

파이썬으로 배우는 머신러닝과 딥러닝

# 인공지능

## CHAPTER 5

윤정인

2020531001

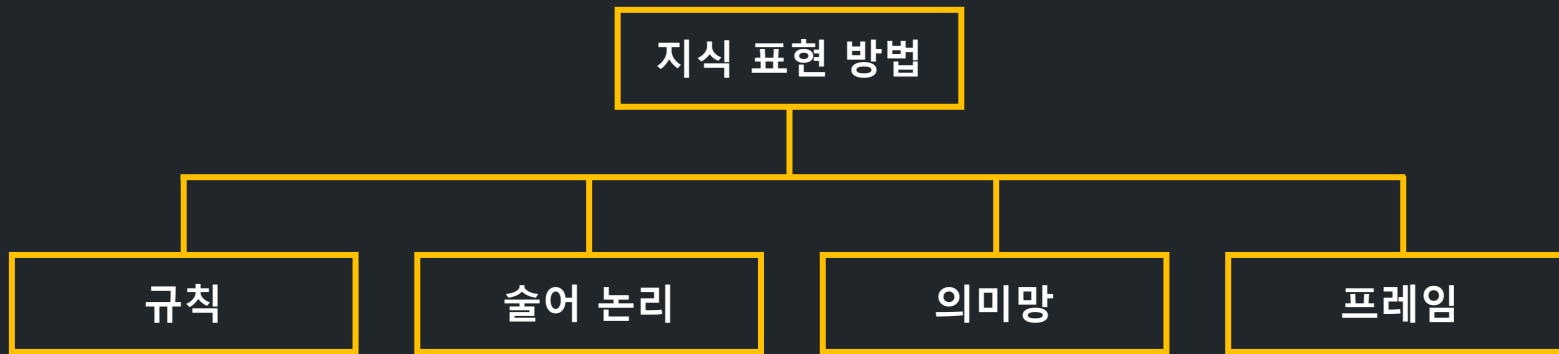
● [6주차 발표]

# Contents

- 지식표현
- 규칙
- 의미망
- 프레임
- 논리
- 명제 논리
- 술어 논리
- 술어 논리에서 추론
- 프롤로그

# 지식 표현(Knowledge Representation)

지식 표현 방법의 종류



# 지식 표현(Knowledge Representation)

지식 표현 모델

## 선언적 모델

사실, 주장

- 술어 논리
- 의미망
- 프레임

## 절차적 모델

행동, 절차

- 생성규칙 or 규칙

# 지식 표현(Knowledge Representation)

규칙 or 생성규칙(Production Rule)

규칙 #1

IF  
THEN

비가 온다  
우산을 가져간다

규칙 #2

IF  
THEN

프로그램에 버그가 없다  
프로그램은 올바르게 동작한다

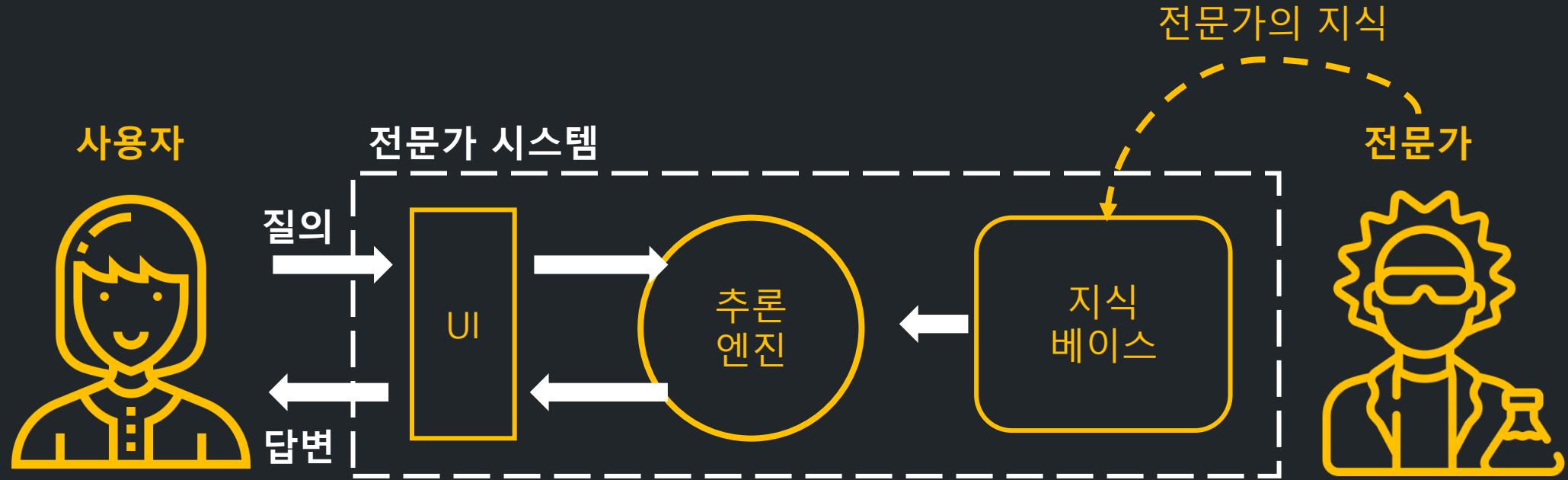
규칙 #3

IF  
THEN

습도가 높다 OR 온도가 30도 이상이다  
에어컨을 가동한다

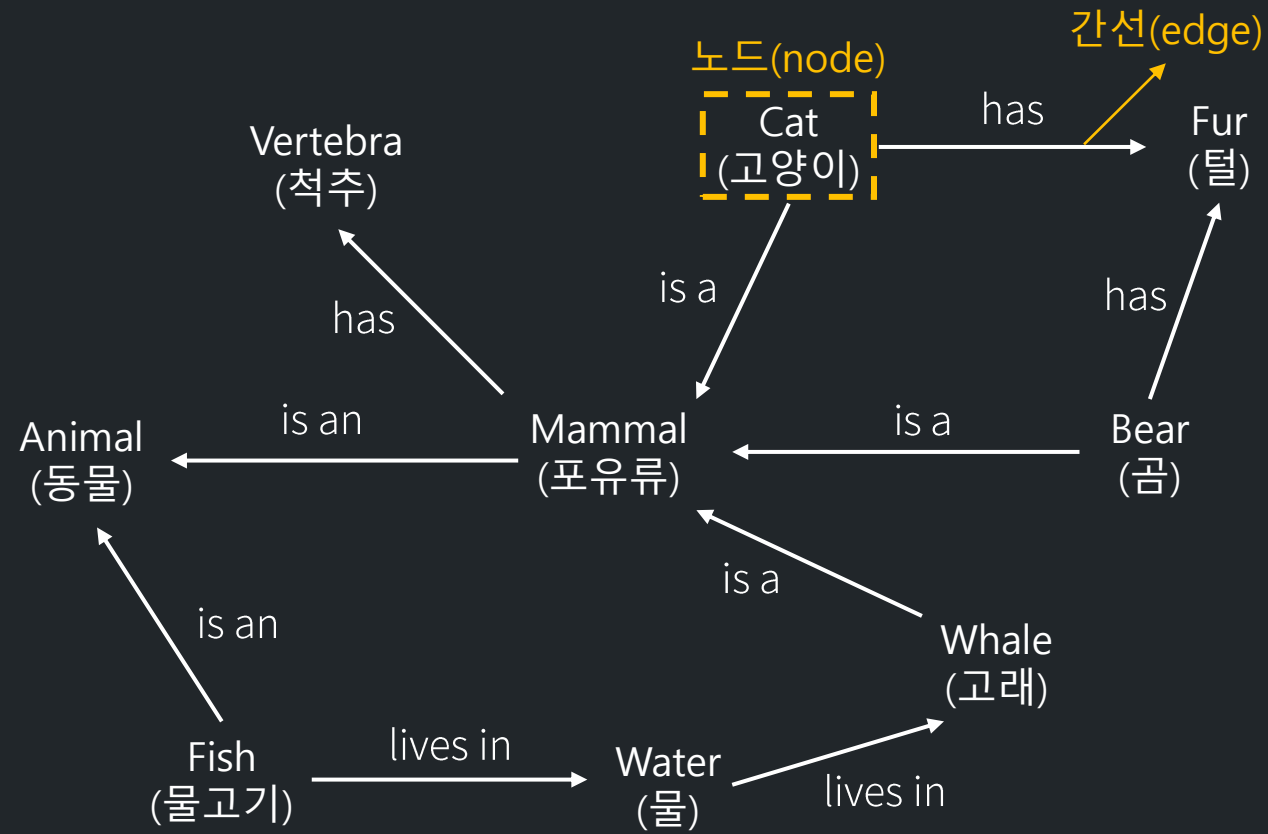
# 지식 표현(knowledge Representation)

규칙 or 생성규칙(Production Rule)



# 지식 표현(Knowledge Representation)

의미망(Semantic Network)



# 지식 표현(Knowledge Representation)

프레임(Frame)

책을 프레임으로 표현한 예

슬롯	값
Publisher	인피니티 북스
Title	인공지능
Author	홍길동
Edition	초판
Year	2019
pages	700



# 지식 표현(Knowledge Representation)

프레임(Frame)

컴퓨터를 프레임으로 표현한 예

슬롯	값
Name	컴퓨터
Subclass	기계
Types	Default: Desktop if-added: Procedure INCREMENT_COMPUTER
Speed	Default: fast if-added: Procedure CALCULATE_SPEED

## - 프로시저의 대표적인 형태

- if-added : 새로운 정보가 그 슬롯에 추가되어야 할 때 실행
- If-deleted : 어떤 값이 슬롯으로부터 제거될 때 실행(슬롯 값도 변경 필요)
- if-needed : 빈 슬롯에 어떤 값이 필요해질 때에 실행

# 지식 표현(Knowledge Representation)

프레임(Frame)과 상속

객체

인스턴스 프레임  
(Instance Frame)

인스턴스	IBM PC 2019 버전
클래스	컴퓨터
모델	IBM PC 2019
CPU	INTEL i9
RAM	32MB
그래픽 카드	GeForce
가격	2000000

클래스

클래스 프레임  
(Class Frame)

클래스	컴퓨터
모델	
CPU	
RAM	[기본값] 8MB
그래픽 카드	
가격	

# 지식 표현(Knowledge Representation)

프레임(Frame)과 상속

일부(a-part-of) 관계 표시



# 지식 표현(Knowledge Representation)

프레임(Frame)의 장단점

## 프레임의 장점

- 각각의 독립된 프레임으로 수집되어 계층적이고 조직적인 구성이 가능하여 효율적으로 문제 해결이 가능
- 관련된 지식을 한곳으로 모을 수 있으며 일반인들도 쉽게 이해할 수 있음

## 프레임의 단점

- 슬롯이나 메소드에 대한 표준이 없고 복잡하기 때문에 지식을 생성하기 힘들
- 프레임과 관련된 추론 방법이 없음
- 추론을 하려면 전문가 시스템과 결합하여 사용하여야 함

# 지식 표현(Knowledge Representation)

논리(Logic)

만약  $x$ 가 새라면  $x$ 는 날개를 가질 것이다



$(\forall x) \{ \text{is\_a}(x, \text{Bird}) \rightarrow \text{has}(x, \text{Wings}) \}$

## 장점

- 수학적 근거를 바탕으로 논리 개념을 자연스럽게 표현할 수 있다
- 지식의 첨가와 삭제가 용이하고 비교적 단순하다

## 단점

- 절차적인 지식 표현이 어렵다
- 사실의 구성 법칙이 부족하므로 실세계의 복잡한 구조를 표현하기 어렵다

# 지식 표현(Knowledge Representation)

명제 논리

## 단순 명제

P = 마트는 월요일부터 토요일까지 영업한다

Q = 오늘은 일요일이다

R = 오늘 마트는 영업하지 않는다

## 복합 명제

C = 오늘은 휴일이다

D = 오늘은 수업이 없다

E =  $C \rightarrow D$

복합 명제의 논리 표

A	B	NOT A	A AND B	A OR B	$A \rightarrow B$
T	T	F	T	T	T
T	F	F	F	T	F
F	T	T	F	T	T
F	F	T	F	F	T

# 지식 표현(Knowledge Representation)

명제 논리에서의 추론


(지식) 만약 오늘이 월요일이면, 홍길동은 일하러 간다  
(사실) 오늘이 월요일이다



(추론된 사실) 따라서 홍길동은 일하러 갈 것이다

# 지식 표현(Knowledge Representation)

명제논리 - 모더스 포넨스(Modus Ponens)

규칙	$A \rightarrow B$		"홍길동이 세계일주 중이라면 $\rightarrow$ 로또에 당첨된 것이다"
사실	A		"홍길동은 세계일주 중이다"
결론	B		"홍길동은 로또에 당첨된 것이다"



# 지식 표현(Knowledge Representation)

명제 논리 - 부정 논법(Modus Tollens)

규칙  $A \rightarrow B$

사실 NOT B



결론 NOT A

"어떤 동물이 강아지라면  $\rightarrow$  어떤 동물은 4개의 다리를 가지고 있지 않다"

"어떤 동물은 4개의 다리를 가지고 있지 않다"

"어떤 동물은 강아지가 아니다"

# 지식 표현(Knowledge Representation)

명제 논리 - 삼단 논법(syllogism)

규칙  $A \rightarrow B$

사실  $B \rightarrow C$



"소크라테스는 인간이다"

"인간은 죽는다"

결론  $A \rightarrow$   
C

"소크라테스는 죽는다"

# 지식 표현(Knowledge Representation)

술어 논리

명제 논리 : Kim has a house  
술어 논리 : HAS(Kim, house)

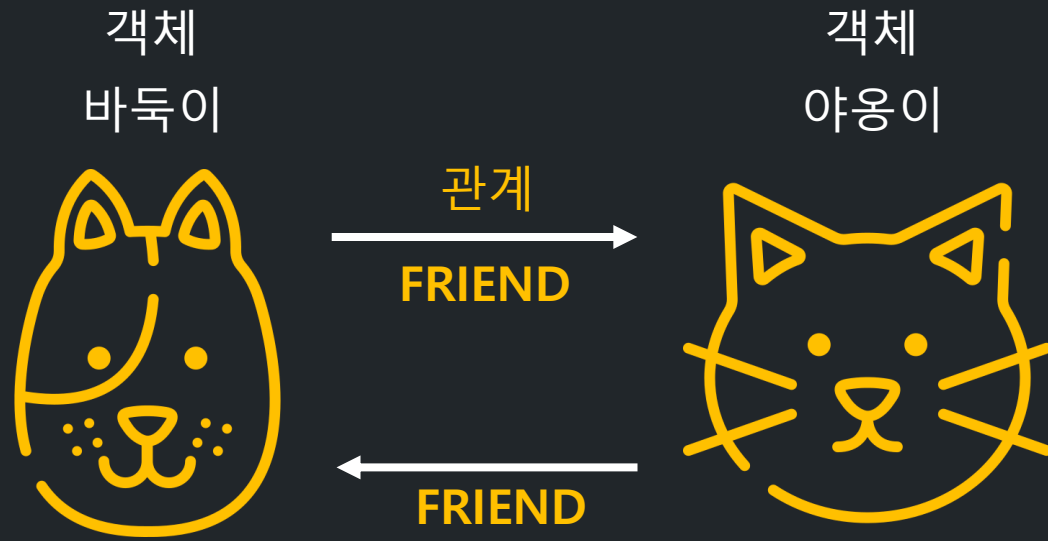
술어 : HAS  
객체 : Kim, house

명제 논리 : The orange is yellow  
술어 논리 : YELLOW(orange)

술어 : YELLOW  
객체 : orange

# 지식 표현(Knowledge Representation)

술어 논리(변수)



HUMAN(Socrates)



HUMAN(x)

# 지식 표현(Knowledge Representation)

술어 논리(한정사)

## 전칭 한정사( $\forall$ )

“모든”

All dog like cats



$\forall x [\text{Dog}(x) \rightarrow \text{LIKES}(x, \text{cat})]$

만약  $x$ 가 강아지라면  
모든  $x$ 는 고양이를 좋아한다

## 존재 한정사( $\exists$ )

“적어도 하나는 존재”

All dog like cats



$\exists x [\text{Dog}(x) \rightarrow \text{LIKES}(x, \text{cat})]$

만약  $x$ 가 강아지라면  
고양이를 좋아하는  $x$ 가 적어도 하나는 존재한다

# 지식 표현(Knowledge Representation)

술어 논리에서 추론(전칭 인스턴스화)

$\wedge$  : AND기호

$\forall x [HUMAN(x) \wedge HAS\_HOUSE(x) \rightarrow RICH(x)]$

HUMAN(Kim)

HAS\_HOUSE(Kim)

↓ 전칭 인스턴스화  
(universal instantiation)

$\forall x [HUMAN(Kim) \wedge HAS\_HOUSE(Kim) \rightarrow RICH(Kim)]$

HUMAN(Kim)

HAS\_HOUSE(Kim)

# 지식 표현(Knowledge Representation)

술어 논리에서 추론 (존재 인스턴스화)

$\wedge$  : AND기호

$$\exists x [\text{HUMAN}(x) \wedge \text{HAS\_HOUSE}(x)]$$

↓ 존재 인스턴스화  
(existential instantiation)

$$\text{HUMAN}(C) \wedge \text{HAS\_HOUSE}(C)$$

# 지식 표현(Knowledge Representation)

정형식

## 항 (term)

상수, 변수, 함수  
(*x*, *john*, *father-of*)

## 원자(atom)

술어가 항을 인수로 취한 것  
HUMAN(*x*), PILOT(*father-of*(*john*))



# 지식 표현(Knowledge Representation)

## 정형식

1. 원자는 정형식이다.
2. P와 Q가 정형식이면  $\neg P$ ,  $P \vee Q$ ,  $P \wedge Q$ ,  $P \rightarrow Q$ 도 형식적이다.
3. P가 정형식이면  $\forall x P(x)$ 와  $\exists x P(x)$ 도 정형식이다.
4. 정형식은 위의 규칙을 반복하여서 형성이 가능하다.

### 정형식이 아닌 문장

$\forall P \forall y (P(x) \vee Q(y)) \rightarrow Q(b)$

→ 전칭 한정사가 술어 P에 적용되어 있음

### 정형식

$\forall x \forall y (P(x) \vee Q(y)) \rightarrow Q(b)$

$MAN(\neg Socrates)$

→ 상수항 앞에 부정 연산자가 있음

→  $\neg MAN(Socrates)$

$father-of(MAN, john)$

→ 함수가 술어 MAN을 인수로 취하고 있음

$MAN(father-of(john))$

# 지식 표현(Knowledge Representation)

도출

## 논리곱 표준형 (CNF)방식

절



논리 합

논리 곱

# 지식 표현(Knowledge Representation)

도출

1. 함축 기호  $\rightarrow$ 를 제거한다.
2. 부정 기호를 기초 공식 안으로 이동한다. 드모르간의 법칙을 이용하여 부정의 범위를 줄인다.
3. 전칭 한정사 변수의 이름을 다르게 변경한다.
4. 존재 한정사에 의하여 한정되는 변수를 함수로 대체하고 존재 한정사를 제거한다.
5. 모든 전칭 한정사를 생략하고, 논리곱 정규형으로 변환한다.
6. 모든 논리곱 기호를 생략한다.

# 지식 표현(Knowledge Representation)

도출

문장

모든 그리스인은 유럽인이다.  
호머는 그리스인이다.  
따라서 호머는 유럽인이다.

술어 논리

$\forall x P(x) \rightarrow Q(x)$   
 $P(a)$   
 $Q(a)$

전칭 한정사 삭제  
함축 기호 삭제

$\neg P(x) \vee Q(x)$   
 $P(a)$   
 $Q(a)$

CNF

$\neg P(a) \vee Q(a)$   
 $P(a)$   
 $Q(a)$

# 지식 표현(Knowledge Representation)

도출

## 문장

- ① 모든 강아지는 포유류 이다.
- ② 바둑이는 강아지이다.
- ③ 바둑이는 포유류이다.
- ④ 모든 포유류는 우유를 먹는다.



## 술어 논리

- ①  $\forall x (DOG(x) \rightarrow MAMMAL(x))$
- ②  $DOG(badook)$
- ③  $MAMMAL(badook)$
- ④  $\forall x (MAMMAL(x) \rightarrow MILK(x))$



## CNF

- ①  $\neg DOG(x) \vee MAMMAL(x)$
- ②  $DOG(badook)$
- ③  $MAMMAL(badook)$
- ④  $\neg MAMMAL(x) \vee MILK(x)$

# 지식 표현(Knowledge Representation)

도출

1. 증명하고자 하는 사실을 부정하여 절들의 리스트에 추가한다.
2. 지식 베이스의 문장들을 CNF형태로 변환한다.
3. 도출할 수 있는 절의 쌍이 더 이상 없을 때 까지 다음을 반복한다.
  - 3.1 도출할 수 있는 절의 쌍을 찾아 도출한다.
  - 3.2 도출 절을 절들의 리스트에 추가한다.
  - 3.3 NIL이 유도되면, 증명하고자 하는 사실이 참이다.
4. 증명하고자 하는 사실이 거짓이다.

# 지식 표현(Knowledge Representation)

도출에 의한 증명

## 절들의 리스트

- ①  $\neg \text{DOG}(x) \vee \text{MAMMAL}(x)$
- ②  $\text{DOG}(\text{badook})$
- ③  $\text{MAMMAL}(\text{badook})$
- ④  $\neg \text{MAMMAL}(x) \vee \text{MILK}(x)$



## 증명하고 싶은 사실

$\text{MILK}(\text{badook})$



## 부정으로 만들어서 리스트에 추가

- ①  $\neg \text{DOG}(x) \vee \text{MAMMAL}(x)$
- ②  $\text{DOG}(\text{badook})$
- ③  $\text{MAMMAL}(\text{badook})$
- ④  $\neg \text{MAMMAL}(x) \vee \text{MILK}(x)$
- ⑤  $\neg \text{MILK}(\text{badook})$

# 지식 표현(Knowledge Representation)

도출에 의한 증명

①  $\neg \text{DOG}(x) \vee \text{MAMMAL}(x)$

②  $\text{DOG}(\text{badook})$

③  $\text{MAMMAL}(\text{badook})$

④  $\neg \text{MAMMAL}(x) \vee \text{MILK}(x)$

⑤  $\neg \text{MILK}(\text{badook})$

도출 법칙

①  $\neg \text{DOG}(x) \vee \text{MAMMAL}(x)$

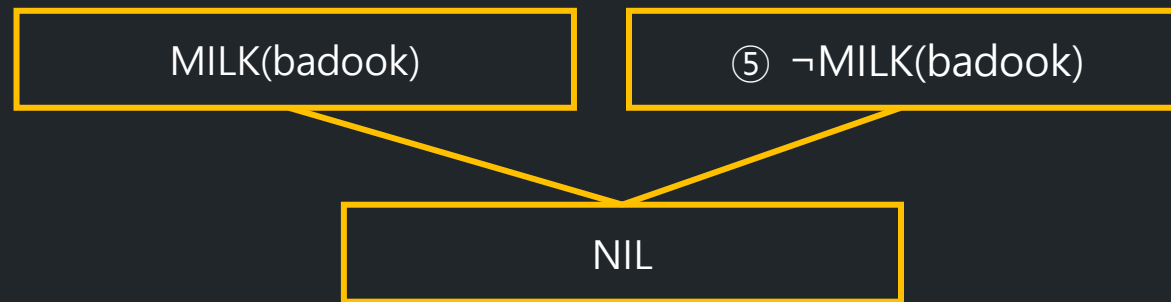
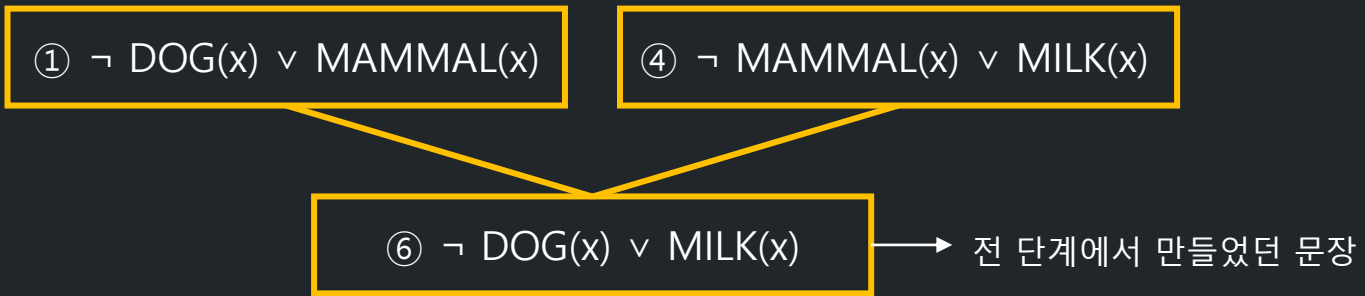
④  $\neg \text{MAMMAL}(x) \vee \text{MILK}(x)$

합치기

⑥  $\neg \text{DOG}(x) \vee \text{MILK}(x)$



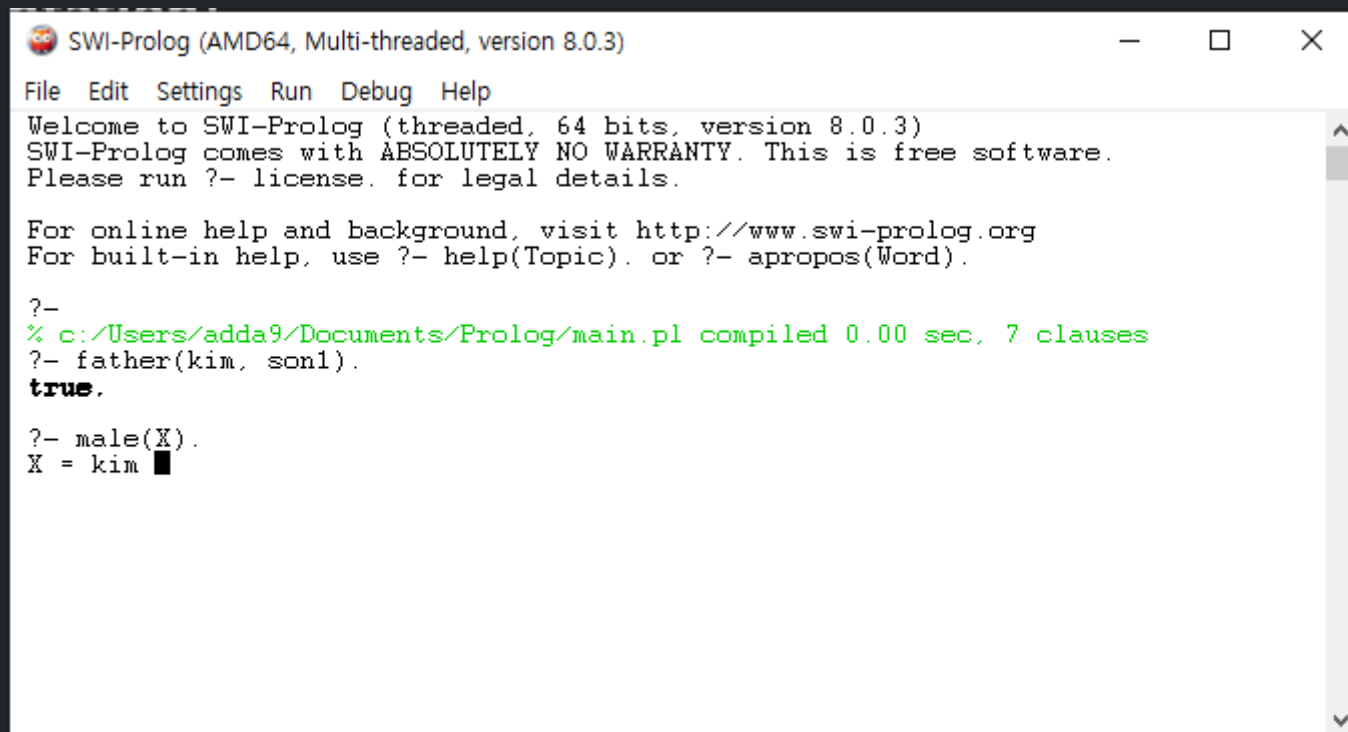




NIL값이 도출된다면 증명하고자 하는 것은 참이라는 결론

# 지식 표현(Knowledge Representation)

프롤로그



```
SWI-Prolog (AMD64, Multi-threaded, version 8.0.3)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

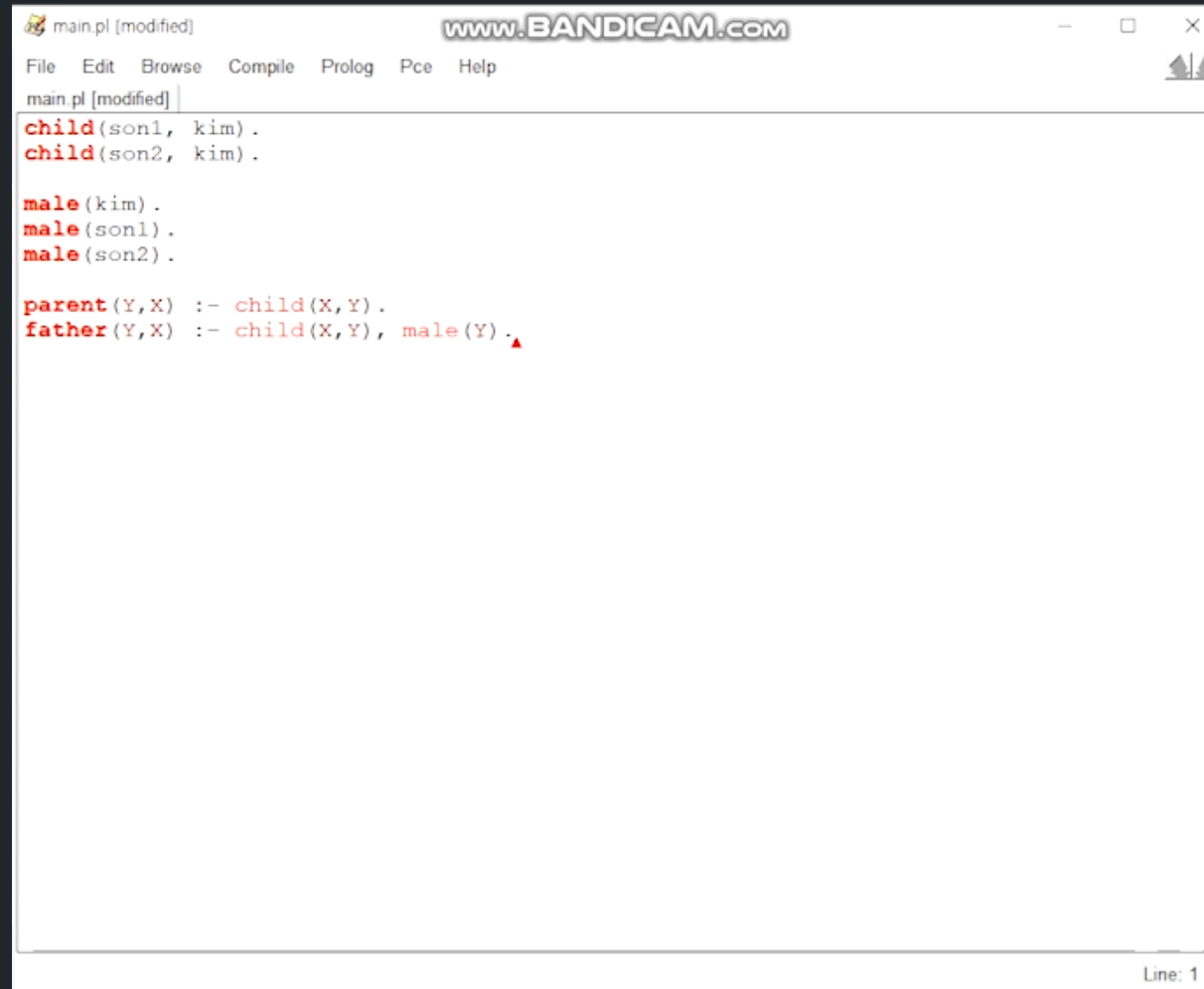
?-
% c:/Users/adda9/Documents/Prolog/main.pl compiled 0.00 sec, 7 clauses
?- father(kim, son1).
true.

?- male(X).
X = kim
```

Download : <https://www.swi-prolog.org/>

# 지식 표현(Knowledge Representation)

프롤로그



The screenshot shows a Prolog IDE window titled 'main.pl [modified]'. The menu bar includes 'File', 'Edit', 'Browse', 'Compile', 'Prolog', 'Pce', and 'Help'. The code editor contains the following Prolog code:

```
child(son1, kim).  
child(son2, kim).  
  
male(kim).  
male(son1).  
male(son2).  
  
parent(Y,X) :- child(X,Y).  
father(Y,X) :- child(X,Y), male(Y).
```

The status bar at the bottom right indicates 'Line: 1'.

1. [File] – [New]를 실행한 뒤 파일명을 [main.pl]로 설정하고 다음과 같이 코드 작성

# 지식 표현(Knowledge Representation)

프롤로그



The screenshot shows a window titled "SWI-Prolog (AMD64, Multi-threaded, version 8.0.3)" with a menu bar (File, Edit, Settings, Run, Debug, Help) and a toolbar. The main text area displays the following content:

```
www.BANDICAM.com
Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% c:/Users/adda9/Documents/Prolog/main.pl compiled 0.00 sec, 7 clauses
?- father(kim, son1).
true.

?- male(X).
X = kim
```

2. 작성이 완료되면 다시 Prolog로 돌아와 [File] – [Consult]를 클릭하여 [main.pl]파일 불러오기
3. 사진과 같이 참이나 거짓 혹은 조건을 만족하는 문장을 입력해보기

THANK YOU