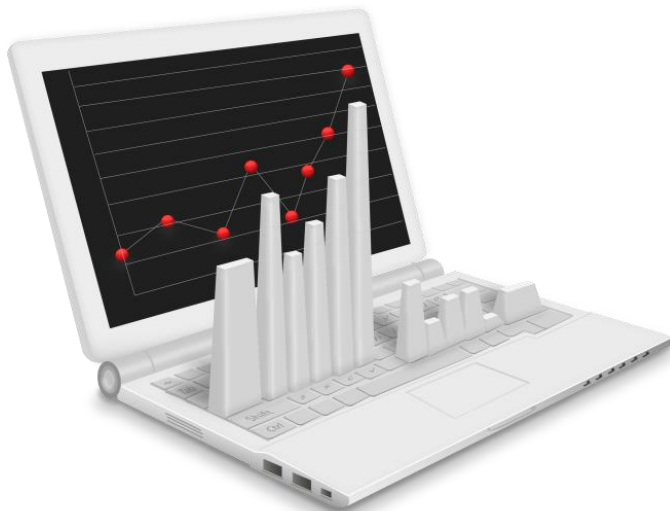


# 프로그래밍 언어론

논리형 언어의 개념

컴퓨터공학과

조은선



## 논리형언어의 개념

### 학 습 목 표

- 대표적인 논리형 프로그래밍 언어인 Prolog의 기본적인 개념과 구문을 공부한다.

### 학 습 내 용

- 논리형 프로그래밍언어인 prolog의 기본 개념
- "fact", "rule", "query" 등 프로그램의 요소
- 자료구조를 나타내는 "term"의 종류
- 연산자의 사용 방법



# 목 차

- 들어가기
- 학습하기
  - 논리형 프로그램의 개요
  - 논리형 프로그램의 요소
    - : fact, rule, query
  - 논리형 프로그램의 자료구조
    - : term
  - 연산자 사용
- 평가하기
- 정리하기



# 알고가기



다음 중 잘못된 추론을 고르시오.

1) 학생은 공부한다.

    뽀로로는 학생이다.

    그러므로 뽀로로는 공부한다

2) 똑똑한 사람들은 홍삼을 먹는다.

    나는 홍삼을 먹는다.

    그러므로 나는 똑똑하다.

3) 아이폰은 혁신적이다.

    혁신적인 것들은 디자인이 안 좋다.

    그러므로 아이폰은 디자인이 안 좋다.

4) 프로그래밍언어론은 재밌다.

    그러므로 프로그래밍언어론은 재밌다.

확인



# | 논리형 프로그래밍 언어의 개요

논리(logic)에 기반을 둔 독특한 프로그래밍 언어

논리 : 명제, 논리곱( $\cap$ ), 논리합( $\cup$ ), 부정( $\neg$ ), 함축( $\rightarrow$  또는  $\supset$ ) ...

특히 논리적 추론을 사용 : “학생은 공부한다. 나는 학생이다. 그러므로 나는 공부한다.”

Prolog(프롤로그)가 가장 널리 사용되고 유명함

“논리형 변수(logical variable)”라는 것을 사용

→ 어느 프로그래밍 언어의 “변수”와는 다른 의미를 가짐

“어떻게” 보다는 “왜” 중심

→ 무엇이 맞는 지만 기술하면 그 해를 어떻게 구하는지는 프로그래머가 상관하지 않는다.



# | 논리형 프로그램의 예

father(태종, 세종) .

father(태조, 태종) .

Fact

parent(X,Y) :- mother(X,Y) .

parent(X,Y) :- father(X,Y) .

grandparent(X,Z) :-

parent(X,Y) , parent(Y,Z) .

Rule

?-father(태종, 세종)

yes

?-grandparent(X, 세종)

태조

Query

# | 논리형 프로그램의 요소

## Fact (사실)

→ 사실을 명시함

예 `father(태종, 세종) .`

## Rule (규칙)

→ head(머리)와 body(몸체)로 구성됨

`head :- body`

→ 의미 : “body가 참이라면 head도 참이다”

예 `grandparent(X, Z) :-  
parent(X, Y), parent(Y, Z) .`

## Query (질의)

→ 답을 얻고자 사용하며, goal (목표) 라고도 불림

예 `?-grandparent(X, 세종)`



# | 자료구조

## Term(항)

→ Fact, rule, query를 구성하는 데에 사용되는 보다 기본적인 요소

## 종류

→ Atom (원자항) : 태종, 세종, mary, dog, 'Lamb', []

→ 설명이 필요없는 기본적인 이름

→ 소문자로 시작하거나 따옴표가 있음.

→ []는 empty list를 나타냄

→ Numbers (상수항) : 57, 3.14

→ Variable Term(변수항) : x, Name, \_

→ \_ 는 아무거나를 나타냄

→ Compound Term(복합항)

→ List

→ String





# | Compound Term (복합항)

## Compound Term(복합항)

- Fact와 rule의 head와 body, query 등을 이름
- 함수 이름을 나타내는 functor부분과 argument들로 구성된다.

### → 앞서 예에서

```
father(태종, 세종) .
```

- father 는 functor(함수자), 태종, 세종 은 argument(인자) 임
- functor 이름은 atom 임
- 인자 개수는 arity라고 함
- atom은 arity 0인 compound term이기도 하다
- 각 인자는 또다시 term 이며 compound term도 가능하다.



# | List (리스트)

## List (리스트)

예 `[a, b, c]`

➔ head와 tail로 나타낼 수도 있음 (함수 언어 참고)

예 `(a, (b, (c, [])))`  
또는 `[a|[b|[c| [] ] ] ]`



# | Operator(연산자)

연산자도 compound term으로 나타냄

(+, =/= 등은 atom 이 되는 functor 이름)

예

`- (+ (15, 17), t)`

편리한 표현(syntactic sugar)도 제공

→ 위의 예는 다음과 같이 표현으로 사용 가능함

예

`(15+17) -t`



# 평가하기

마지막으로 내가 얼마나 이해했는지를 한번 확인해 볼까요?  
총 2문제가 있습니다.

START



# 평가하기 1

## 1. 다음중 프롤로그에 대한 설명으로 맞지 않는 것은?

- ① 논리에 기반을 둔 논리형 프로그래밍 언어이다.
- ② 프로그램의 결과를 얻을 때는 query를 사용한다.
- ③ rule은 head와 body로 이루어지며, head가 참이면 body도 참이라는 뜻이다.
- ④ Compound term은 함수 이름을 나타내는 functor와 argument들로 구성된다.

확인



## 평가하기 2

### 2. 다음 Prolog 문장에 대한 설명으로 틀린 것은?

```
elephant(mary) .  
    hasLong_nose(X) :- elephant(X)  
    hasBig_foot(X)  :- elephant(X)  
    ?- hasBig_foot(mary)
```

- ① 첫번째 문장은 mary가 elephant 임이 사실이라는 것을 의미한다.
- ② 두번째와 세번째 문장은 rule로서 elephant(X) 부분이 body 이다.
- ③ 두번째 문장은 X가 elephant 이면 X는 hasLong\_nose 임을 의미한다.
- ④ 마지막 문장을 입력하면 나오는 결과는 mary이다.

확인



# 정리하기

- ➡ 논리형 언어는 논리와 추론에 기반을 두고 있다.
- ➡ 논리형 프로그래밍언어 중 가장 많이 사용되는 것은 prolog이다.
- ➡ 논리형프로그램은 “fact”, “rule”, “query” 등 프로그램의 요소를 가진다.
- ➡ “term”은 자료구조이며 atom, number term, variable term, compound term, list, 등이 있다.
- ➡ 연산자는 term으로 표현되나 편리한 표기방법도 함께 제공된다.



“ 강의를 마치겠습니다. 수고하셨습니다. ”

