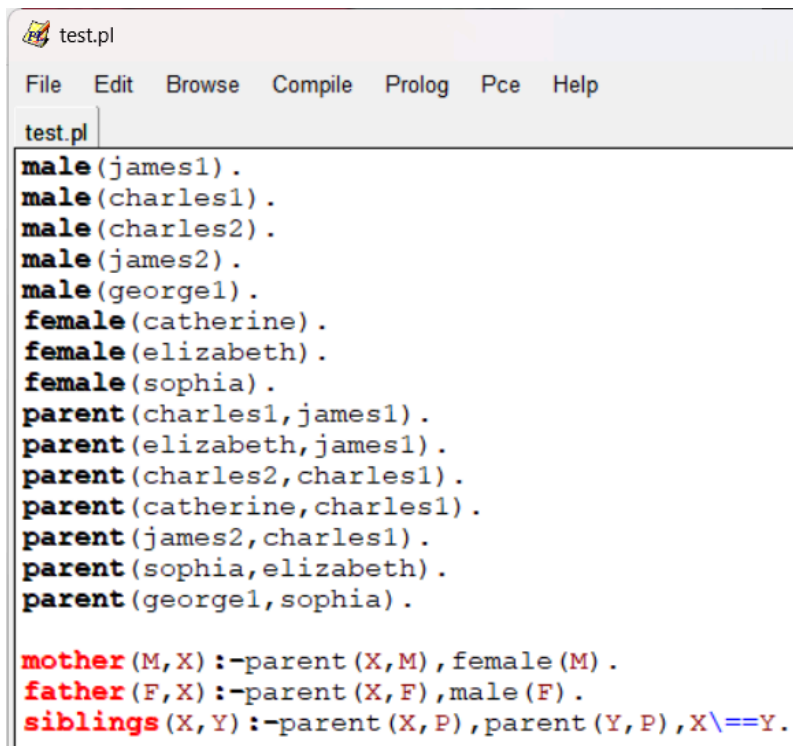
결과:

```
Y = mary.
```

실행결과

```
6 ?- likes(Y, wine), likes(john, Y).  
Y = mary
```

2. 주어진 가계도 (family tree)로부터 지식베이스가 다음과 같이 정의되었다.
주어진 query을 설계하고 실행하시오.



```
test.pl
File Edit Browse Compile Prolog Pce Help
test.pl
male(james1).
male(charles1).
male(charles2).
male(james2).
male(george1).
female(catherine).
female(elizabeth).
female(sophia).
parent(charles1,james1).
parent(elizabeth,james1).
parent(charles2,charles1).
parent(catherine,charles1).
parent(james2,charles1).
parent(sophia,elizabeth).
parent(george1,sophia).

mother(M,X):-parent(X,M),female(M).
father(F,X):-parent(X,F),male(F).
siblings(X,Y):-parent(X,P),parent(Y,P),X\==Y.
```

P2. 다음 query를 설계하고 실행하시오.

1. Was George I the parent of Charles I?
2. Who was Charles I's parent?
3. Who were the children of Charles I?

[실행 결과]

```
?- parent(george1,charles1).  
false.
```

```
?- parent(charles1,X).  
X = james1.
```

```
?- parent(X,charles1).  
X = charles2 ;  
X = catherine ;  
X = james2.
```

[설명]

- 1) george1이 charles1의 부모였는지 묻는 쿼리는 “parent(george1, charles)”이 지식베이스에 존재하는지 확인하는 것이고 없으므로 **false**가 출력된다.
- 2) charles1의 부모를 찾는 쿼리는 “parent(charles1, X)”가 참인 X를 찾는 것으로 지식베이스에 “parent(charles1, james1)”가 있으므로 “X = james1”가 출력된다.
- 3) charles1의 자식을 찾는 쿼리는 charles1이 부모인 경우를 찾는 쿼리와 동일해 “parent(X, charles1)”을 수행하면 지식베이스에 “parent(charles2, charles1)”, “parent(catherine, charles1)”, “parent(james2, charles1)”가 있으므로 “X = charles2, catherine, james2”이 출력된다.

P3. 다음 rule을 설계하고 검증하시오.

1. M is the mother of X if she is a parent of X and is female.
2. F is the father of X if he is a parent of X and is male.
3. X is a sibling of Y if they have the same parent.

[실행 결과]

```
?- mother(M,sophia).  
M = elizabeth.  
  
?- father(F,charles1).  
F = james1.  
  
?- siblings(charles2,catherine).  
true
```

[설명]

- 1) “mother(M,X) :- parent(X,M), female(M).”
M이 X의 부모인 경우 “parent(X,M)”과 M이 여성인 경우 “female(M)”을 AND 연산자로 연결하고 참일 경우 “mother(M,X)”로 규칙을 표현하였고, “mother(M,sophia)”는 sophia의 엄마가 누구인지 묻는 쿼리로 지식베이스에 “parent(sophia,elizabeth)”가 있고 “female(elizabeth)”가 있으므로 “M = elizabeth”가 출력된다.
- 2) “father(F,X) :- parent(X,F), male(F).”
F가 X의 부모인 경우 “parent(X,F)”와 F가 남성인 경우 “male(F)”을 AND 연산자로 연결하고 참일 경우 “father(F,X)”로 규칙을 표현하였고, “father(F,charles1)”는 charles1의 아버지가 누구인지 묻는 쿼리로 지식베이스에 “parent(charles1,james1)”가 있고 “male(james1)”가 있으므로 “F = james1”이 출력된다.
- 3) “siblings(X,Y) :- parent(X,P), parent(Y,P), X\==Y.”
X와 Y의 부모가 같을 경우를 AND 연산자를 사용하여 “parent(X,P), parent(Y,P)”로 표현하였고, X와 Y가 서로 달라야 하므로 “X\==Y”를 부모가 같을 경우와 AND 연산자로 연결하여 참일 경우 “siblings(X,Y)”로 규칙을 표현하였다.
“siblings(charles2, catherine)”는 charles2와 catherine이 형제자매인지 묻는 쿼리로 지식베이스에 “parent(charles2, charles1)”, “parent(catherine, charles1)”가 있고 “charles2 \== catherine”이므로 true가 출력된다.

3

활용 목적에 따른 지식의 표현 방법으로 의미망 (semantic network), 프레임 (frame), 술어 논리 (predicate logic) 등 방법들이 사용된다. 주어진 문제에 답하시오.

3.1. 의미망 (semantic network), 프레임 (frame), 술어 논리 (predicate logic) 등을 표현 방법, 입력 방법, 응용 사례 등 관점에서 비교하시오.

3.2. 다른 형태의 지식 표현 방법의 활용 사례를 찾아서 기술하시오.

3.3. 보편적으로 선택되는 관계형데이터베이스와 술어논리의 지식 표현을 비교하시오.

3.1) 지식 표현 방식에는 의미망(semantic network), 프레임(frame), 술어 논리(predicate logic)가 대표적으로 사용됩니다. 의미망은 개념과 개념의 관계를 그래프 구조로 시각적으로 연결하여 표현하는 방식으로, 노드에는 객체나 개념이, 엣지에는 두 객체 간의 관계를 입력합니다. 이 방식은 관계성과 계층 구조를 직관적으로 드러내기 쉽고, 시맨틱 웹이나 개념 지도와 같은 분야에서 널리 활용되지만, 아주 복잡한 규칙이나 방대한 데이터 처리에는 한계가 있습니다. 프레임은 의미망에서 발전된 형태로, 객체 혹은 개념 하나를 ‘프레임’으로 표현하며 그 속성을 ‘슬롯’이라는 구조로 저장합니다. 입력 시 각 프레임에 다양한 속성과 값을 할당할 수 있어, 복잡한 대상이나 상황을 체계적으로 관리하는 데 강점을 보입니다. 전문가 시스템이나 자연어 처리에서 객체와 상황에 대한 세부 정보를 다루는 데 적합하지만, 관계 표현이나 규칙 추론에는 제한이 있습니다. 술어 논리는 논리적 진술과 규칙을 이용해 사실과 관계를 공식적으로 입력하는 방식으로, Prolog 등에서 “parent(철수, 영희)”처럼 직접 명시적으로 입력합니다. 복잡한 규칙을 정의하고, 논리적 관계를 바탕으로 새로운 사실을 추론하는 데 탁월하여 추론 엔진이나 질의응답 시스템에서 활용됩니다. 다만, 데이터가 방대해질 경우 관리 효율이 떨어지며, 초보자에게는 상대적으로 이해와 활용에 어려움이 있을 수 있습니다.

3.2) 온톨로지 (Ontology)는 존재하는 사물과 사물 간의 관계 및 여러 개념을 컴퓨터가 처리할 수 있는 형태로 표현하는 것으로 인공 지능(AI), 시맨틱 웹(semantic Web), 자연어 처리(NLP: Natural Language Processing), 문헌정보학 등 여러 분야에서 지식 처리, 공유, 재사용 등에 활용됩니다.[1]

인공지능(AI) 전문기업 솔트룩스가 지능형 검색만으로 3D 가상공간을 손쉽게 제작하고 편집할 수 있는 혁신적인 AI 기술을 공개했다고 밝혔습니다. 자연어 기반 메타공간 창작을 위한 LLM과 온톨로지 기술을 접목한 뉴로 심볼릭 AI 기술이 적용된 것이 특징입니다. 이를 활용해 단순 텍스트 검색으로는 찾기 어려운 연관 정보를 온톨로지 형태로 구조화하였습니다.[2]

은닉 마르코프 모델 (HMM)은 상태 정보가 은닉된 통계적 마르코프 모델로 출력된 정보만 관찰되어 출력된 정보만을 가지고 숨겨진 상태(state) 정보를 추정하는 것으로 음성인식, 자연어 처리, 몸짓 인식(gesture recognition) 등과 같이 대량으로 출력된 데이터를 통계적으로 패턴 분석하여 입력된 정보를 추론하는 데에 응용됩니다.[3]

앤드류 D. 베일리 책임연구원이 이끄는 지능화장비 연구팀이 개발한 기술은 PECVD 공정에서 발생한 불량 원인을 AI로 분석하는 기술로 '은닉 마르코프 모델(HMM)'을 이용하여 현상 변화를 확률로 표현합니다. 불량 시 발생하는 원인을 분석하고, 관련 데이터를 찾아내주는 기술입니다.[4]

참고 자료

- [1] 한국통신기술협회, "온톨로지", TTA정보통신용어사전. [온라인]. Available: http://terms.tta.or.kr/dictionary/dictionaryView.do?word_seq=050917-5 (접속일 : 2025-11-18)
- [2] 이수민, "솔트룩스, LLM과 온톨로지 결합한 AI 기술 공개... 말 한마디로 3D 공간이 똑딱", 뉴스로드. 2025-11-18. [온라인]. Available: <https://www.newsroad.co.kr/news/articleView.html?idxno=49140> (접속일 2025-11-18)
- [3] 한국통신기술협회, "은닉 마르코프 모델", TTA정보통신용어사전. [온라인]. Available: http://terms.tta.or.kr/dictionary/dictionaryView.do?word_seq=044949-4 (접속일 : 2025-11-18)
- [4] 박설민, "반도체·디스플레이, 중국 추격 따돌릴 열쇠는 '인공지능'", 시사위크. 2023-08-30. [온라인] Available: <https://www.sisaweek.com/news/articleView.html?idxno=207593> (접속일 : 2025-11-18)

3.3) 관계형데이터베이스와 술어논리의 지식표현은 지식을 디지털데이터로 변환하고, 쿼리를 통해서 정보를 추출한다는 점에서는 공통점을 갖습니다. 하지만 존재하는 지식으로부터 새로운 정보를 도출해내는 부분에서는 차이점을 갖습니다.

더 자세하게 살펴보자면, 두 방식은 모두 논리적인 관계를 기반으로 사실을 표현합니다. RDB에서 **table**, **row**는 술어논리에서 술어, 사실로 대응됩니다. 추가적으로 두 시스템 DB나 지식 베이스에 존재하지 않는 사실은 거짓으로 간주되기 때문에 모두 폐쇄 세계 가정을 따릅니다. 예를 들어 RDB의 **Parent** 테이블에 철수 영희라는 행이 있는 것은 술어 논리의 **parent(철수, 영희)**라는 사실과 대응되며 만약 철수 길동이라는 데이터가 없다면 두 시스템 모두 철수의 자식이 길동이라는 사실은 없다고 판단합니다.

하지만, 지식의 확장성과 추론 능력에 있어서는 차이점이 존재합니다. RDB는 **table**에 구체적으로 저장된 값을 **query**를 이용해서 검색하거나, 분류하는 것에 초점이 맞추어져 있습니다. 따라서 RDB에선 서로 다른 테이블에 이미 존재하는 사실들을 **join** 하여 사용자가 입력한 대로 출력할 수는 있지만, 그것에 바탕해서 새로운 결과를 추론하는 것에는 명확한 한계가 존재합니다. 예를 들어, **Parent** 테이블에 할아버지와 아버지 그리고 아버지와 나 관계가 각각 저장되어 있을 때, RDB에서 할아버지가 나의 조상인가를 확인하려면 사용자가 직접 두 관계를 연결하는 쿼리를 명시적으로 작성해야 합니다. DB 자체는 두 사람이 가족 관계라는 사실을 스스로 인지하지 못합니다.

반면에 술어 논리에선 데이터 간의 관계를 설정하는 규칙을 표현하기 때문에, 단순히 사실을 저장하는 것을 넘어서 사실들의 논리적 관계를 통해서 새로운 사실을 추론할 수 있습니다. 예를 들어, 술어논리에서 부모의 부모는 조부모다라는 규칙을 정의할 수 있습니다. 이를 통해 지식 베이스에 할아버지와 나라는 직접적인 사실이 저장되어 있지 않더라도 추론 엔진이 자동으로 중간 관계를 찾아내어 조부모라는 새로운 사실을 도출해낼 수 있습니다.

이처럼 술어 논리는 RDB와는 달리, 일반적인 규칙을 설정할 수 있고, 명시적으로 저장되지 않은 사실에 대해서 기존의 사실을 결합하여 논리적으로 추론할 수 있다는 차이점이 존재합니다.

따라서, RDB는 대용량의 정형화된 데이터를 효율적으로 관리할 수 있는 저장소의 역할에 특화되어있고, 술어 논리는 사실과 논리 규칙을 결합해서 지식 베이스에 없는 새로운 결과를 추론하는 도구의 역할에 특화되어있다고 생각합니다.