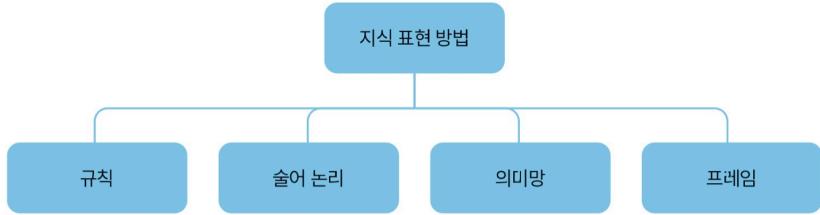


지식 표현, 추론, 그리고 증명

내용

- 지식 표현과 방법
- 술어논리
- 술어논리의 추론 기법
- 술어논리를 실습
 - Prolog 언어

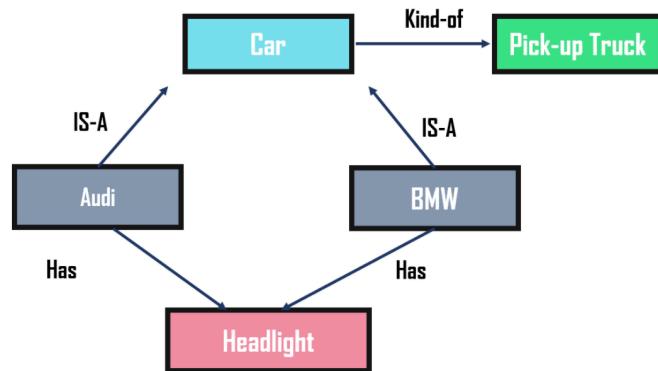
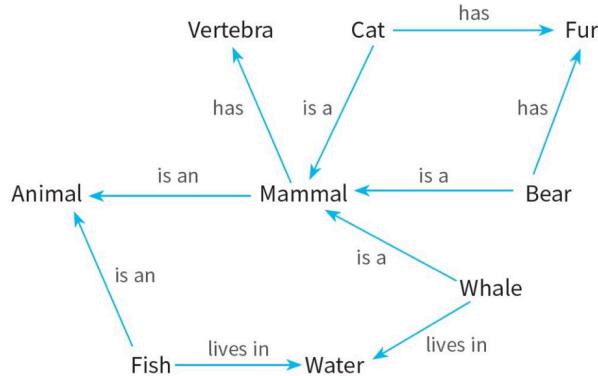
지식표현 방법



표현 모델	종류
생성규칙(production rule) 또는 규칙 (rule)	절차적 (procedural)
술어 논리 (predicate logic)	
의미망 (semantic net)	선언적 (declarative)
프레임 (frame)	

의미망, Semantic Network

- 의미망(semantic network)은 방향 그래프를 이용하여 개념 간의 관계를 표현
- 그래프는 노드와 간선으로 구성
 - 노드는 사물(objects), 개념(concepts) 등을 대표
 - 간선(edge)는 사물이나 개념 사이의 관계



의미망

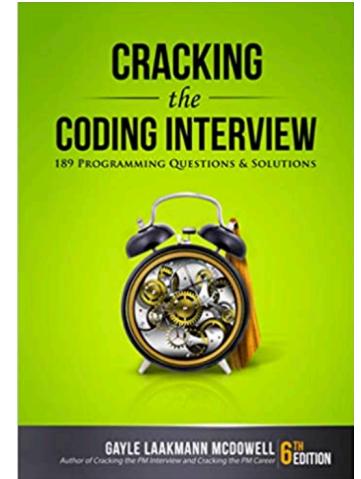
- 간선
 - ..의 일종이다 (is-a, inheritance)
 - ..을 가지고 있다 (has)
 - ..은 하나의 예 (instance-of, kind-of)
- 의미망은 매우 복잡한 개념이나 인과 관계를 잘 표현
- 지식의 양이 커지면 너무 복잡해져서 조작이 어렵다는 단점을 갖음
- 의미망을 위한 표준이 없어 사용하는 응용시스템에 따라 의미 망의 형태가 다양함

프레임, Frame



- 프레임(frame)은 1970년대에 Marvin Minsky가 제안한 지식 표현 방법으로 의미망에서 파생
- 프레임은 특정 객체와 그 속성을 하나로 조직화하는 방법이며 개념, 객체, 상황들을 기술하는데 유리

Title	Cracking the coding interview: 189 Programming Questions and Solutions
Publisher	CareerCup
Author	Gayle Laakmann McDowell
Edition	6th edition
Year	2015
Pages	687



프레임



- 프레임은 객체의 속성과 값으로 구성
- 속성은 슬롯 (slot)이며 값 (value)을 갖음
 - default 값
 - 프레임 포인터
 - 프로시저 : 슬롯 검색, 변경, 제거 시 실행되는 함수
- 프레임은 프로그래밍 언어의 구조체나 객체(object)와 유사
- 객체의 필드에 해당하는 것은 슬롯임

Title	Cracking the coding interview: 189 Programming Questions and Solutions
Publisher	CareerCup
Author	Gayle Laakmann McDowell
Edition	6th edition
Year	2015
Pages	687

프로시저, Procedure

- “if-added” 프로시저
 - 새로운 정보가 그 슬롯에 추가되어야 할 때 실행
- “if-deleted” 프로시저
 - 어떤 값이 슬롯으로부터 제거될 때 실행
 - 다른 슬롯의 값도 변경할 필요가 있을 수 있음
- "if-needed" 프로시저
 - 빈 슬롯에 어떤 값이 필요해질 때에 실행

프레임과 객체 지향 프로그래밍

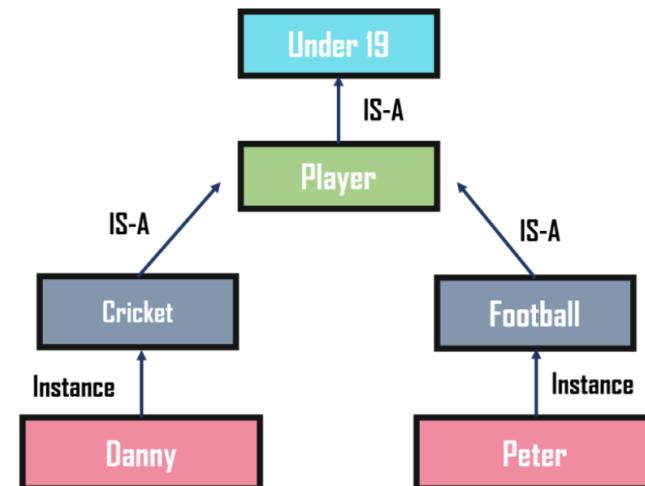
- 프레임은 객체 (object)임
- 객체의 필드에 해당하는 것이 프레임의 슬롯
- 프레임의 슬롯은 값뿐만 아니라 프로시저를 갖음
- 슬롯에 붙은 프로시저가 바로 객체의 메소드가 됨

프레임과 상속

- 인스턴스 프레임(instance frame)과 클래스 프레임(class frame)
으로 나눔

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

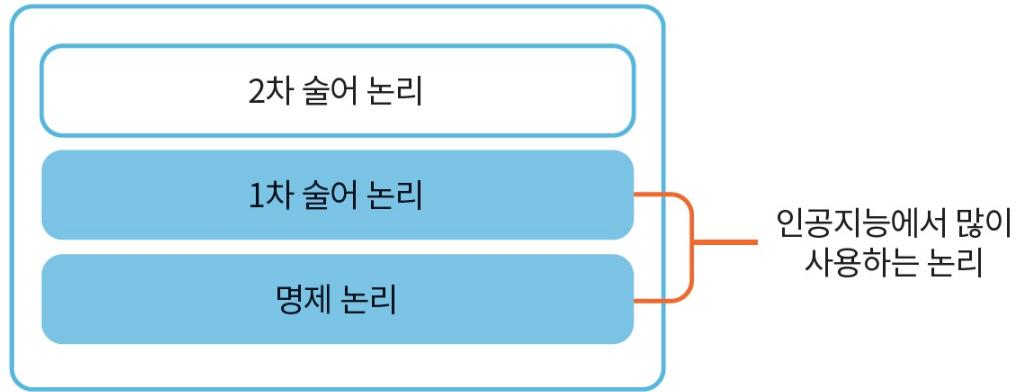
인스턴스	IBM PC 2019 버전
클래스	컴퓨터
모델	IBM PC 2019
CPU	INTEL i9
RAM	32MB
그래픽 카드	GEFORCE
가격	2000000



논리, Logic



전통적인 논리



- 지식을 표현하기 위해 수학에서 논리를 사용
- 명제 논리와 술어 논리는 수학 논리에 기초

술어논리, Predicate

- “만약 x가 새라면, x는 날개를 가질 것이다”

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

- “만약 x가 사람이라면, x는 어떤 언어 y로 말할 수 있다”

$$(\forall x) \{ \text{people}(x) \rightarrow \text{can_speak}(x, y) \text{ and } \text{is_language}(y) \}$$

술어논리

장점

- 수학적인 근거를 바탕으로 논리 개념을 자연스럽게 표현
- 지식의 정형화 영역에 적합
- 정리 증명(theorem proving) 기법을 사용
- 지식의 추가와 삭제가 용이하고 비교적 단순

단점

- 절차적인 지식 표현이 어려움
- 사실의 구성 법칙이 부족하여 실세계의 복잡한 구조를 표현하기 어려움

명제 논리, Propositional Logic

- 기호 논리학에서 명제(proposition)는 참(true, T)이거나 또는 거짓(false, F)을 판별할 수 있는 문장

P = 마트는 월요일부터 토요일까지 영업한다.

Q = 오늘은 일요일이다.

R = 오늘 마트는 영업하지 않는다.

- 논리 연산자 사용 가능

Q = 오늘은 일요일이다.

NOT Q = 오늘은 일요일이 아니다.

J = 옷은 파랑색이다.

K = 옷은 스트라이프 무늬가 있다.

L = J AND K = 옷은 파랑색이고 스트라이프 무늬가 있다.

논리 연산자

C = 오늘은 휴일이다.

D = 오늘은 수업이 없다.

E = C → D

A	B	NOT A	A AND B	A OR B	A→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

명제논리에서의 추론



- 추론(inference)이란 우리가 가지고 있는 지식과 우리가 이미 알고 있는 사실로부터 새로운 사실을 유추

(지식) 우리집 강아지는 집안에 있거나 앞마당에 있다.

(사실) 강아지가 집안에 없다.

=====

(추론된 사실) 강아지는 앞마당에 있을 것이다.

추론법칙

- 모더스 포넌스(Modus Ponens)

규칙 | $A \rightarrow B$

사실 | A

결론 B

"홍길동이 세계 일주 중이라면 \rightarrow 로또에 당첨된 것이다. "

"홍길동은 세계 일주 중이다."

.:"홍길동은 로또에 당첨된 것이다."

추론법칙



- 부정 논법(Modus Tollens)

규칙 | $A \rightarrow B$

사실 | NOT B

결론 NOT A

"어떤 동물이 강아지라면 \rightarrow 어떤 동물은 4개의 다리를 가지고 있다."

"어떤 동물은 4개의 다리를 가지고 있지 않다."

.:"어떤 동물은 강아지가 아니다."

추론법칙

- 삼단논법(syllogism)

규칙 | $A \rightarrow B$

사실 | $B \rightarrow C$

결론 $A \rightarrow C$

“소크라테스는 인간이다”

“인간은 모두 죽는다”

.：“소크라테스는 죽는다”

술어논리



- 명제 논리에서 우리는 전체 명제가 참이냐 거짓이냐 만을 말할 수 있음
- 예: “신호등이 파랑색이다.”라는 명제가 있다면 이것을 “신호등”과 “파랑색이다”로 나눌 수 있음
- 술어 논리에서는 하나의 명제가 객체(object, 또는 인수)와 술어(predicate)로 구성됨
 - 변수, variable
 - 한정자, quantifier
- 이와 같은 술어 논리의 특징 때문에 명제를 사용하는 것보다, 훨씬 더 구체적으로 지식을 표현할 수 있음

명제 논리: Kim has a house

술어 논리: HAS(Kim, house)

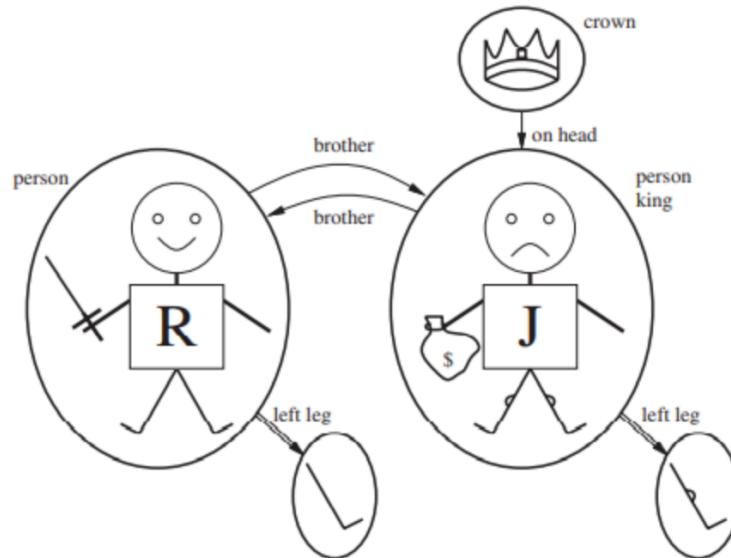
명제 논리: The orange is yellow

술어 논리: YELLOW(orange)

객체와 관계

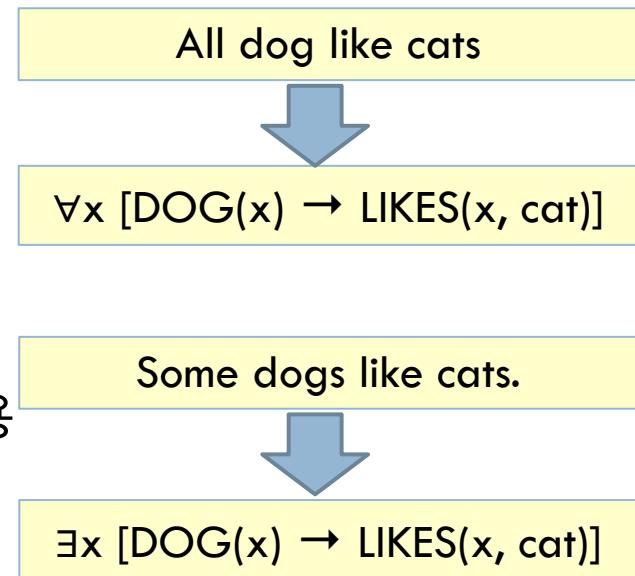


- 객체는 상수 기호로 나타낸다: 바둑이, 야옹이, Richard, ...
- 관계는 술어 기호로 나타낸다: HUMAN, DOG, CAT, HAT, ...



변수와 한정자

- “ x 가 인간이라면”은 다음과 같이 표현
 - $\text{HUMAN}(x)$
- x 가 인간이라면 위의 술어 논리식은 참임
- 한정자는 변수의 범위를 서술하는 기호
- 술어 논리에는 전칭 한정사(universal quantifier)
 - \forall 와 존재 한정사(existential quantifier) \exists 를 사용
 - 전칭 한정사 \forall 는 “모든”
 - 존재 한정사 \exists 는 “적어도 하나는 존재”



술어논리에서의 추론

- 방법
 1. 술어 논리식을 명제 논리식으로 변환한 후에 명제 논리의 추론
 2. 도출(resolution)
- 모든 지식이 정형식(WFF: well-formed formula)으로 표현되어야 함
 - 기초 공식은 정형식이다.
 - P와 Q가 정형식이면 $\neg P$, $P \vee Q$, $P \wedge Q$, $P \rightarrow Q$ 도 정형식이다.
 - P가 정형식이면 $\forall x P(x)$ 와 $\exists x P(x)$ 도 정형식이다.
 - 정형식은 위의 규칙을 반복하여서 형성 가능하다.

정형식 표현

$human(socrates)$

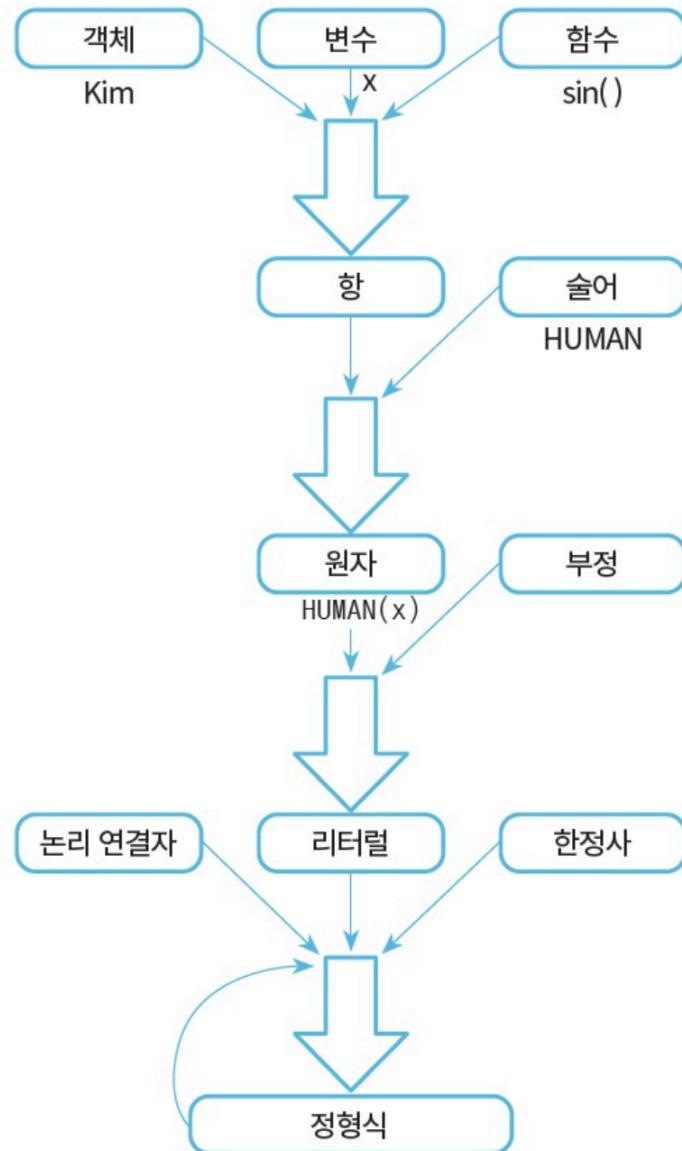
$pilot(father_of(john, jim))$

$\forall x \ number(X) \rightarrow less_then(y, x)$

$\forall x \forall y \ (p(x) \vee q(x) \rightarrow q(b))$

$\neg human(socrates)$

$father_of(john, jim)$



$\forall x[HUMAN(x) \rightarrow Mortal(x)]$

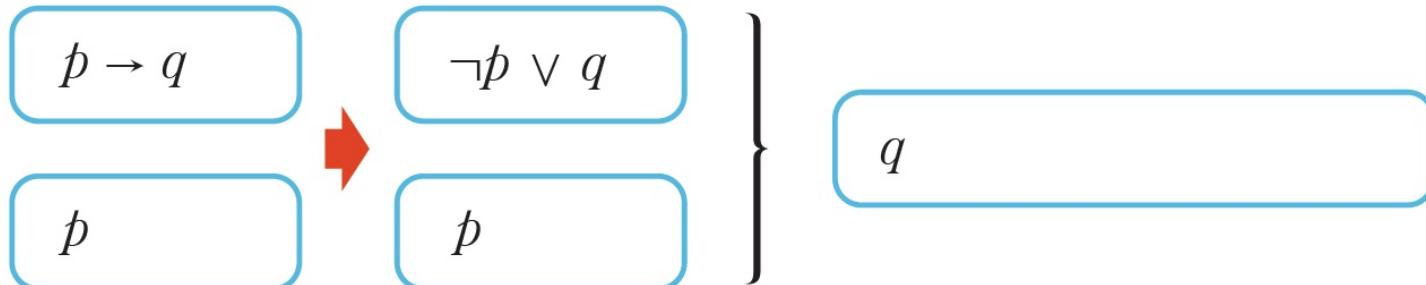
도출



- 도출은 리터럴과 부정 리터럴을 가지고 있는 2개의 절을 조합하여서 새로운 절을 생성



- 모더스 포넌스를 유도



도출 사용

- 도출 기법으로 증명하려면 모든 논리식들을 논리곱 표준형(CNF: conjunctive normal form)으로 변경
- 모든 전칭 한정사는 삭제
- 존재 한정사는 스콜렘 (Skolem) 함수로 변경

CNF로 변환

1. 함축 기호 →를 제거
2. 부정 기호를 기초 공식 안으로 이동
 - 드모르간의 법칙을 이용하여 부정의 범위를 축소
3. 전칭 한정사 변수의 이름을 다르게 변경
4. 존재 한정사에 의하여 한정되는 변수를 함수로 대체하고
존재 한정사를 제거
5. 모든 전칭 한정사를 생략하고, 논리곱 정규형으로 변환
6. 모든 논리곱 기호를 생략

CNF로 변환하기 예

- ① 모든 강아지는 포유류이다.
- ② 바둑이는 강아지이다.
- ③ 바둑이는 포유류이다.
- ④ 모든 포유류는 우유를 생산한다.



술어논리로 변환

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



CNF로 변환

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

도출 기반 증명



1. 증명하고자 하는 사실을 부정하여 절들의 리스트에 추가
2. 지식 베이스의 문장들을 CNF 형태로 변환
3. 도출할 수 있는 절의 쌍이 더 이상 없을 때까지 다음 반복
 - 도출할 수 있는 절의 쌍을 찾아 도출
 - 도출절을 절들의 리스트에 추가
 - NIL이 유도되면, 증명하고자 하는 사실이 참임
4. 증명하고자 하는 사실은 거짓임

도출 기반 증명의 예

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



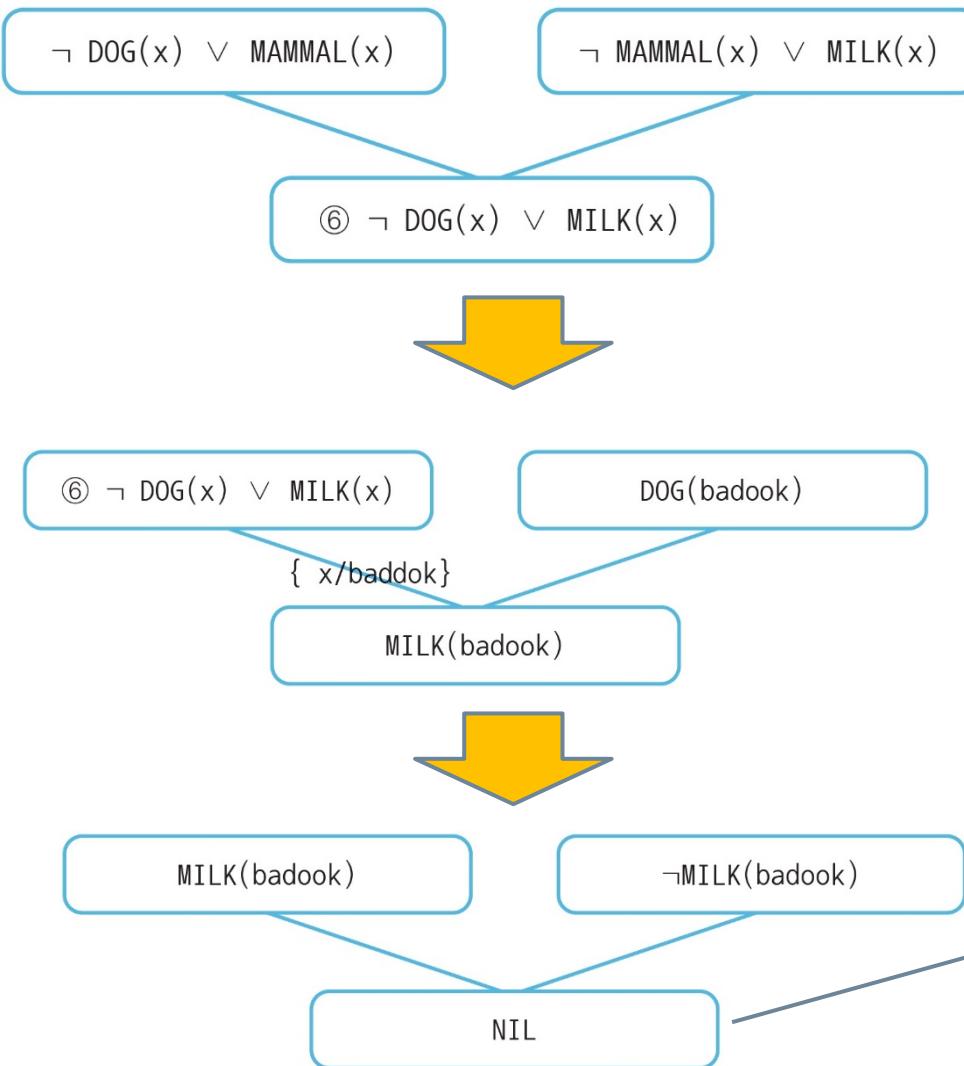
증명하고 싶은 사실

MILK(badook)



- 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 기반 증명의 예



프롤로그, Prolog

The screenshot shows the SWISH Prolog interface. At the top, there's a toolbar with icons for file operations, a search bar containing "swish.swi-prolog.org", and a user count of "143 users online". Below the toolbar is a navigation bar with tabs for "examples" (selected), "kb" (closed), and a "+" button. The main area contains a code editor with the following Prolog code:

```
1 % Some simple test Prolog programs
2 %
3 %
4 % Knowledge bases
5
6 loves(vincent, mia).
7 loves(marcellus, mia).
8 loves(pumpkin, honey_bunny).
9 loves(honey_bunny, pumpkin).
10
11 jealous(X, Y) :- loves(X, Z),
12      loves(Y, Z).
13
14
15 /** <examples>
16
17 ?- loves(X, mia).
18 ?- jealous(X, Y).
19
20
21 */
22
23
```

To the right of the code editor is a large, cartoonish owl logo. Below the owl, there are three query windows:

- Top Query Window:** Shows the query `loves(X, mia)` with the result `X = vincent`. It includes buttons for "Next" (10, 100, 1,000) and "Stop".
- Middle Query Window:** Shows the query `loves(X, mia), jealous(X, Y).` with the result `X = Y, Y = vincent`. It includes buttons for "Next" (10, 100, 1,000) and "Stop".
- Bottom Query Window:** Shows the query `?- loves(X, mia), jealous(X, Y).` with the result (empty). It includes buttons for "Next" (10, 100, 1,000) and "Stop".

At the bottom of the interface, there are buttons for "Examples▲", "History▲", "Solutions▲", "table results" (unchecked), and a "Run!" button.

프롤로그, Prolog



```
SWI-Prolog (AMD64, Multi-threaded, version 8.1.11)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.1.11-7-g38f795726)
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/chun/Documents/Prolog/main.pl compiled 0.00 sec, 7 clauses
?- father(kim, son1).
true.

?- male(X).
X = kim ;
X = son1 ;
X = son2.

?-
```

프롤로그, Prolog

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

- 변수를 항상 대문자로 표현
- “father(Y,X) :- child(X,Y), male(Y).”은 X가 Y의 자식이고 Y가 남성다면 Y는 X의 아버지이다. “를 의미
 - $\text{child}(X,Y) \wedge \text{male}(Y) \rightarrow \text{father}(Y,X)$
- 프롤로그의 규칙에서 논리합은 사용하지 않으며 논리곱으로 지식을 표현

사실 (fact) 표현



- 상수는 소문자로 입력한다.

```
child(son1, kim).  
child(son2, kim).  
male(kim).  
male(son1).  
male(son2).
```

예

- ① 다음과 같은 문장들을 저장하는 파일 main.pl을 생성한다.

```
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).
```

- ② [File]->[Consult...] 메뉴 항목을 선택하여서 main.pl을 불러들인다.

정리

- 지식 표현의 방법: 규칙, 의미망, 프레임, 술어논리 등
 - 의미망(semantic network)은 방향 그래프를 이용하여 개념 간의 관계를 나타냄
 - 프레임(frame)은 특정 객체와 그 속성을 묶어서 하나로 조직화하는 표현임
 - 술어 논리(predicate logic)는 하나의 명제가 객체(object, 또는 인수)와 술어(predicate)로 나누어질 수 있고 변수와 한정자를 사용할 수 있는 논리
- 명제 논리에서 추론 방법: 모더스 포넌스, 부정 논법, 삼단논법 등
- 술어 논리식에서는 추론을 위하여 도출(resolution)을 이용
- 논리 프로그램 언어 Prolog의 소개