



# I. 명제 논리 命題 論理

## Propositional Logic

CSE1007 논리학

Logical Foundations of Programming

# 1. 핵심 문장 논리

The Logic of Atomic Sentences



## 핵심 문장의 논리

- 1차 논리(First-Order Logic: FOL) 언어를 배우고 익힌다.
- 이미 알고 있는 사실로부터 어떻게 논리적 결과를 이끌어내는지 개념과 방법을 이해한다.
- 그러기 위해서 일단 핵심 문장 (atomic sentences)의 논리부터 알아보자.

## 논리 Logic

- 논리란? – 이치<sub>理致</sub>에 맞는지를 따지는 언어
- Syntax = 구문<sub>構文</sub> = 생김새 = 모양
  - 문장이 어떻게 생겼는지를 정의
  - 일단 문장이 제대로 생겨야 의미를 부여할 수 있다.
- Semantics = 의미<sub>意味</sub>
  - 해석 interpretation
  - 문장이 맞다(true)/틀리다(false)를 판별

## 명사 Individual Constants

- 실제로 존재하는 사람/물체를 가리키는 이름

자연어	박지성	한양대학교 에리카캠퍼스	일곱
FOL의 명사 모두 소문자로 표현	park	hyu-erica	7

- 제약
  - 여러 사람/물체에 한 이름을 붙일 수 없다.
    - » 박지성? 박찬호? 박세리? → park
  - 한 사람/물체에 이름을 둘 이상 붙이는 건 가능하다.
    - » 조용우 → cho, professor-cho, doctor-cho
  - 이름이 없어도 된다.

## 명사구 Terms

- 단순 명사구 Simple terms
  - 명사
- 복합 명사구 Complex terms
  - 1개의 이상의 (단순/복합) 명사구를 인수argument로 갖고 있는 함수관계기호function symbol로 만들
  - 즉, 함수관계기호(명사구)의 형태

- 예:

자연어	호돌이 엄마	호돌이 엄마의 엄마
FOL의 명사구 모두 소문자	<i>mother(hodori)</i>	<i>mother(mother(hodori))</i>

- 복합 명사구도 명사와 같은 제약
- 즉, 반드시 하나의 사람/물건을 가리켜야 한다.

## 술어 Predicates

- 사람/물건의 **특징** 또는 사람/물건 사이의 **관계**를 표현
- 동사 또는 형용사에 해당한다고 보면 됨

자연어	박지성이 <u>집에 있다</u> .	돌이는 순이의 여동생을 <u>좋아한다</u> .
FOL의 술어 대문자로 시작	Home	Likes

## 핵심 문장 Atomic Sentences

- 핵심 문장은 술어가 명사구들을 인수로 가진 형태로 만든다.
- 즉, 술어(명사구,...,명사구)의 형태

자연어	돌이가 집에 있다.	돌이는 순이의 여동생을 좋아한다.
FOL	Home(dori)	Likes(dori,sister(suni))

- 술어에 달려있는 인수arguments의 개수(=arity)는 **고정**

핵 문장	술어	명사구 인수	인수개수
Home(dori)	Home	dori	1 (unary)
Likes(dori,sister(suni))	Likes	dori, sister(suni)	2 (binary)
Sent(dori,suni,email)	Sent	dori, suni, email	3 (ternary)

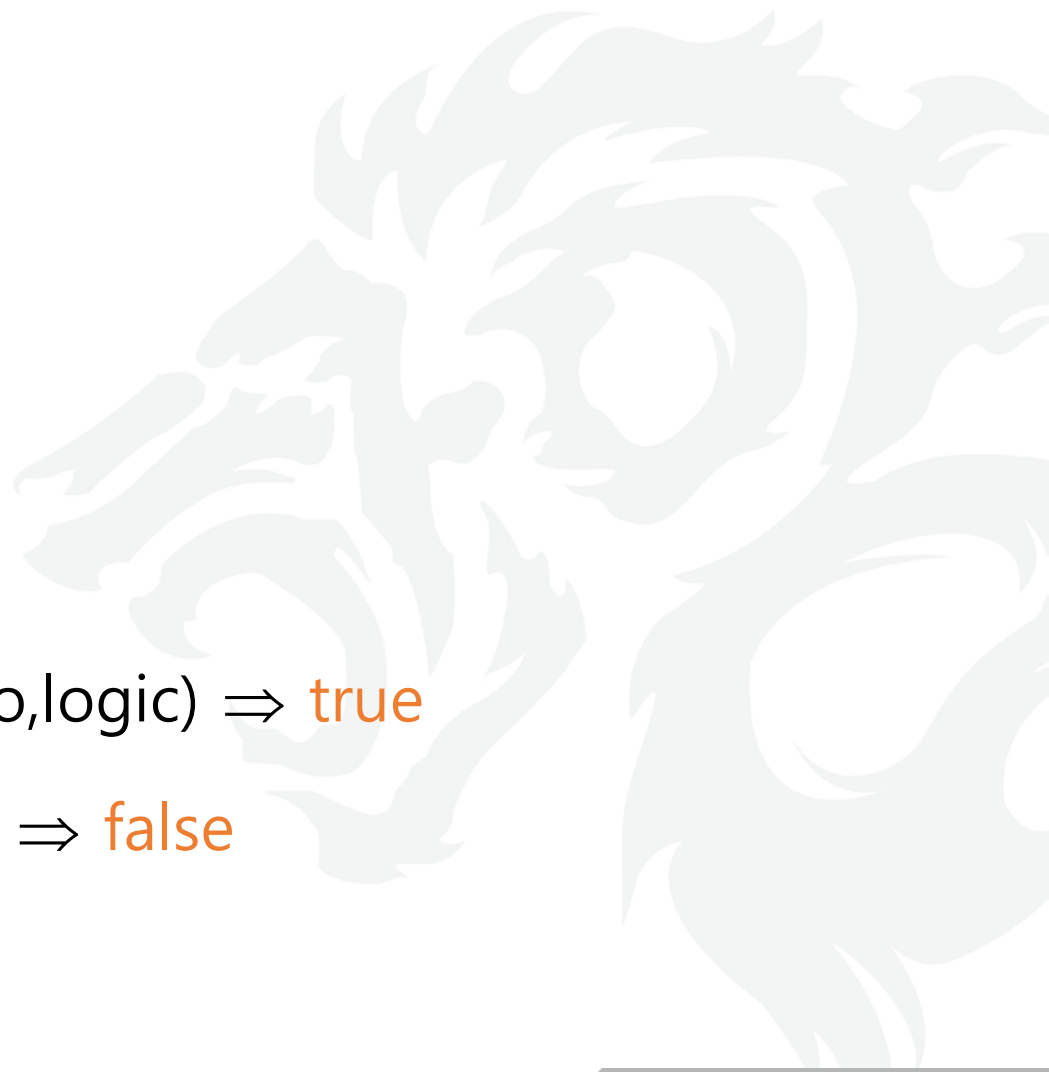


## 핵심 문장의 중위/전위 표기법

- 전위 표기법 prefix notation
  - 술어 이름이 먼저 오고, 괄호로 둘러싸인 명사 인수가 다음에 온다.
  - 예: Likes(dori,suni)
- 중위 표기법 infix notation
  - 술어 이름이 인수 사이에 낀다.
  - 예:  $a = b$

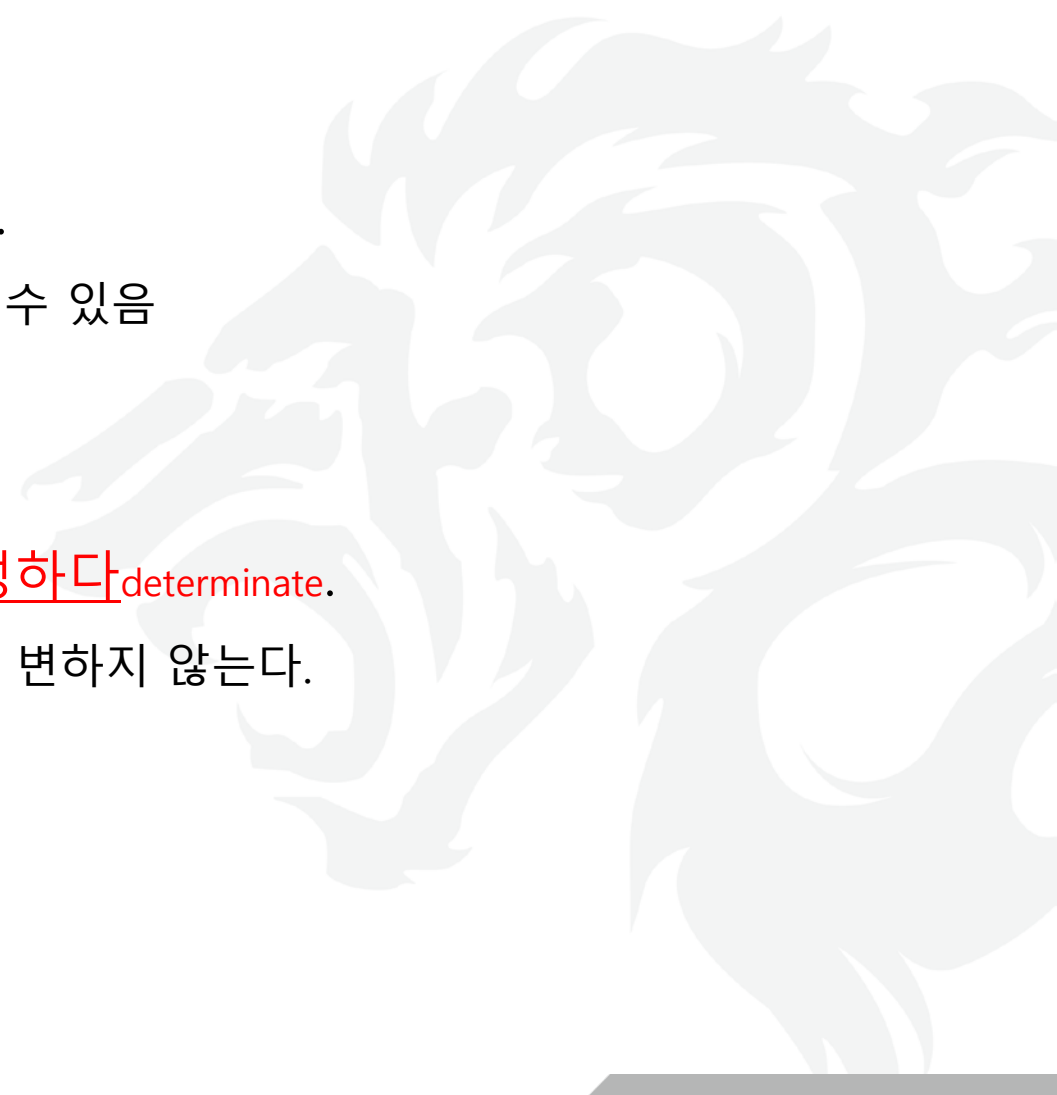
## 핵심 문장의 의미

- 참 또는 거짓
- true 또는 false
- 예:
  - Teaches(professor-cho,logic)  $\Rightarrow$  true
  - Teaches(hodori,logic)  $\Rightarrow$  false



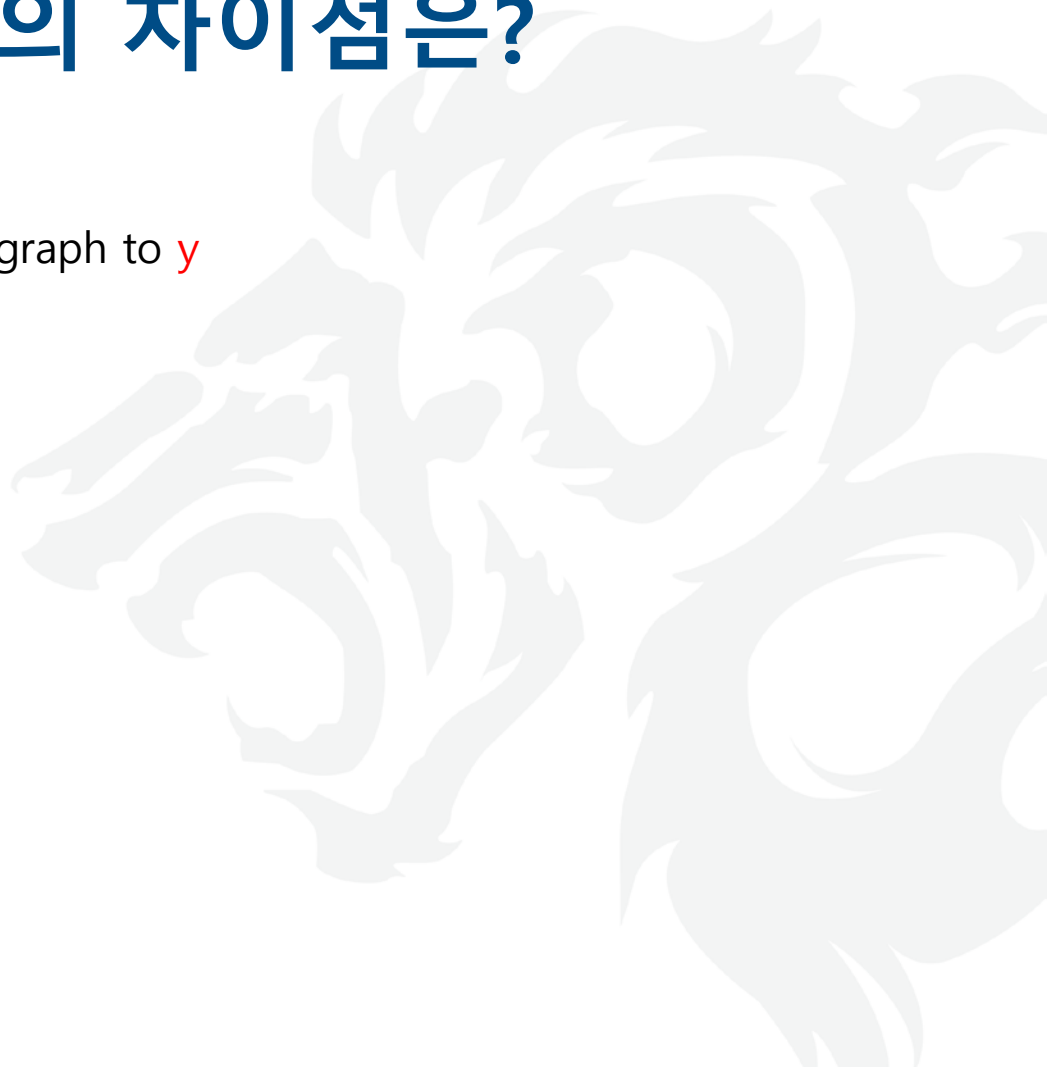
## 핵심 문장의 특징

- 명사 인수의 순서는 중요하다.
  - 순서가 다르면 의미가 달라질 수 있음
  - Likes(dori,suni)
  - Likes(suni,dori)
- 핵심 문장의 의미는 항상 일정하다 determinate.
  - 언제나 같은 의미이다. 의미는 변하지 않는다.
  - Home(dori)
  - Young(suni)
  - Loves(dori,suni)



## 아래 세 핵심 문장의 차이점은?

- 구문: ParkjsGaveAutographTo(y)
- 의미: Park Ji Seong gave an autograph to y
  - ParkjsGaveAutographTo(suni)
- 구문: GaveAutograph(x,y)
- 의미: x gave an autograph to y
  - GaveAutograph(parkjs,suni)
- 구문: Gave(x,y,z)
- 의미: x gave z to y
  - Gave(parkjs,suni,autograph)



## Tarski's World 하고 놀자!

- Tarski's World와 노는 방법
  - 참조: LPL Software Manual  
Chapter 3 Using Tarski's World.
- 숙제: Tarski's World와 친해지기
- Tarski's World에 들어가서 핵심 문장의 의미를 해석하면서 놀아보자.
  - Wittgenstein's World 파일을 열고 놀아보자.
  - Wittgenstein's Sentences 파일을 열고 놀아보자.

## FOL 언어의 예 1: 집합론set theory의 FOL 언어

- 술어

- $=$

- » 집합이 같은지 비교하는 술어 (set equality)
- » 중위infix, 이항binary 연산자
- » 핵심 문장의 예:  $\text{ansan} = \text{haengdang}$ ,  $\text{hanyang} = \text{hanyang}$

- $\in$

- » 원소가 집합의 소속인지를 알려주는 술어 (set membership)
- » 중위Infix, 이항binary 연산자
- » 핵심 문장의 예: a는 1의 이름, b는 {1,3,5}의 이름이라고 하면, 다음의 의미는?
  - :  $a \in a$
  - :  $a \in b$
  - :  $b \in a$
  - :  $b \in b$

## FOL 언어의 예 2: 산수의 FOL 언어

- 구문<sub>syntax</sub>
  - 명사: 0, 1
  - 술어: =, < (중위, 이항)
  - 함수관계기호: +, × (중위, 이항)
- 명사식
  - 고찰: 명사구를 모두 몇 개 만들 수 있을까?
  - 정의\*: (명사구<sub>term</sub>)    \*귀납적정의<sub>inductive definition</sub>
    - 1) 명사 0과 1은 명사구이다.
    - 2)  $t_1$ 과  $t_2$ 가 명사구이면,  $(t_1+t_2)$ 와  $(t_1\times t_2)$ 도 명사구이다.
    - 3) 1)과 2)번 규칙을 반복해서 적용해서 얻을 수 없으면 명사구가 아니다.
- 명사구의 예:  $((1+1)\times(1+1))$

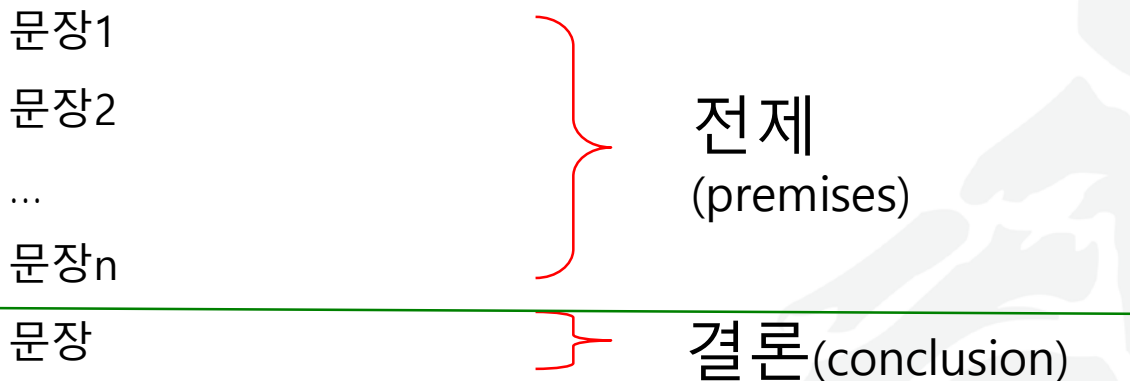
# FOL 언어의 예 2: 산수의 FOL 언어 (계속)

- 핵심 문장의 예
  - $(1 \times 1) < (1 + 1)$
  - $(1 \times 1) = (1 + 0)$





## 주장 Arguments



- **주장의 예:** 모든 사람은 언젠간 죽는다. 소크라테스는 사람이다. 따라서 소크라테스도 언젠간 죽는다.

**전제1:** 모든 사람은 언젠가 죽는다.

**전제2:** 소크라테스는 사람이다.

**결론:** 소크라테스는 언젠가 죽는다.

# 논리적으로 타당한 주장

## Logically Valid Arguments

- 전제<sub>premise</sub>가 참<sub>true</sub>이라고 가정하고,  
논리적으로 결론<sub>conclusion</sub>이 참<sub>true</sub>이 되는 주장
- 결론은 전제의 논리적 결과(logical consequence)

## 다음 주장은 논리적으로 타당한가?

- **주장1:** 사람은 모두 언젠가 죽는다. 소크라테스는 사람이다. 따라서 소크라테스도 언젠가 죽는다.
  - **전제1:** 사람은 모두 언젠가 죽는다.
  - **전제2:** 소크라테스는 사람이다.
  - **결론:** 소크라테스는 언젠가 죽는다.
- **주장2:** 미남미녀는 모두 언젠가 죽는다. 한양대학교 학생과 교수는 모두 미남미녀이다.  
따라서 한양대학교 학생과 교수는 언젠가 죽는다.
  - **전제1:** 미남미녀는 모두 언젠가 죽는다.
  - **전제2:** 한양대학교 학생과 교수는 모두 미남미녀이다.
  - **결론:** 한양대학교 학생과 교수는 언젠가 죽는다.

논리적으로 타당하다(valid)

논리적으로 타당하다(valid)

## 다음 주장은 논리적으로 타당한가?

- **주장3**: 한양대학교 학생과 교수는 모두 미남미녀이다. 왜냐하면, 미남미녀는 모두 언젠가 죽고 한양대학교 학생과 교수도 언젠가 죽을 것이기 때문이다.
  - **전제1**: 미남미녀는 모두 언젠가 죽는다.
  - **전제2**: 한양대학교 학생과 교수는 모두 언젠가 죽는다.
  - **결론**: 한양대학교 학생과 교수는 모두 미남이다.

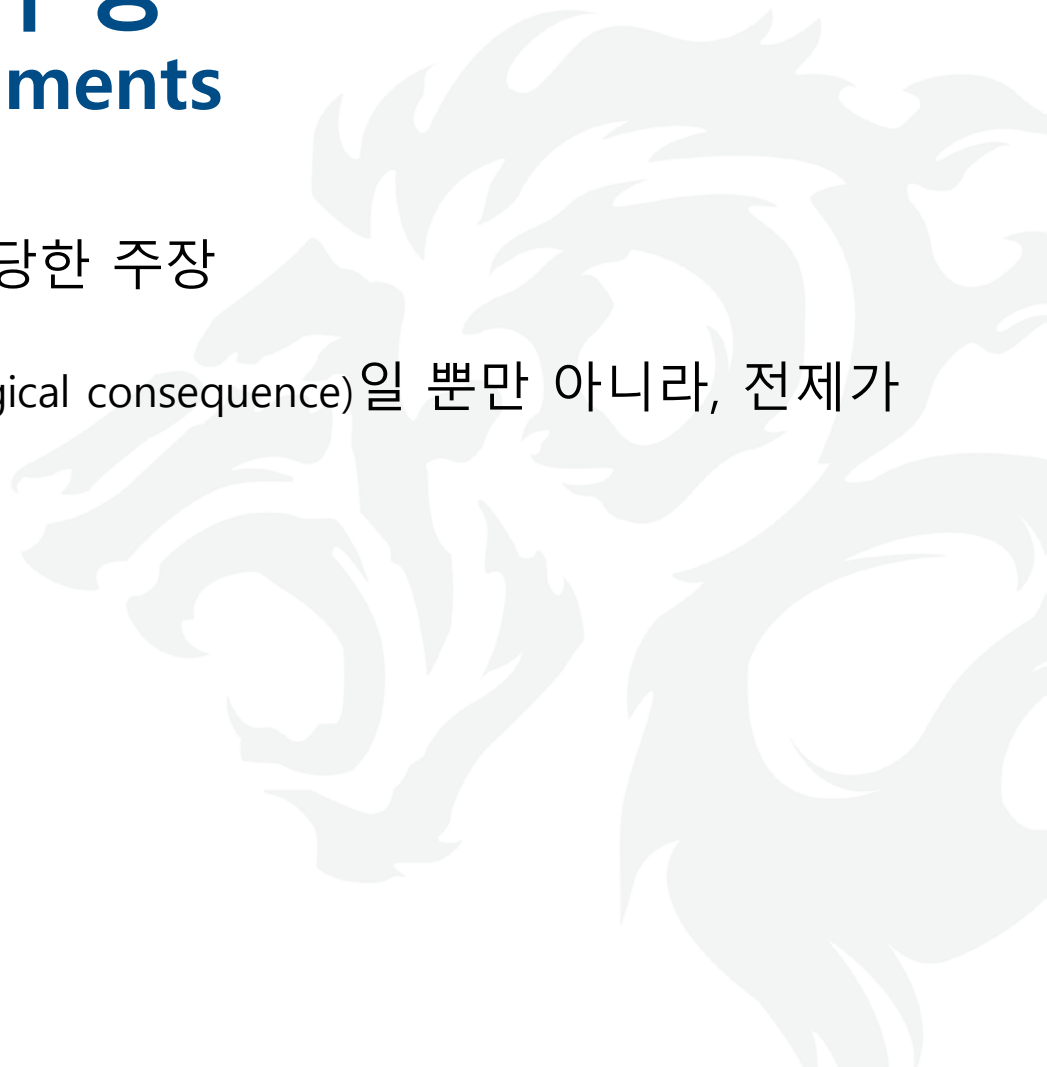
논리적으로 타당하지 않다(invalid)

## 다음 주장은 논리적으로 타당한가?

- **주장3**: 한양대학교 학생과 교수는 모두 미남미녀이다. 왜냐하면, 미남미녀는 모두 언젠가 죽고 한양대학교 학생과 교수도 언젠가 죽을 것이기 때문이다.
  - **전제1**: 미남미녀는 모두 언젠가 죽는다.
  - **전제2**: 한양대학교 학생과 교수는 모두 언젠가 죽는다.
  - **결론**: 한양대학교 학생과 교수는 모두 미남이다. **논리적으로 타당하지 않다(invalid)**
- 타당하지 않음을 어떻게 증명하나?
  - 반례를 찾으면 됨
  - 반례 = 전제가 모두 참인데도 불구하고 결론이 거짓이 되는 예
  - 주장3의 반례
    - » 미남미녀는 모두 언젠가 죽고, 17학번 "안미남"도 언젠가 죽음에도 불구하고, "안미남"은 미남이 아니라고 하네요. ㅎㅎ

# 논리적으로 맞는 주장 Logically Sound Arguments

- 전제가 참이고 논리적으로 타당한 주장
- 결론은 전제의 논리적 결과(logical consequence)일 뿐만 아니라, 전제가 모두 참이다.



## 다음 주장은 논리적으로 맞는가?

- **주장1:** 사람은 모두 언젠가 죽는다. 소크라테스는 사람이다.  
따라서 소크라테스도 언젠가 죽는다.
  - **전제1:** 사람은 모두 언젠가 죽는다.
  - **전제2:** 소크라테스는 사람이다.
  - **결론:** 소크라테스는 언젠가 죽는다.

논리적으로 타당(valid)하고 맞다(sound).

- **주장2:** 여자는 모두 언젠가 죽는다. 조용우는 여자다.  
따라서 조용우는 언젠가 죽는다.
  - **전제1:** 여자는 모두 언젠가 죽는다.
  - **전제2:** 조용우는 여자다.
  - **결론:** 조용우는 언젠가 죽는다.

논리적으로 타당(valid)하지만  
꼭 맞다 할 수는 없다(unsound).

## 논리학에서 중요한 건?

- 답: 논리적 타당성 logical validity
- 논리학은 주장이 논리적으로 타당한지를 증명하는데 초점을 맞추고 있다.
- 논리적으로 타당한 주장이 논리적으로 맞으려면, 전제가 참이면 된다.
- 그러나, 전제가 참인지를 알아보는 건 논리학의 영역이 아니다.
- 즉, “조용우는 여자다”라는 전제가 참인지 거짓인지 판정하는 건 논리학의 영역이 아니다.



## 주장을 적는 형식

Fitch\* Format

여자는 모두 언젠가 죽는다.  
조용우는 여자다.

조용우는 언젠가 죽는다.

} 전제  
(premises)

} 결론(conclusion)

\*Frederick Fitch: 미국 논리학자

## 증명 Proof

문장<sub>1</sub>  
문장<sub>2</sub>

...  
문장<sub>n</sub>

문장

증명이란?

전제 문장<sub>1</sub>, 문장<sub>2</sub>, ..., 문장<sub>n</sub>이 모두 참이라고  
가정한 상태에서, 문장이 참일 수 밖에  
없음을 단계별로 보여주는 절차

## 증명 방법

- 비정형적인 informal 증명
  - 말로, 자연어로
  - 대부분의 수학적 증명
- 정형적인 formal 증명
  - 추론 deduction 시스템
  - 추론 규칙 사용
  - 컴퓨터를 이용하여 증명의 자동화 가능
- 차이점
  - 둘 다 엄밀함 rigor
  - 스타일이 다를 뿐: 자연어 사용 vs 기호논리언어 사용



## 비정형적인 증명 Informal Proofs

소크라테스는 사람이다.  
모든 사람은 언젠간 죽는다.  
결국 죽게 되는 자는 가끔 죽는 걱정을 한다.

소크라테스는 가끔 죽는 걱정을 한다.

증명 : 소크라테스는 사람이고 사람은 모두 언젠간 죽는다. 따라서 소크라테스도 언젠간 죽는다. 그러나 언젠가 죽게 되는 자는 가끔 죽는 걱정을 한다고 전제하였으므로, 소크라테스도 가끔 죽는 걱정을 한다.

## 정형적인 증명 Formal Proofs

- Fitch식 증명
  - Frederic Fitch가 고안한 추론 시스템 deductive system
  - "System *F*" 라고 한다.



정당한 근거 1

...

정당한 근거 n

정당한 근거 n+1

## 증명 규칙 (정식)

[= 생성 규칙] Identity Introduction (= Intro)

▷ |  $n = n$

[= 제거 규칙] Identity Elimination (= Elim)

▷ |  $P(n)$   
...  
 $n = m$   
...  
 $P(m)$

[복사 규칙] Reiteration (Reit)

▷ |  $P$   
...  
 $P$

## 정형 증명의 예

1. Cube(c) 2. $c = b$	
3. Cube(b)	<b>= Elim: 1,2</b>

=의 대칭성 Symmetry of Identity

1. $a = b$	1. $a = b$	
...	2. $a = a$	<b>= Intro</b>
?. $b = a$	3. $b = a$	<b>= Elim: 2,1</b>

## 정형 증명의 예 (하나 더)

- 1. SameRow(a,a)
- 2.  $b = a$

?. SameRow(b,a)





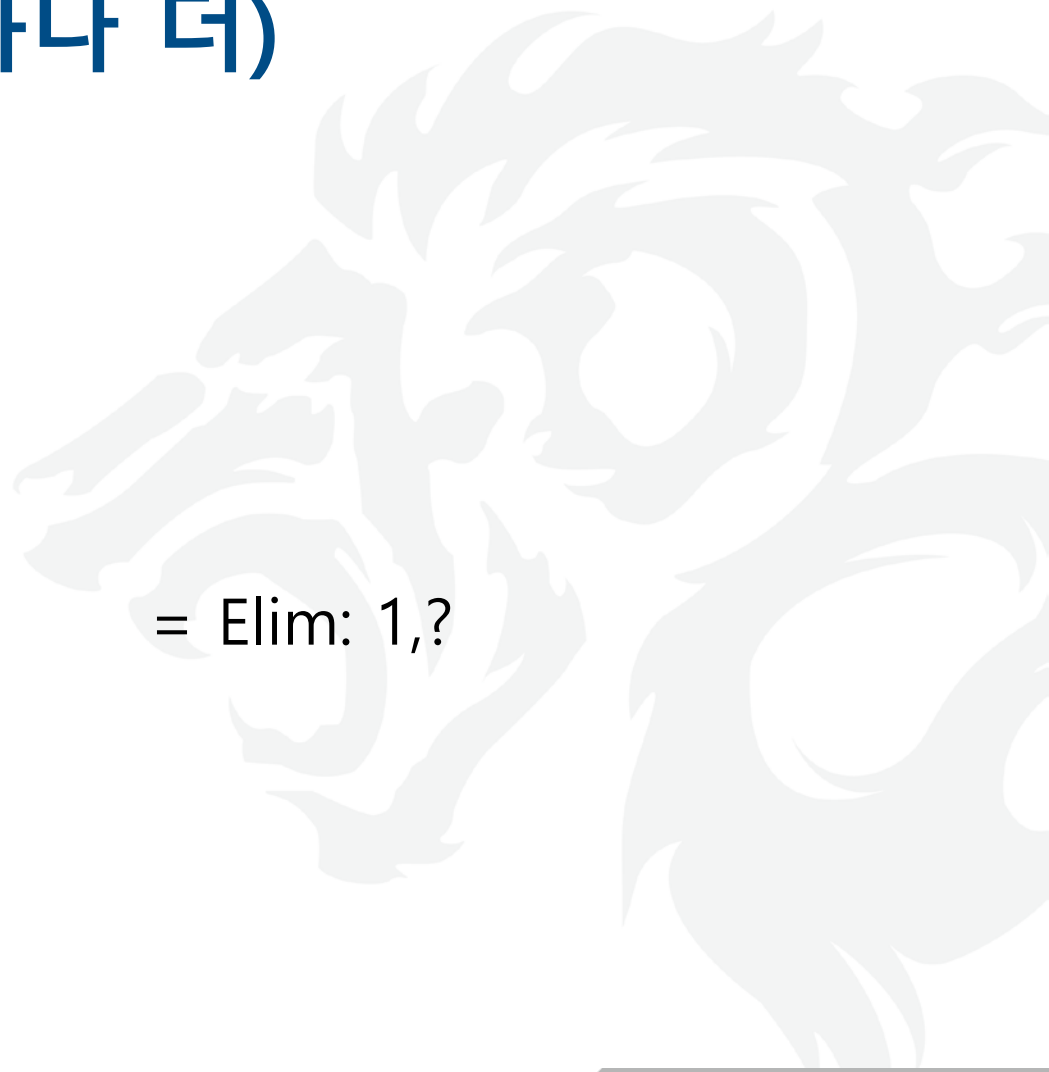
## 정형 증명의 예 (하나 더)

- 1. SameRow(a,a)
- 2.  $b = a$

?.  $a = b$

?. SameRow(b,a)

= Elim: 1,?



## 정형 증명의 예 (하나 더)

1. SameRow(a,a)

2.  $b = a$

3.  $b = b$

4.  $a = b$

5. SameRow(b,a)

= intro

= Elim: 3,2

= Elim: 1,4

## Fitch 하고 놀자!

- Fitch

- 증명 시스템인 "System F"를 구현해 놓은 정형 증명 구축 소프트웨어
- 구축한 증명의 자동 검사
- 앞에서 한 증명을 Fitch로 해보자.

- 분석적 결과 (Ana Con = Analytic Consequence)

- Ana Con 1 파일을 열어서 놀아보자.

## 타당하지 않음 invalidity을 증명하기 = Proof of nonconsequences

$P_1$	<u>반례 counterexample</u> 를 보이면 된다. 즉, $P_1, P_2, \dots, P_n$ 이 모두 사실인데 $Q$ 가 참이 아닌 예를 찾으면 된다. 그러면 이 주장이 타당하지 않음을 증명한 게 된다.
$P_2$	
$\dots$	
$P_n$	
$Q$	

이 주장이 타당한가?

반례:

"거시기"는 정치가이다.  
정치가는 대부분 거짓말쟁이다.

"거시기"는 거짓말쟁이다.

김구는 정치가이다.  
정치가는 대부분 거짓말쟁이다.

김구는 거짓말쟁이다.