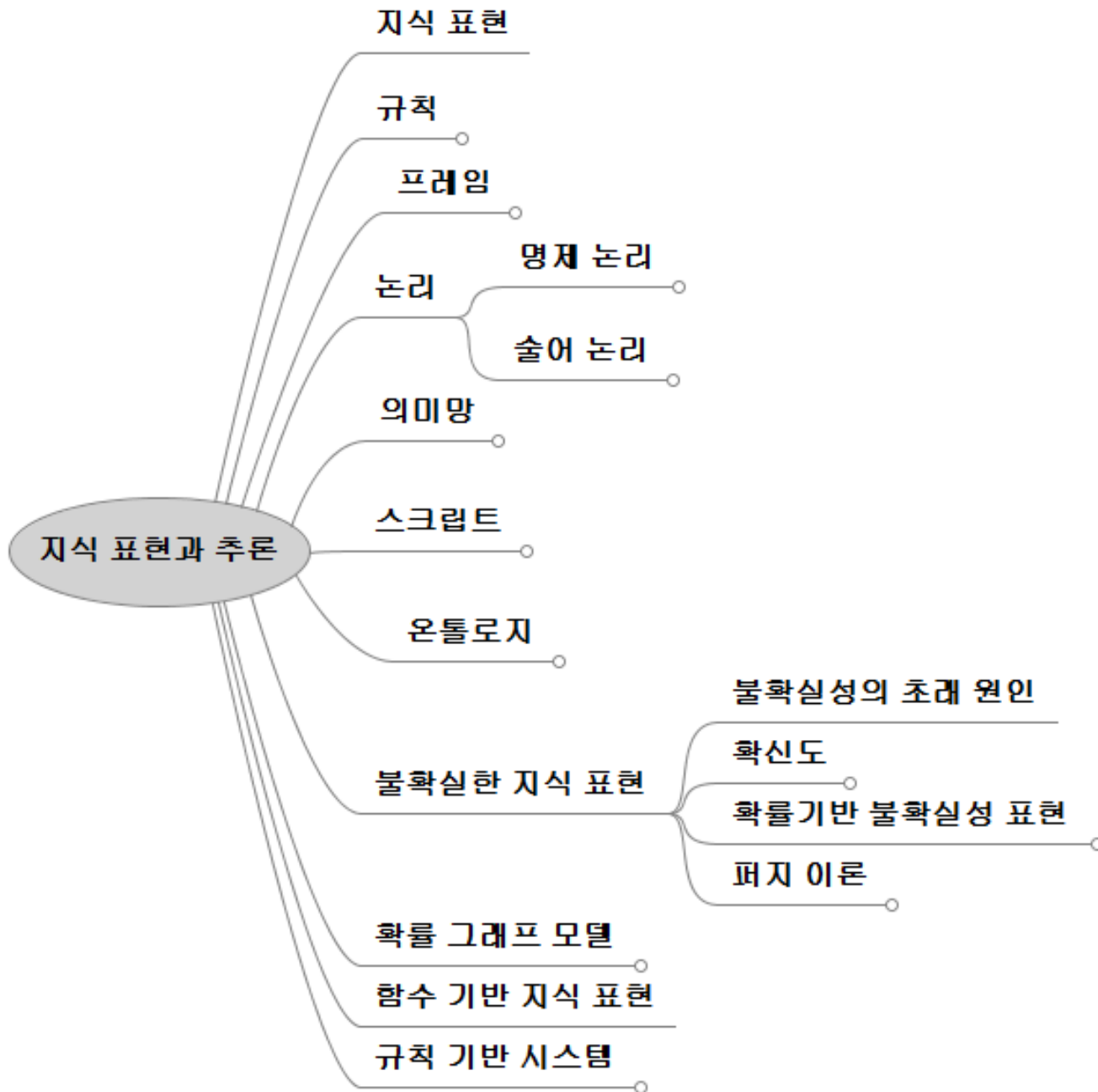


지식표현과 추론

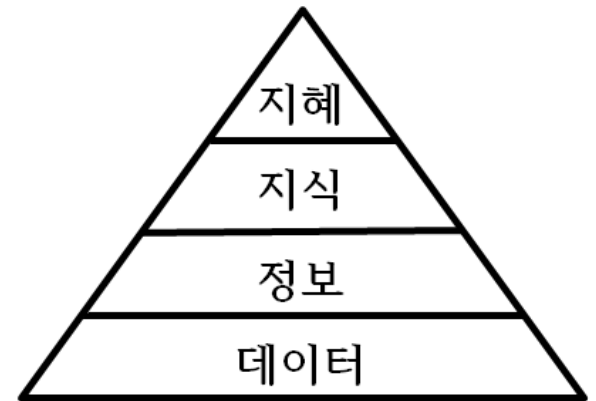
1. 지식
2. 규칙
3. 프레임
4. 논리
5. 의미망
6. 스크립트
7. 온톨로지
8. 함수에 의한 지식 표현
9. 불확실한 지식 표현
10. 규칙 기반 시스템
11. 심볼 그라운드링 문제와 프레임 문제
12. CYC 프로젝트



1. 지식 표현

❖ 데이터 피라미드

- **데이터 (data)**
 - 특정 분야에서 **관측된 아직 가공되는 않은 것**
 - 사실인 것처럼 관측되지만 **오류나 잡음**을 포함 가능
- **정보 (information)**
 - 데이터를 **가공**하여 어떤 **목적이나 의미**를 갖도록 한 것
- **지식 (knowledge)**
 - 정보를 **취합**하고 **분석**하여 얻은 대상에 대해 사람이 **이해**한 것
- **지혜 (wisdom)**
 - 경험과 학습을 통해서 얻은 지식보다 높은 수준의 **통찰**



지식 표현

- ❖ **지식(知識, knowledge)**
 - **경험이나 교육을 통해 얻어진 전문적인 이해(understanding)와 체계화된 문제 해결 능력**
 - 어떤 주제나 분야에 대한 **이론적 또는 실제적인 이해, 또는 현재 알려진 사실과 정보의 모음**
 - **암묵지(暗黙知, tacit knowledge)**
 - 형식을 갖추어 표현하기 어려운, 학습과 경험을 통해 쌓은 지식
 - **형식지(形式知, explicit knowledge)**
 - 비교적 쉽게 형식을 갖추어 표현될 수 있는 지식
 - **절차적 지식(procedural knowledge)**
 - 문제해결의 절차 기술
 - **선언적 지식(declarative knowledge)**
 - 어떤 대상의 성질, 특성이나 관계 서술
 - **컴퓨터를 통한 지식 표현 및 처리**
 - 프로그램이 쉽게 처리할 수 있도록 **정형화된 형태로 표현**
 - 규칙, 프레임, 논리, 의미망, 스크립트, 수치적 함수 등

2. 규칙

❖ 규칙 (rule)

- '~이면, ~이다' 또는 '~하면, ~하다'와 같은 **조건부의 지식**을 표현하는 IF-THEN 형태의 문장
- 직관적
- 이해하기 쉬움

❖ 규칙 획득 및 표현

- 예: 신호등이 녹색일 때는 건널목을 안전하게 건널 수 있고, 빨간색일 때는 길을 건너지 말아야 한다
- 대상, 속성, 행동 또는 판단의 정보 추출
 - 대상 : 신호등
 - 속성 : 녹색, 빨간색
 - 행동/판단 : 건넌다, 멈춘다.
- 표현
 - IF 신호등이 녹색이다 THEN 행동은 건넌다
 - IF 신호등이 빨간색이다 THEN 행동은 멈춘다

규칙

❖ 규칙 (rule)

- IF 신호등이 녹색이다 THEN 행동은 건너다
- IF 신호등이 빨간색이다 THEN 행동은 멈춘다

- IF trafficLight = green THEN action = cross
- IF trafficLight = red THEN action = stop

- IF 부분
 - 주어진 정보나 사실에 대응될 조건
 - 조건부(conditional part, antecedent)

- THEN 부분
 - 조건부가 만족될 때의 판단이나 행동
 - 결론부(conclusion, consequent)

규칙

❖ 규칙의 구성

▪ 조건부

- 둘 이상의 조건을 **AND** 또는 **OR**로 결합하여 구성 가능
 - **IF** <조건1> **AND** <조건2> **AND** <조건3> **THEN** <결론>
 - **IF** <조건1> **OR** <조건2> **OR** <조건3> **THEN** <결론>

▪ 결론부

- 여러 개의 판단 또는 행동 포함 가능
 - **IF** <조건>
THEN <결론1>
AND <결론2>
AND <결론3>

규칙

❖ 규칙을 통한 지식 표현

▪ 인과관계

- 원인을 조건부에 **결과**는 결론부에 표현
 - IF 연료통이 빈다
THEN 차가 멈춘다

▪ 추천

- 상황을 조건부에 기술하고 이에 따른 **추천 내용**을 결론부에 표현
 - IF 여름철이다 AND 날이 흐리다
THEN 우산을 가지고 가라

▪ 지시

- 상황을 조건부에 기술하고 이에 따른 **지시 내용**을 결론부에 표현
 - IF 차가 멈추었다 AND 연료통이 비었다
THEN 주유를 한다

규칙

❖ 규칙을 통한 지식 표현

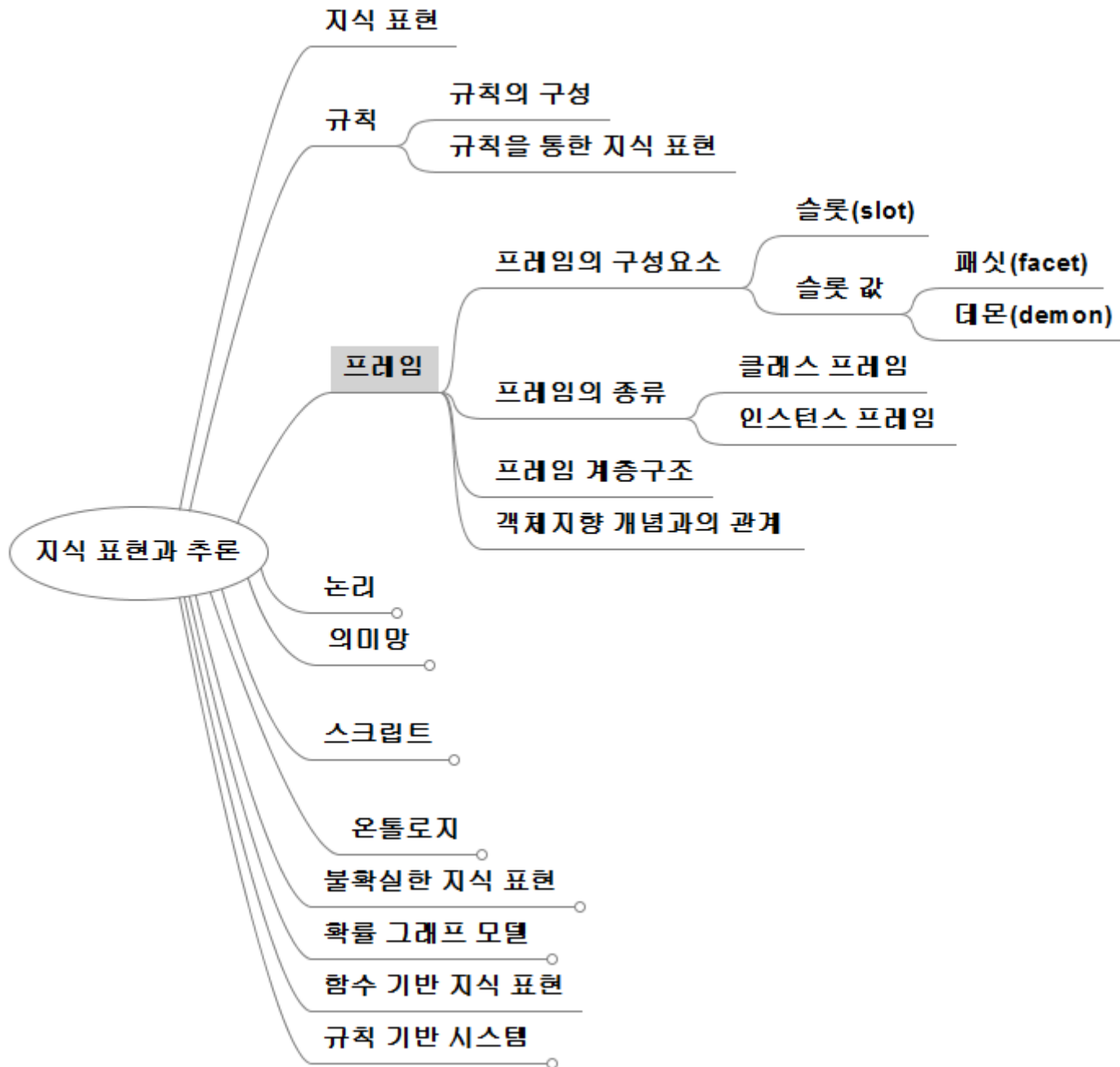
▪ 전략 (strategy)

- 일련의 규칙들로 표현
- 이전 단계의 판정 결과에 따라 다음 단계에 고려할 규칙이 결정
 - IF 차가 멈추었다
THEN 연료통을 확인한다 AND 단계1을 끝낸다
 - IF 단계1이 끝났다 AND 연료통은 충분히 찼다
THEN 배터리를 확인한다 AND 단계2를 끝낸다

▪ 휴리스틱 (heuristic)

- 경험적인 지식을 표현하는 것
- 전문가적 견해는 최적을 항상 보장하는 것이 아니고 일반적으로 바람직한 것을 표현
 - IF 시료가 액체이다 AND 시료의 pH가 6미만이다 AND 냄새가 시큼하다
THEN 시료는 아세트산이다





3. 프레임

❖ 프레임(frame)

- 민스키(M. Minsky, 1927~2016)가 제안한 지식표현 방법
- 특정 객체 또는 개념에 대한 전형적인 지식을 슬롯(slot)의 집합으로 표현하는 것
- 예. 컴퓨터를 표현한 프레임

frame-name	Computer	
frame-type	Class	
CPU	default	Intel
	data-type	string
	require	Intel AMD ARM SPARC
OS	default	Windows
	data-type	string
memory	data-type	integer
warranty	default	3years
HDD	default	1TB
price	data-type	integer
stock	default	in-stock

(frame

(frame-name Computer)

(frame-type class)

(CPU (default Intel)

(data-type string)

(require (Intel AMD ARM SPARC)))

(OS (default Windows)

(data-type string)

(memory (data-type integer))

(warranty (default 3years)

(HDD (default 1TB))

(price (data-type integer))

(stock (default in-stock)))

프레임

❖ 프레임의 구성요소

▪ 슬롯(slot)

- 객체의 **속성**(attribute)을 기술하는 것
- **슬롯 이름**(slot name)과 **슬롯 값**(slot value)으로 구성
 - 슬롯 이름: 속성 이름
 - 슬롯 값: 속성의 값
- **슬롯 값(slot value)**
 - 복수 개의 **패싯**(facet)과 **데몬**(demon)으로 구성

프레임

❖ 프레임의 구성요소 – Cont.

▪ 패싯 (facet)

- '측면' 또는 '양상'을 의미
- 속성에 대한 추가적인 정보를 지정하기 위해 사용
- 패싯 이름과 패싯 값의 쌍으로 구성

• 패싯 이름

- **value** : 속성값 (수, 문자열, 다른 프레임의 포인터 등)
- **data-type** : 속성값의 자료형
- **default** : 디폴트값(속성값이 주어지지 않을 때 사용되는 초기값)
- **require** : 슬롯에 들어갈 수 있는 값이 만족해야 할 제약조건

CPU	default	Intel
	data-type	string
	require	Intel AMD ARM SPARC
memory	data-type	integer

프레임

❖ 프레임의 구성요소 – Cont.

▪ 데몬(demon)

- 지정된 조건을 만족할 때 실행할 절차적 지식(procedure)을 기술
- 슬롯 값으로 데몬 실행조건과 데몬 이름의 쌍

• 데몬의 실행조건의 예

- **if_needed** : 슬롯 값을 알아야 할 때(즉, 사용하려고 할 때)
- **if_added** : 슬롯 값이 추가될 때
- **if_removed** : 슬롯 값이 제거될 때
- **if_modified** : 슬롯 값이 수정될 때

HDD	value	512GB
price	if-needed	look-up-the-list
stock	if-needed	ask-for-vendor

↑
데몬 실행조건

↑
데몬 이름

프레임

❖ 프레임의 종류

- **클래스(class) 프레임**
 - 부류(class)에 대한 정보 표현
- **인스턴스(instance) 프레임**
 - 특정 객체에 대한 정보 표현

❖ 프레임 계층구조(hierarchy)

- **상위 프레임**
 - 클래스를 나타내는 프레임
- **하위 프레임**
 - 하위 클래스 프레임 또는 상위 클래스 프레임의 객체
 - 상위 프레임을 **상속**(inheritance) 받음

프레임

- ❖ 프레임 표현의 예 : 컴퓨터
 - 클래스 프레임 Computer

```
(frame
  (frame-name Computer)
  (frame-type class)
  (CPU (default Intel)
    (data-type string)
    (require (Intel AMD ARM SPARC)))
  (OS (default Windows)
    (data-type string))
  (memory (data-type integer))
  (warranty (default 3years))
  (HDD (default 1TB))
  (price (data-type integer))
  (stock (default in-stock)))
```

frame-name	Computer	
frame-type	Class	
CPU	default	Intel
	data-type	string
	require	Intel AMD ARM SPARC
OS	default	Windows
	data-type	string
memory	data-type	integer
warranty	default	3years
HDD	default	1TB
price	data-type	integer
stock	default	in-stock

프레임

❖ 프레임 표현의 예 : 컴퓨터

▪ 인스턴스 프레임 Ultra-Slim-Notebook

(frame

(**frame-name** Ultra-Slim-Notebook)
 (**frame-type** **instance** (class Computer))
 (**CPU** (**value** ARM))
 (**OS** (**value** Android))
 (**memory** (**value** 4G))
 (**HDD** (**value** 512GB))
 (**price** (**if-needed** look-up-the-list))
 (**stock** (**if-needed** ask-for-vendor)))

(**warranty** 3years) ?

frame-name	Computer	
frame-type	Class	
CPU	default	Intel
	data-type	string
	require	Intel AMD ARM SPARC
OS	default	Windows
	data-type	string
memory	data-type	integer
warranty	default	3years
HDD	default	1TB
price	data-type	integer
stock	default	in-stock

frame-name	Ultra-Slim-Notebook	
frame-type	instance (class Computer)	
CPU	value	ARM
OS	value	Android
memory	value	4G
warranty	value	3years
HDD	value	512GB
price	if-needed	look-up-the-list
stock	if-needed	ask-for-vendor

← 데몬

프레임

❖ 프레임과 규칙을 결합한 지식 표현

- 프레임은 특정 개념이나 대상에 대한 속성들 표현
 - 관련된 속성들을 하나의 덩어리로 관리
- 규칙을 사용하여 조건적인 지식 표현
 - 데몬에 규칙 사용
 - 또는 규칙의 조건부나 결론부에서 프레임 사용
- 대부분의 규칙기반 시스템에서 객체(object) 개념 사용
 - 객체의 표현에 프레임 사용 가능

프레임

❖ 프레임 vs 클래스와 객체

▪ 클래스와 객체

- 소프트웨어 개발에 있어서 **모듈화, 재사용성** 및 **유지보수의 용이성**을 고려한 프로그래밍 개념
- **정보은닉** 등 **정보 접근에 대한 제한 메커니즘**

▪ 프레임

- 사람이 특정 **대상**에 대해 갖는 **지식의 표현**을 목표
- 슬롯의 특정 상황에 따라 **자동으로 호출되는 데몬** 개념

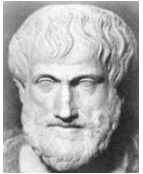


4. 논리

❖ 논리(論理, logic)

- 말로 표현된 **문장**들에 대한 타당한 추론을 위해, **기호**를 사용하여 문장들을 **표현**하고 기호의 **조작**을 통해 문장들의 **참 또는 거짓**을 **판정**하는 분야

❖ 논리학의 역사



- **아리스토텔레스**(Aristotle, BC384-BC322)
 - 기호의 대수적 조작을 통해 추론을 하는 **삼단 논법**(syllogism) 도입

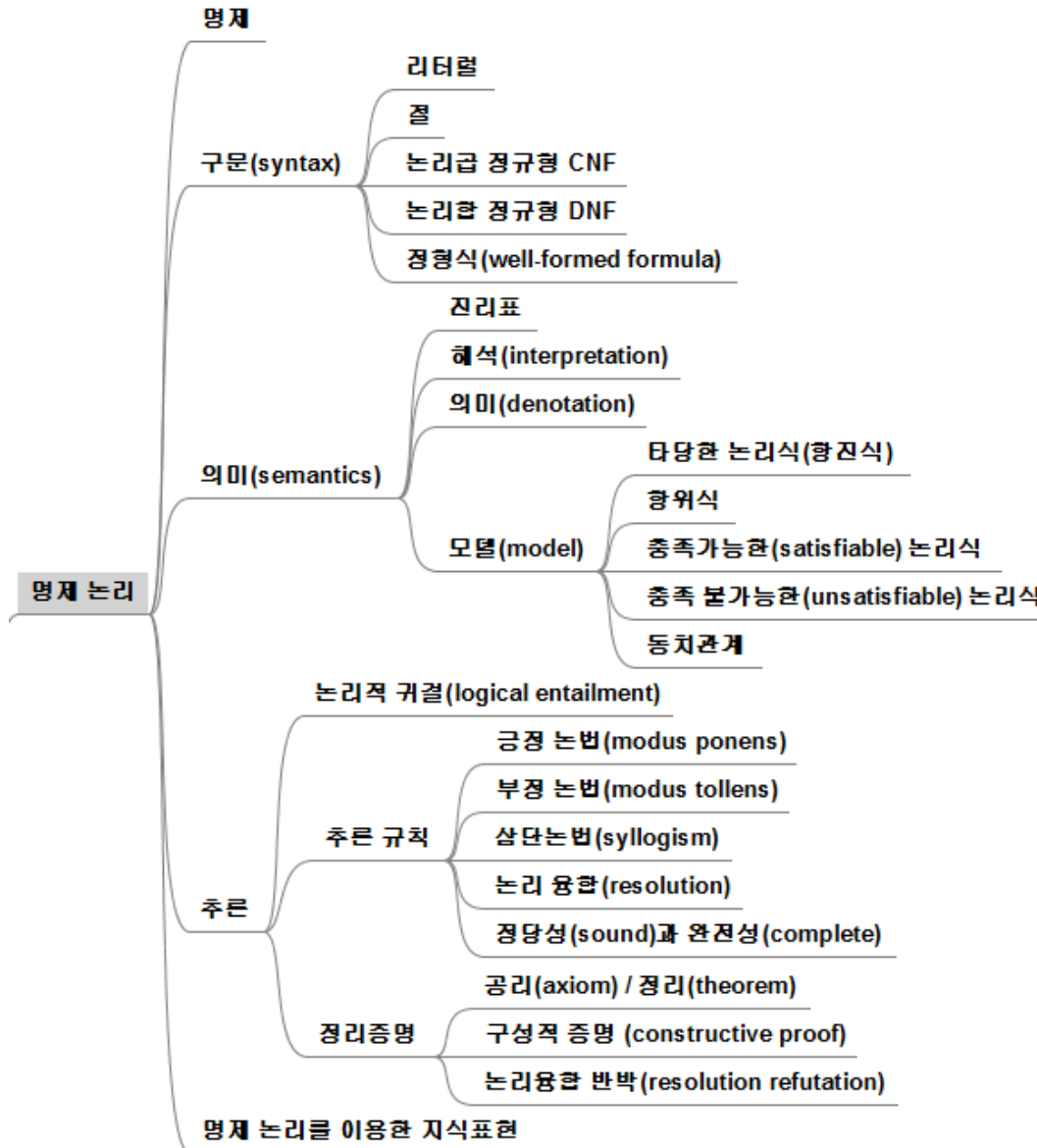


- **부울**(George Boole, 1815-1864)
 - **명제 논리**(propositional logic)의 이론적 기초 확립



- **프리게**(Gottlob Frege, 1848-1925)
 - **술어 논리**(predicate logic)의 이론적 기초를 확립

4.1 명제 논리



4.1 명제 논리

❖ 명제 논리 (propositional logic)

- 명제(命題, proposition)
 - 참, 거짓을 분명하게 판정할 수 있는 문장
 - 아리스토텔레스는 플라톤의 제자이다. (명제)
 - $1+1 = 3$. (명제)
 - 일어나서 아침 먹자. (명제 아님)
- 명제를 **P, Q**등과 같은 **기호로 표현**
- 명제 기호의 **진리값**(truth value)을 사용하여 명제들에 의해 표현되는 문장들의 진리값 결정
- 문장 자체의 **내용**에 대해서는 **무관심**, 문장의 **진리값**에만 **관심**

명제 논리

- ❖ **기본 명제**(primitive proposition)
 - 하나의 진술(statement)로 이루어진 최소 단위의 명제
- ❖ **복합 명제**(compound proposition)
 - 기본 명제들이 결합되어 만들어진 명제
- ❖ 예.
 - 알렉산더는 아시아를 넘본다 $\Leftrightarrow P$
 - 징기스칸은 유럽을 넘본다 $\Leftrightarrow Q$
 - 알렉산더는 아시아를 넘보고, 징기스칸은 유럽을 넘본다 $\Leftrightarrow P \wedge Q$

명제 논리의 구문

❖ 논리식(logical expression)

- 명제를 기호로 표현한 형식
- 명제기호, 참과 거짓을 나타내는 **T**와 **F**, 명제 기호를 연결하는 논리기호인 $\neg, \vee, \wedge, \rightarrow, \equiv$ 를 사용하여 구성

표 3.1 논리기호

논리기호	이름	논리식	의미
\neg	부정(negation)	$\neg P$	P 가 아님
\vee	논리합(disjunction)	$P \vee Q$	P 또는 Q
\wedge	논리곱(conjunction)	$P \wedge Q$	P 그리고 Q
\rightarrow	함의(implication)	$P \rightarrow Q$	P 이면 Q
\equiv	동치(equivalence)	$P \equiv Q$	$(P \rightarrow Q) \wedge (Q \rightarrow P)$

명제 논리의 구문

❖ 리터럴(literal)

- 명제 기호 P 와 명제 기호의 $\neg P$ 부정

❖ 절(clause)

- 리터럴들이 논리합으로만 연결되거나 논리곱으로 연결된 논리식

$$P \vee Q \vee \neg R \quad (\text{논리합 절})$$

$$P \wedge Q \wedge \neg R \quad (\text{논리곱 절})$$

❖ 논리곱 정규형 (conjunctive normal form, CNF)

- 논리합 절들이 논리곱으로 연결되어 있는 논리식

$$(P \vee Q \vee \neg R) \wedge (\neg Q \vee R \vee S) \wedge (P \vee R \vee S)$$

❖ 논리합 정규형 (disjunctive normal form, DNF)

- 논리곱 절들이 논리합으로 연결되어 있는 논리식

$$(P \wedge Q \wedge \neg R) \vee (\neg Q \wedge R \wedge S) \vee (P \wedge R \wedge S)$$

명제 논리의 구문

❖ 정형식(well-formed formula, wff)

- 논리에서 문법에 맞는 논리식

- 명제 논리에 대한 정형식

(1) 진리값 T, F와 명제 기호들 P, Q, R, \dots 은 정형식이다.

(2) p 와 q 가 정형식이면, 논리 기호를 사용하여 구성되는 논리식 $\neg p$, $p \vee q$,
 $p \wedge q$, $p \rightarrow q$, $p \equiv q$ 도 정형식이다.

(3) (1)과 (2)에 의해 정의되는 논리식만 정형식이다.

$$(P \wedge Q) \rightarrow \neg P$$

$$P \rightarrow \neg P$$

$$P \vee P \rightarrow P$$

$$(P \rightarrow Q) \rightarrow (\neg Q \rightarrow \neg P)$$

명제 논리의 의미

❖ 진리표(truth table)

- 논리기호에 따라 참, 거짓 값을 결합하는 방법을 나타낸 표

표 3.2 진리표

P	Q	$\neg P$	$P \vee Q$	$P \wedge Q$	$P \rightarrow Q$	$P \equiv Q$
F	F	T	F	F	T	T
F	T	T	T	F	T	F
T	F	F	T	F	F	F
T	T	F	T	T	T	T

❖ 논리식의 해석(interpretation)

- 논리식의 진리값을 결정하는 것

- $P = T, Q = F, R = T$
 - $(P \vee \neg Q) \wedge (Q \vee \neg R) = F$

- 우선 각 명제기호의 진리값 결정 필요

- 명제 기호에 "명제"를 대응시키고, 해당 명제의 진리값을 결정
 - 예. $P \Rightarrow$ "토마토는 과일이다", $P = F$

- 대응된 명제를 명제 기호의 외연(外延) 또는 의미(denotation)라 함

명제 논리의 의미

❖ 논리식의 **모델**(model)

- 논리식의 명제기호에 참값(T) 또는 거짓값(F)을 할당한 것

- $(P \vee \neg Q) \wedge (Q \rightarrow \neg R) : P = T, Q = F, R = T$

- 모델이 주어지면, **진리표**를 사용하여 논리식의 진리값 결정, 즉 **해석** 가능

P	Q	$\neg P$	$P \vee Q$	$P \wedge Q$	$P \rightarrow Q$	$P \equiv Q$
F	F	T	F	F	T	T
F	T	T	T	F	T	F
T	F	F	T	F	F	F
T	T	F	T	T	T	T

- n 개의 명제기호가 논리식에 사용된다면, 각각 T 또는 F값을 가질 수 있기 때문에, 총 2^n 개의 모델이 존재

명제 논리의 의미

❖ 타당한 논리식 (valid logical expression)

- 모든 가능한 모델에 대해서 항상 참(T)인 논리식
- 항진식(恒眞式, tautology)

• 예. $P \vee \neg P$

$$P = T \text{인 경우} : P \vee \neg P = T$$

$$P = F \text{인 경우} : P \vee \neg P = T$$

❖ 항위식(恒僞式, contradiction)

- 모든 가능한 모델에 대해서 항상 거짓이 되는 논리식

• 예. $P \wedge \neg P$

$$P = T \text{인 경우} : P \wedge \neg P = F$$

$$P = F \text{인 경우} : P \wedge \neg P = F$$

명제 논리의 의미

❖ 충족가능한(satisfiable) 논리식

- 참으로 만들 수 있는 모델이 하나라도 있는 논리식

- 예. $(P \vee \neg Q) \wedge (Q \vee \neg R)$

$$P = \text{T}, Q = \text{T}, R = \text{F}$$

❖ 충족불가능한(unsatisfiable) 논리식

- 참으로 만들 수 있는 모델이 전혀 없는 논리식

- 항위식인 논리식

- 예. $P \wedge \neg P$

명제 논리의 의미

- ❖ 동치관계(equivalence relation)의 논리식
 - 어떠한 모델에 대해서도 같은 값을 갖는 두 논리식

$$(1) \neg(\neg p) \equiv p$$

$$(2) p \vee F \equiv p, p \wedge T \equiv p$$

$$(3) p \vee \neg p \equiv T, p \wedge \neg p \equiv F$$

$$(4) \neg(p \wedge q) \equiv \neg p \vee \neg q, \neg(p \vee q) \equiv \neg p \wedge \neg q$$

$$(5) p \rightarrow q \equiv \neg p \vee q$$

$$(6) p \vee (q \vee r) \equiv (p \vee q) \vee r, p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$$

$$(7) p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r), p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$$

명제 논리의 의미

❖ 동치관계를 이용한 논리식의 변환

- 논리식의 동치관계를 이용하면

임의의 논리식을 논리곱 정규형(CNF)과 같은 정형식으로 변환

$$p \wedge (q \rightarrow r) \rightarrow p \wedge q$$

$$\equiv \neg(p \wedge (\neg q \vee r)) \vee (p \wedge q)$$

$$\equiv (\neg p \vee \neg(\neg q \vee r)) \vee (p \wedge q)$$

$$\equiv (\neg p \vee (q \wedge \neg r)) \vee (p \wedge q)$$

$$\equiv ((\neg p \vee q) \wedge (\neg p \vee \neg r)) \vee (p \wedge q)$$

$$\equiv ((\neg p \vee q) \vee (p \wedge q)) \wedge ((\neg p \vee \neg r) \vee (p \wedge q))$$

$$\equiv ((\neg p \vee q) \vee p) \wedge ((\neg p \vee q) \vee q) \wedge ((\neg p \vee \neg r) \vee p) \wedge ((\neg p \vee \neg r) \vee q)$$

$$\equiv (T \vee q) \wedge (\neg p \vee q) \wedge (T \vee \neg r) \wedge (\neg p \vee \neg r \vee q)$$

$$\equiv (\neg p \vee q) \wedge (\neg p \vee q \vee \neg r)$$

명제 논리의 의미

❖ 논리적 귀결(logical entailment)

- Δ : 정형식(wff)의 집합. $\Delta = \{P, P \rightarrow Q\}$
 ω : 정형식. $\omega = Q$
- Δ 에 있는 모든 정형식을 참(T)으로 만드는 모델이, ω 를 참(T)으로 만든다
= Δ 는 ω 를 논리적으로 귀결한다(logically entail)
= ω 는 Δ 를 논리적으로 따른다(logically follow)
= ω 는 Δ 의 논리적 결론(logical consequence)이다

- Δ 가 참이면, ω 도 참이다

▪ 추론

- 참으로 알려진 Δ 로 부터, 알려지지 않은 참인 ω 를 찾는 것

명제 논리의 추론

❖ 추론(推論, inference)

▪ 귀납적 추론(inductive inference)

- 관측된 복수의 사실들을 일반화(generalization)하여 일반적인 패턴 또는 명제를 도출하는 것

▪ 연역적 추론(deductive inference)

- 참인 사실들 또는 명제들로 부터 새로운 참인 사실 또는 명제를 도출하는 것

▪ 논리에서의 추론

- 함의(\rightarrow)의 논리적 관계를 이용하여 새로운 논리식을 유도해 내는 것
- 함의 $p \rightarrow q$
 - p : 전제(premise)
 - q : 결론(conclusion, consequence)

명제 논리의 추론

❖ 추론규칙(inference rule)

- 참인 논리식들이 논리적으로 귀결하는 새로운 논리식을 만들어내는 기계적으로 적용되는 규칙

- 긍정 논법 (modus ponens)

$p \rightarrow q$	새이다 \rightarrow 날 수 있다	
p	새이다	
q	날 수 있다	$p \rightarrow q, p \vdash q$

- 부정 논법 (modus tollens)

$p \rightarrow q$	새이다 \rightarrow 날 수 있다	
$\neg q$	날 수 없다	
$\neg p$	새가 아니다	$p \rightarrow q, \neg q \vdash \neg p$

- 삼단 논법 (syllogism)

$p \rightarrow q$	새이다 \rightarrow 날개가 있다	
$q \rightarrow r$	날개가 있다 \rightarrow 날 수 있다	
$p \rightarrow r$	새이다 \rightarrow 날 수 있다	$p \rightarrow q, q \rightarrow r \vdash p \rightarrow r$

명제 논리의 추론

❖ 추론규칙 – cont.

▪ 논리 융합 (resolution)

- 일반화된 추론규칙
 - 긍정 논법, 부정 논법, 삼단 논법의 규칙을 포함한 추론 규칙
- 두 개의 논리합절이 같은 기호의 긍정과 부정의 리터럴을 서로 포함하고 있을 때, 해당 리터럴들을 제외한 나머지 리터럴들의 논리합절을 만들어 내는 것

$$P \vee q, \neg P \vee r \vdash q \vee r$$

↑
논리융합식(resolvent)

명제 논리의 추론

❖ 추론 규칙의 정당성과 완전성

▪ 추론 규칙의 **정당성** (sound)

- 추론 규칙에 의해 생성된 논리식은 주어진 논리식들이 논리적으로 귀결하는 것이다.
- 즉, 추론 규칙이 만들어 낸 것은 항상 참이다.

$$\Delta \vdash \omega \rightarrow \Delta \models \omega$$

▪ 추론 규칙의 **완전성** (complete)

- 주어진 논리식들이 논리적으로 귀결하는 것들은 추론 규칙이 찾아 낼 수 있다.

명제 논리의 추론

❖ 정리증명(theorem proving)

- **공리(axiom)**

- 추론을 할 때, 참인 것으로 주어지는 논리식

- **정리(theorem)**

- 공리들에 추론 규칙을 적용하여 얻어지는 논리식

- **정리 증명**

- 공리들을 사용하여 정리가 참인 것을 보이는 것

- **구성적 증명(constructive proof)**

- 공리들에 추론 규칙들을 적용하여 증명을 만들어 보이는 증명

- **논리융합 반박(resolution refutation)**

- 증명할 정리를 부정(negation)한 다음, 논리융합 방법을 적용하여 모순이 발생하는 것을 보여서, 정리가 참임을 증명하는 방법

명제 논리의 추론

❖ 논리융합 반박을 이용한 정리증명의 예

▪ 공리

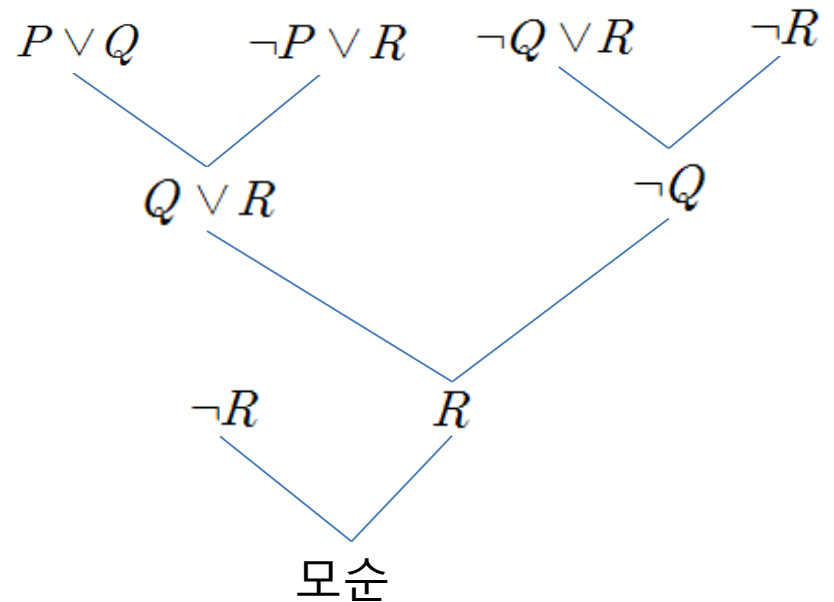
$$P \vee Q$$

$$P \rightarrow R \quad \neg P \vee R$$

$$Q \rightarrow R \quad \neg Q \vee R$$

▪ 정리

$$R \quad \Rightarrow \quad \neg R$$



명제 논리의 지식표현

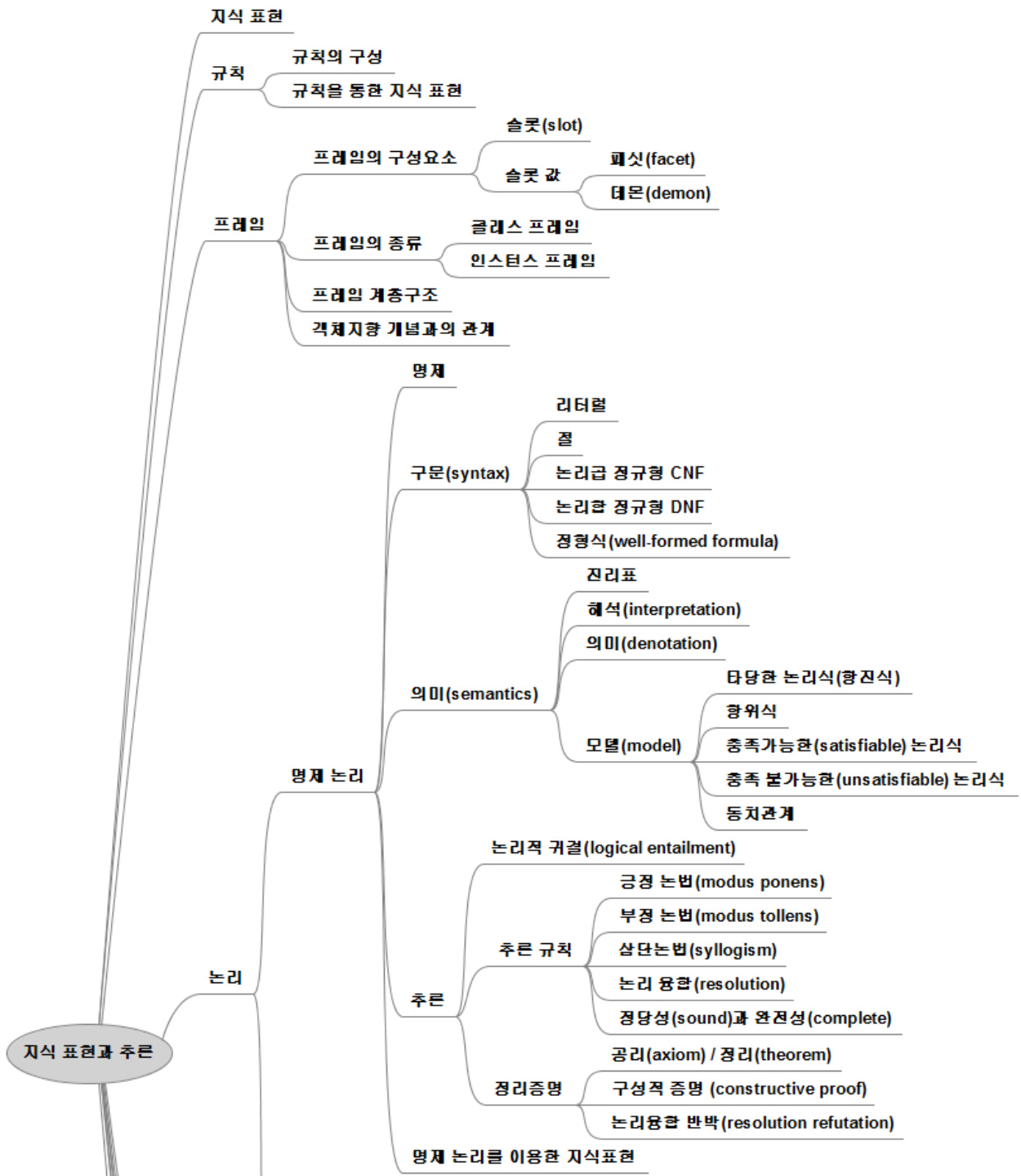
❖ 명제 논리를 이용한 지식 표현

- 문장으로 표현된 지식으로부터 기본 명제들을 추출
- 각 명제에 대해 명제기호 부여
- 기본 명제들의 논리적 연결 관계를 참고하여 대응되는 명제 기호들을 논리기호로 연결하여 논리식 구성

❖ 명제 논리로 표현된 지식에 대한 추론

- 명제 기호가 나타내는 명제의 의미와는 무관
- 대수적인 기호 연산을 통해서 추론 수행

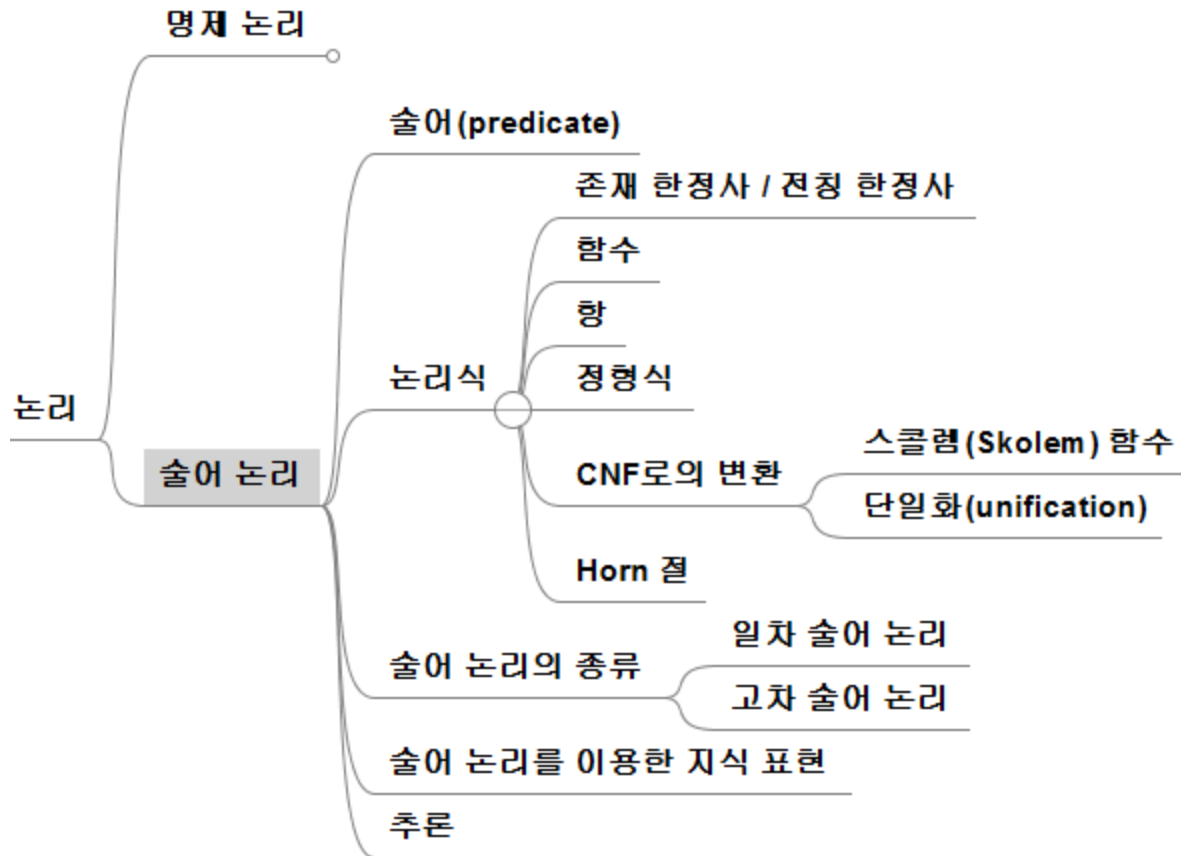




지식표현과 추론

Part II

충북대학교 소프트웨어학과
이건명



4.2 술어 논리

❖ 술어 논리(predicate logic)

- 명제의 내용을 다루기 위해 **변수, 함수** 등을 도입하고 이들의 값에 따라 참, 거짓이 결정되도록 명제 논리를 확장한 논리
- **술어**(述語, predicate)
 - 문장의 '주어+서술어'형태에서 **서술어**에 해당
 - 대상의 **속성**이나 **대상 간의 관계**를 기술하는 기호
 - **참(T)** 또는 **거짓(F)** 값을 갖는 **함수**
 - 예.
 - **Student**(John)
 - **Friend**(John, Mary)

술어 논리의 구문

❖ 술어 논리

- **존재 한정사**(existential quantifier) \exists 와 **전칭 한정사**(universal quantifier)

\forall 사용

- 변수의 범위를 고려한 지식을 표현
- $\exists x \text{ Friend}(\text{John}, x)$
 - 'John은 친구가 한 명은 있다'
- $\forall x \exists y \text{ Friend}(x, y)$
 - '누구나 친구가 한 명은 있다'

술어 논리의 구문

❖ 함수(function)

- 주어진 인자에 대해서 참, 거짓 값이 아닌 일반적인 값을 반환
- 술어나 다른 함수의 인자로 사용

❖ 항(term)

- 함수의 인자가 될 수 있는 것
- 항이 될 수 있는 것 : 개체상수, 변수, 함수

(1) 개체상수, 변수는 항이다.

(2) t_1, t_2, \dots, t_n 이 모두 항이고, f 가 n 개의 인자를 갖는 함수 기호일 때,

$f(t_1, t_2, \dots, t_n)$ 은 항이다.

(3) (1)과 (2)에 의해 만들어질 수 있는 것만 항이다.

술어 논리의 구문

❖ 술어 논리식에 대한 정형식

- (1) t_1, t_2, \dots, t_n 이 모두 항이고, p 가 n 개의 인자를 갖는 술어 기호일 때, $p(t_1, t_2, \dots, t_n)$ 은 정형식이다.
- (2) p 와 q 가 정형식이면, 논리 기호를 사용하여 구성되는 논리식 $\neg p$, $p \vee q$, $p \wedge q$, $p \rightarrow q$, $p \equiv q$ 도 정형식이다.
- (3) $p(x)$ 가 정형식이고, x 가 변수일 때, $\forall x p(x)$, $\exists x p(x)$ 는 정형식이다.
- (4) (1), (2), (3)에 의해 만들어질 수 있는 것만 술어 논리의 정형식이다.

$$\forall x \forall y \text{Horse}(x) \wedge \text{Dog}(y) \rightarrow \text{Faster}(x,y)$$

$$\exists y \text{Greyhound}(y) \wedge (\forall z \text{Rabbit}(z) \rightarrow \text{Faster}(y,z))$$

$$\text{Horse}(\text{Harry})$$

$$\text{Rabbit}(\text{Ralph})$$

$$\forall y \text{Greyhound}(y) \rightarrow \text{Dog}(y)$$

$$\forall x \forall y \forall z \text{Faster}(x,y) \wedge \text{Faster}(y,z) \rightarrow \text{Faster}(x,z)$$

술어 논리의 종류

- ❖ 일차 술어논리(first-order predicate logic, **FOL**)
 - 변수에만 전칭 한정사와 존재 한정사를 쓸 수 있도록 한 술어논리

- ❖ 고차 술어논리(high-order predicate logic)
 - 변수뿐만 아니라 함수, 술어기호 등에 대해서 전칭 한정사와 존재 한정사를 쓸 수 있도록 한 술어논리
 - $\exists S S(x)$
 - $\exists g \forall x (f(x) = h(g(x)))$

술어 논리의 지식표현

❖ 술어 논리를 이용한 지식 표현

- (a) Whoever can read is literate. (읽을 수 있으면 문맹이 아니다)
- (b) Monkeys are not literate. (원숭이는 문맹이다)
- (c) Some monkeys are intelligent. (어떤 원숭이는 지능적이다)
- (d) Some who are intelligent cannot read. (지능적이어도 문맹일 수 있다) ← 증명

$$(a) \forall x[CanRead(x) \rightarrow Literate(x)]$$

$$(b) \forall x[Monkey(x) \rightarrow \neg Literate(x)]$$

$$(c) \exists x[Monkey(x) \wedge Intelligent(x)]$$

$$(d) \exists x[Intelligent(x) \wedge \neg CanRead(x)]$$

술어 논리의 추론

❖ 술어 논리식의 CNF로의 변환 과정

1. 전칭 한정사와 존재 한정사를 논리식의 맨 앞으로 끌어내는 변환
2. 전칭 한정사에 결합된 변수
 - 임의의 값 허용
3. 존재 한정사에 결합된 변수
 - 대응되는 술어 기호를 참(T)으로 만드는 값을 변수에 대응시킴
 - **스콜렘 함수(Skolem function)**
 - 존재 한정사에 결합된 변수를 해당 술어의 전칭 한정사에 결합된 다른 변수들의 새로운 함수로 대체
 - $\forall x \exists y [P(x) \wedge Q(x, y)] \Rightarrow \forall x [P(x) \wedge Q(x, s(x))]$
 - $s(x)$
 - » $Q(x, s(x))$ 를 어떤 x 에 대해서도 참으로 만드는 마법의 함수(magic function)

술어 논리의 추론

❖ 단일화(unification) 과정

- 논리융합(resolution)을 적용할 때는 대응되는 리터럴이 같아지도록, 변수의 값을 맞춰주는 과정

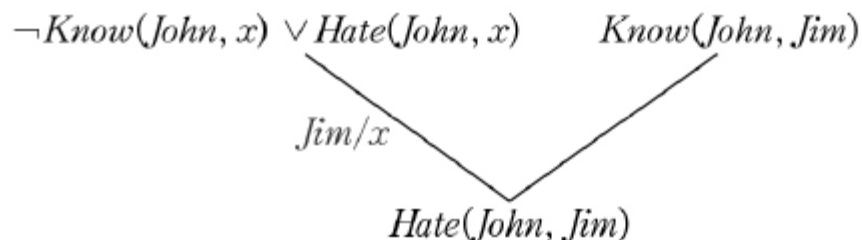


그림 3.5 단일화 과정의 예

x 를 Jim 으로 대체하면 논리융합에 의해 $Hate(John, Jim)$ 이 도출됨.

술어 논리의 추론

❖ 술어 논리로 지식의 증명

$$(a) \forall x [CanRead(x) \rightarrow Literate(x)]$$

$$(b) \forall x [Monkey(x) \rightarrow \neg Literate(x)]$$

$$(c) \exists x [Monkey(x) \wedge Intelligent(x)]$$

$$(d) \exists x [Intelligent(x) \wedge \neg CanRead(x)] \leftarrow \text{증명}$$

▪ 논리곱 형태로 변환

$$(1) \neg CanRead(x) \vee Literate(x)$$

$$(2) \neg Monkey(x) \vee \neg Literate(x)$$

$$(3) Monkey(A)$$

$$(4) Intelligent(A)$$

$$(5) \neg Intelligent(x) \vee CanRead(x)$$



(d)를 논리융합 논박을 이용하여 증명하기 부정

▪ (1)과 (5)

$$(6) \neg Intelligent(x) \vee Literate(x)$$

▪ (2)과 (6)

$$(7) \neg Intelligent(x) \vee \neg Monkey(x)$$

▪ (3)과 (7)

$$(8) \neg Intelligent(A)$$

▪ (4)과 (8)

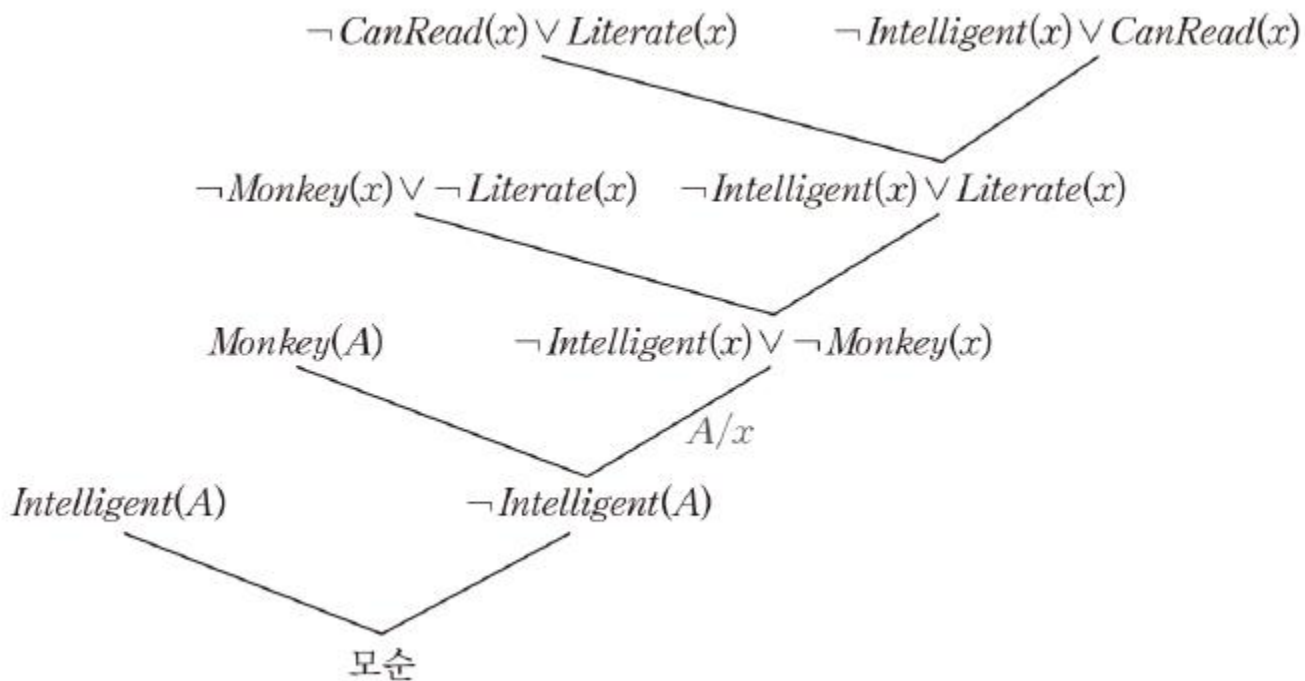
$$(9) Intelligent(A) \wedge \neg Intelligent(A) \equiv nil$$

(d)를 부정하여 모순이 발생하므로, (d)가 참임

술어 논리의 추론

❖ 술어 논리로 지식의 증명

- (1) $\neg CanRead(x) \vee Literate(x)$
- (2) $\neg Monkey(x) \vee \neg Literate(x)$
- (3) $Monkey(A)$
- (4) $Intelligent(A)$
- (5) $\neg Intelligent(x) \vee CanRead(x)$



논리 프로그래밍 언어

❖ Horn 절 (Horn clause)

- 논리식을 논리합의 형태로 표현할 때, $\neg A(x) \vee \neg B(x) \vee C(x)$ 와 같이 긍정인 리터럴을 최대 하나만 허용

❖ Prolog

- Horn 절만 허용하는 논리 프로그래밍 언어

father(noah, shem).

father(noah, ham).

father(shem, elam).

father(shem, arphaxad).

father(arphaxad, caina).

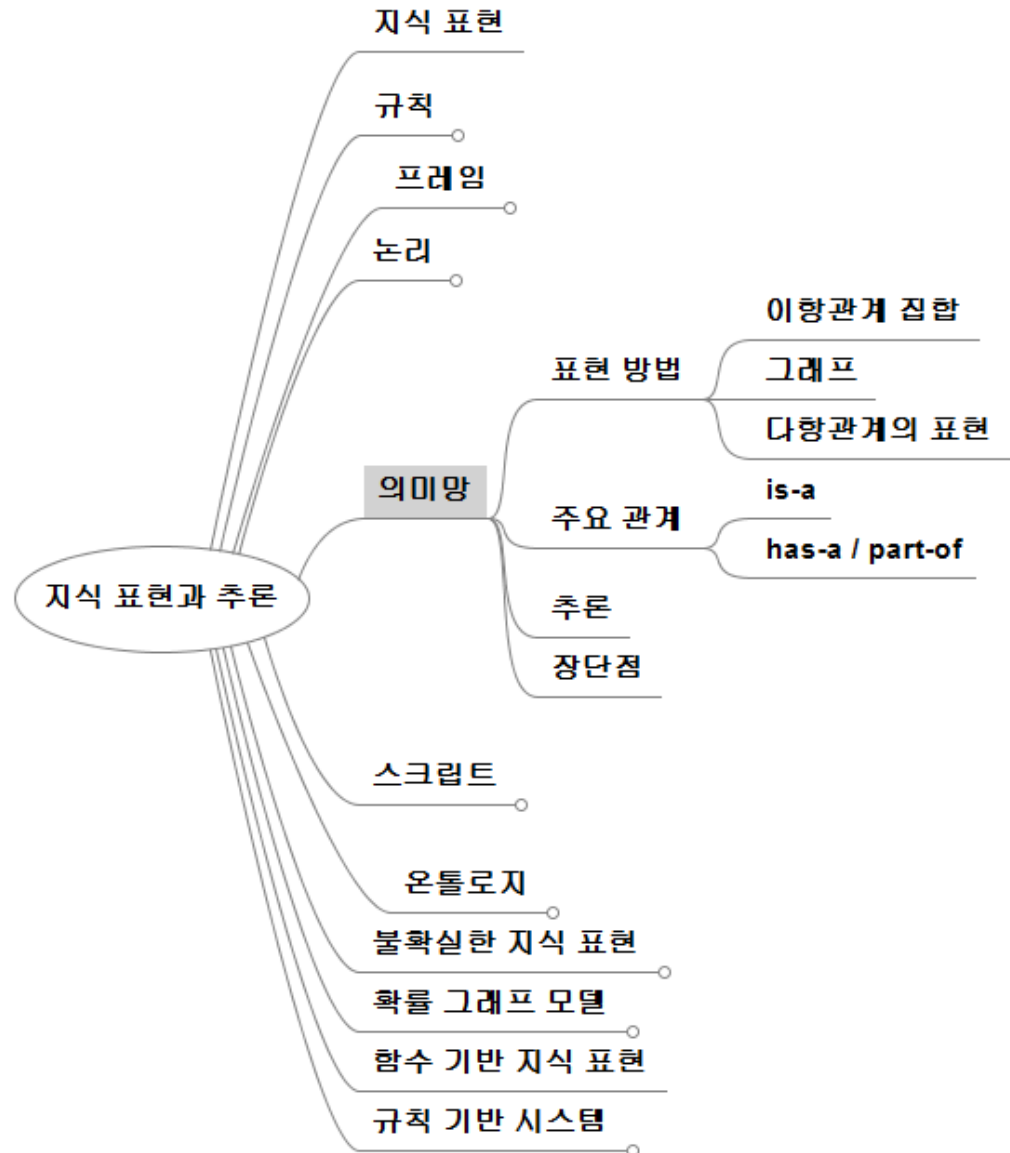
grandfather(X,Y) :- father(X,Z), father(Z,Y).

:- grandfather(X,Y).

- 백트래킹(backtracking)을 이용하여 실행



5. 의미망



의미망

❖ 의미망(semantic network)

- 지식을 **이항 관계**(binary relation)의 집합으로 표현
- 노드와 방향성 간선으로 구성되는 **그래프**를 사용해 지식 표현
 - **노드 (node)**
 - 대상, 개념, 행위, 상태, 사건
 - **간선 (edge)**
 - 관계가 있는 노드를 연결
 - 관계에 따른 **방향**
 - 관계의 의미를 나타내는 **라벨** 부여

의미망

❖ 의미망의 예

is-a(조류, 동물),
 has-a(조류, 깃털),
 can(조류, 날기),
 cannot(펭귄, 날기),
 do(동물, 호흡),
 do(동물, 이동),
 do(호랑이, 육식),
 do(토끼, 초식),
 can(펭귄, 수영)
 do(포유류, 수유)

is-a(포유류, 동물)
 can(조류, 알 낳기)
 is-a(펭귄, 조류)
 is-a(종달새, 조류)
 has-a(동물, 피부)
 has-a(호랑이, 날카로운 발톱)
 is-a(토끼, 포유류)
 is-a(트위터, 종달새)
 is-a(바니, 토끼)

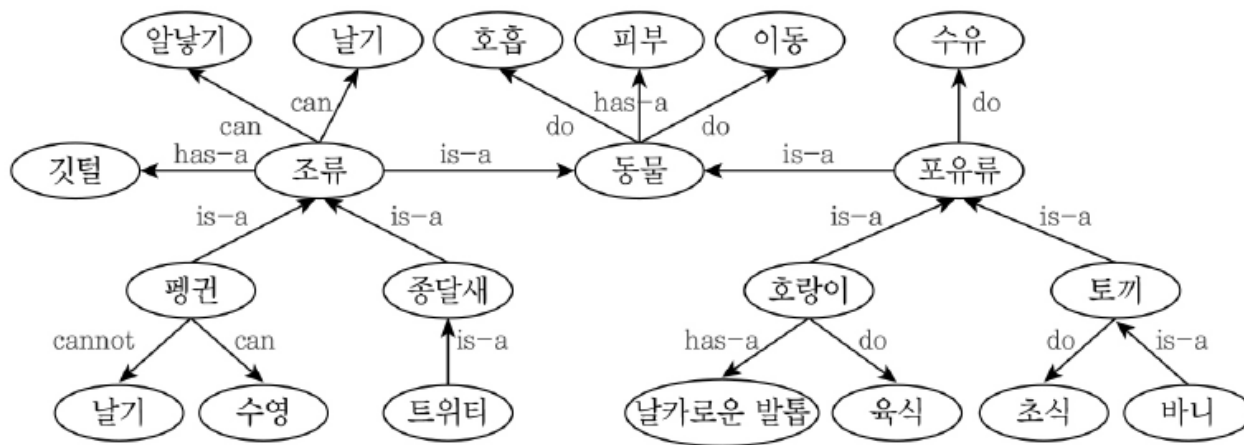


그림 3.7 의미망의 예

의미망

❖ 의미망에서 사용되는 관계(relationship)

▪ is-a

- 상위 클래스와 하위 클래스 관계(예, is-a(조류,동물)), 또는 클래스와 객체의 관계(예, is-a(트위티,종달새))를 나타내어 계층 관계 표현
- 상위 계층의 속성을 상속
- **추이적(transitive) 관계 만족**

$$\text{is-a}(X,Y) \wedge \text{is-a}(Y,Z) \rightarrow \text{is-a}(X,Z).$$

$$\text{is-a}(\text{펭귄},\text{조류}) \wedge \text{is-a}(\text{조류},\text{동물}) \rightarrow \text{is-a}(\text{펭귄},\text{동물})$$

▪ has-a

- 전체-부분 관계
- **part-of**와 역관계
 - has-a(X,Y)이면 part-of(Y,X) 성립
- **추이적 관계 만족**

$$\text{has-a}(X,Y) \wedge \text{has-a}(Y,Z) \rightarrow \text{has-a}(X,Z).$$

$$\text{part-of}(X,Y) \wedge \text{part-of}(Y,Z) \rightarrow \text{part-of}(X,Z).$$

의미망

❖ 다항 관계를 이항 관계로 전개하여 표현한 의미망

- 의미망은 이항 관계만을 표현
- 다항 관계는 "관계(relationship)"을 "객체(object)"로 간주하여 표현
 - 사물화(reification)
 - 예. 길동이는 지난 가을부터 현재까지 고양이를 키우고 있다

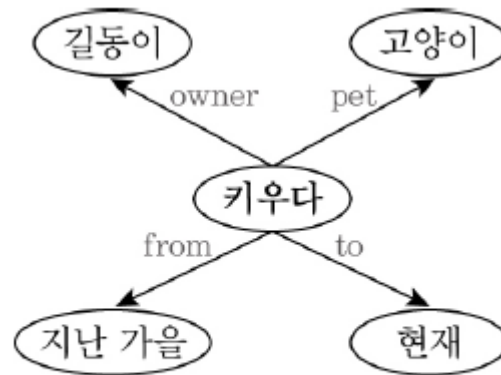
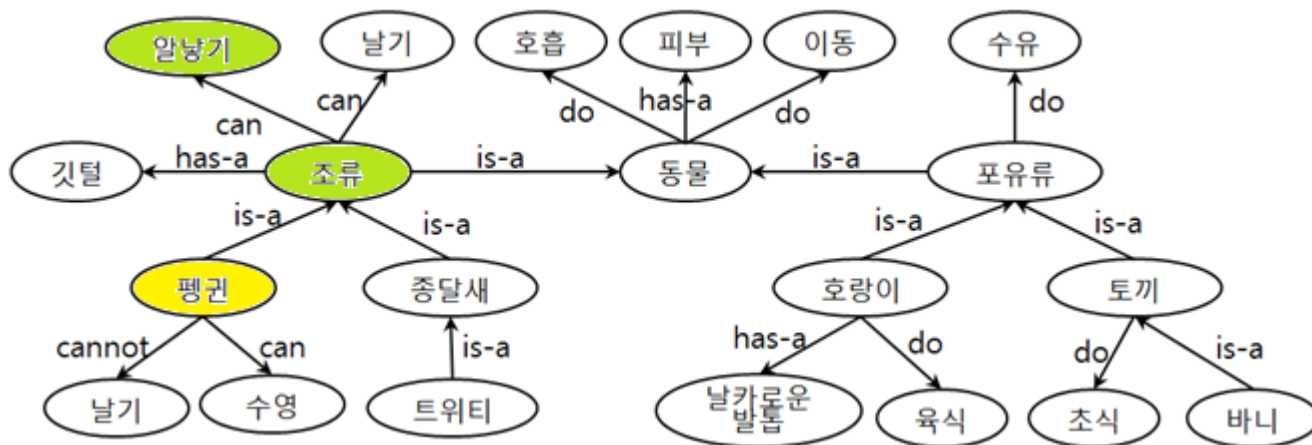


그림 3.8 다항 관계를 이항 관계로 전개하여 표현한 의미망

의미망의 추론

❖ 의미망에서 추론

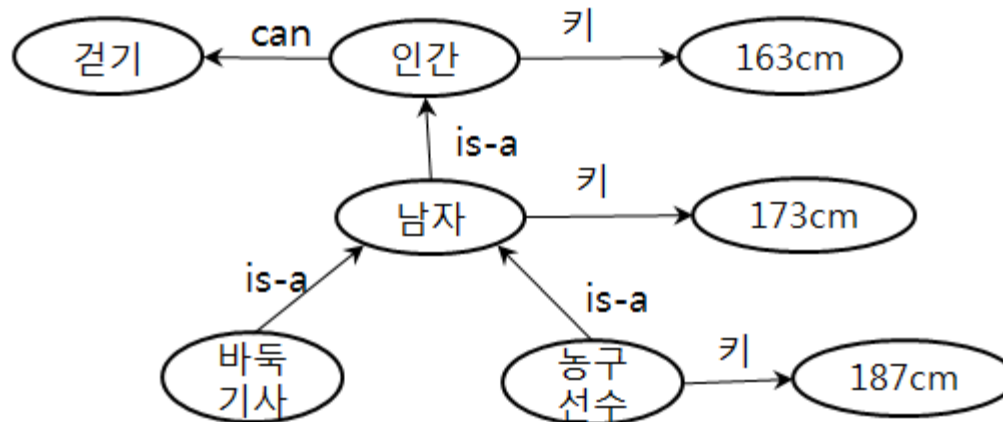
- 상속(inheritance)을 이용
- 질문에 대한 의미망과 지식을 나타내는 의미망을 비교
- 질문 예. '펭귄은 알을 낳는가?'
 - can(펭귄, 알내기)에 해당
 - is-a 관계의 간선을 따라 조류 노드로 이동
 - can(조류, 알내기)가 있으므로, 질문의 답은 참



의미망의 추론

❖ 의미망에서 추론

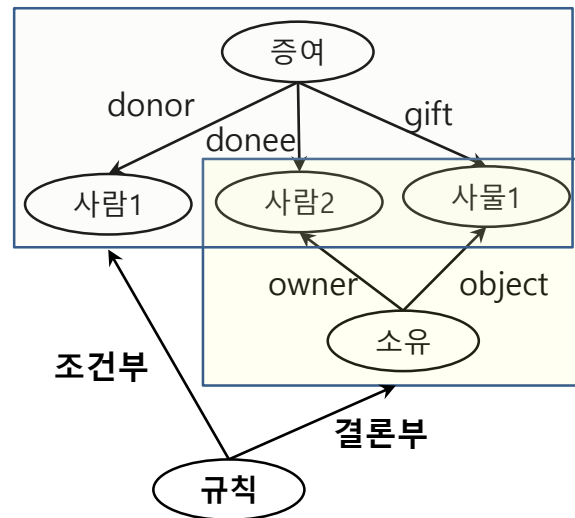
- 디폴트값(default value, 기본값)을 이용한 추론
- 상속 관계 이용
- 예. 바둑 기사의 키는?



의미망의 추론

❖ 의미망에서 추론

- 주어진 지식으로부터 새로운 사실을 이끌어내는 추론도 가능
- 규칙의 의미망 표현
 - 예. 사람1이 사람2에게 사물1을 주면, 사람2는 사물1를 소유하게 된다

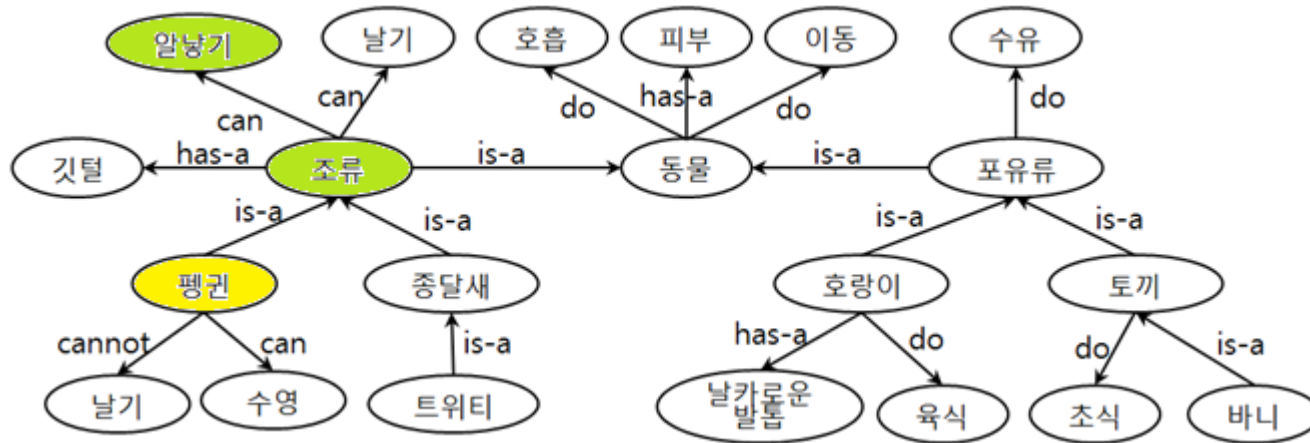


- 주어진 사실: 철수가 영희에게 반지를 주었다
 - » 사람1=철수, 사람2=영희, 사물1=반지
 - » 추론 결과 : 영희가 반지를 가지고 있다

의미망과 프레임

❖ 의미망의 프레임으로의 변환

- 노드별로 프레임 생성
- 노드에서 나가는 간선들을 슬롯(slot)으로 구성



frame-name	펭귄
frame-type	Class
is-a	조류
can	수영
cannot	날기

frame-name	조류
frame-type	Class
is-a	동물
can	날기 알내기
has-a	깃털

의미망

❖ 의미망 표현의 장점

- 지식을 시각적으로 표현할 수 있어, 직관적 이해 용이
- 노드 추가 또는 변경으로 비교적 쉽게 지식의 추가 및 변경 가능
- 개념의 계층관계를 정의하여 속성의 상속 관계 지정 가능
- 복잡한 지식을 구조화하여 표현 가능

❖ 의미망 표현의 단점

- 지식의 양이 많아지면 관리 복잡
- 개념이나 관계를 임의로 정의하기 때문에 통일성이 부족
- 공유나 재사용에 대한 고려 없음
- 논리적 결합 관계나 인과 관계를 기술하려고 하면 and, or, implies와 같은 링크 도입 필요
 - 일관성을 떨어뜨리고 추론과정을 복잡
- 기본적으로 정적인 지식의 표현
 - 추론 과정에서 동적으로 지식의 내용을 바꾸려면 그래프를 동적으로 바꿀 수 있도록 해야 함



6. 스크립트

❖ 스크립트(script)

- 일반적으로 발생할 수 있는 **전형적인 상황**에 대한 **절차적 지식**을 일목요연하게 표현
- 전형적인 상황에서 일어나는 **일련의 사건(event)**을 **시간적 순서를 고려**하여 기술하는 프레임과 같은 구조의 지식 표현
- 구성요소
 - **진입 조건(entry conditions)**
 - 스크립트에 기술된 사건들이 일어나기 전에 만족되어야 하는 전제조건
 - **역할자(roles)**
 - 스크립트에 관련된 사람 및 대상
 - **자산(properties)**
 - 사건 진행에 과정에서 사용되는 객체
 - **트랙(track)**
 - 어떤 스크립트에서 발생할 수 있는 일련의 사건들이 변형된 형태 식별자
 - **장면(scenes)**
 - 실제 일어나는 일련의 사건
 - **결과 조건(results)**
 - 스크립트의 장면에 있는 사건들이 일어난 이후에 만족되는 조건

스크립트

❖ 패스트푸드 식당에서의 상황을 표현한 스크립트.

스크립트 이름	식당
트랙(track)	패스트푸드 식당
역할자(roles)	고객, 점원
자산(properties)	카운터, 쟁반, 음식, 돈, 냅킨, 소금/후추/시럽/빨대
진입조건 (entry conditions)	고객이 배가 고프다. 고객은 돈이 있다.
장면 (scenes)	장면 1: 입장 고객이 식당에 들어선다. 고객은 카운터에서 줄을 서서 기다린다. 고객은 벽에 있는 메뉴를 보고, 주문할 것을 결정한다.
	장면 2: 주문 고객이 점원에게 주문을 한다. 고객은 점원에게 음식값을 지불한다. 점원이 주문한 것을 쟁반에 올려 놓는다.
	장면 3: 식사 고객은 소금/후추/냅킨/빨대 등을 집는다. 고객은 쟁반을 받아들고 빈자리에 가서 앉는다. 고객은 음식을 빨리 먹는다.
	장면 3-1(선택적) : 들고 가기(take-out) 고객이 음식을 가지고 나간다.
	장면 4: 퇴장 고객이 식탁을 치운다. 고객이 출입문 옆 쓰레기통에 버린다. 고객이 식당을 나간다.
결과 조건 (results)	고객은 더 이상 배고프지 않다. 고객의 돈이 줄었다. 고객은 만족스럽다. (선택적) 고객은 만족스럽지 못하다. (선택적)

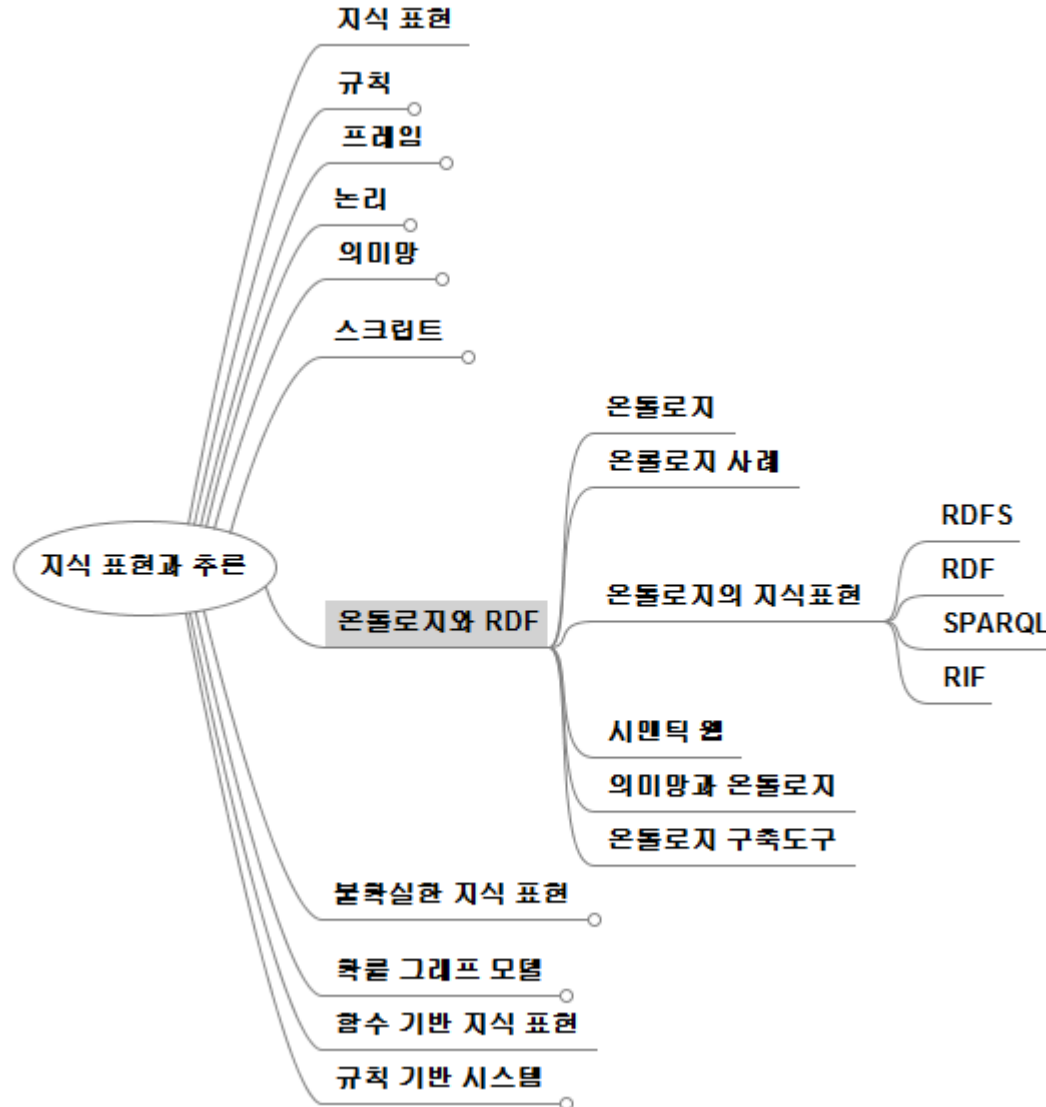
고객이 식당에 들어오면 맨 먼저 뭘 하는가?

누가 돈을 지불하는가?

식사후 식당을 나가기 전에 뭘 하는가?



7. 온톨로지



온톨로지

❖ 온톨로지(ontology)

- 철학에서 **존재론**(存在論)을 가리키는 말
- 어떤 영역의 **지식**을 **공유**하고 **재사용**할 수 있도록 **해당 영역의 개념과 관계**를 나타내는 **어휘**를 **정의**하고 이를 이용해 **지식**을 **표현**해 놓은 것
- 영역(domain)에 있는 **개념**, 개념에 대한 **특성** 및 **속성**, 이들 특성과 속성에 대한 **제약조건**, 영역에 있는 일부 **개체**에 대한 정보가 기술(記述)
- 영역에 대한 **공통된 어휘**(vocabulary) 사용
- 영역에 대한 **공통된 이해** 지원
- 서로 간 토의를 통해 **합의**를 이룬 것을 표현
- **컴퓨터**에서 다룰 수 있는 형태로 **정형화**하여 표현

온톨로지

❖ 온톨로지의 사례

- Amazon.com의 온라인 쇼핑 카탈로그
- **dmoz**(Directory of Mozilla, www.dmoz.org)
 - 분야별로 정리한 디렉토리 서비스

The screenshot shows the dmoz.org website interface. At the top, there is a dark grey header with the 'dmoz' logo on the left and 'In partnership with Aol.' on the right. Below the header is a navigation bar with links for 'about dmoz', 'dmoz blog', 'suggest URL', 'help', 'link', and 'editor login'. A search bar is located below the navigation bar, with a 'Search' button and a link to 'advanced' search. The main content area is a grid of category links, each with a bolded title and a list of sub-topics.

<u>Arts</u> Movies, Television, Music...	<u>Business</u> Jobs, Real Estate, Investing...	<u>Computers</u> Internet, Software, Hardware...
<u>Games</u> Video Games, RPGs, Gambling...	<u>Health</u> Fitness, Medicine, Alternative...	<u>Home</u> Family, Consumers, Cooking...
<u>Kids and Teens</u> Arts, School Time, Teen Life...	<u>News</u> Media, Newspapers, Weather...	<u>Recreation</u> Travel, Food, Outdoors, Humor...
<u>Reference</u> Maps, Education, Libraries...	<u>Regional</u> US, Canada, UK, Europe...	<u>Science</u> Biology, Psychology, Physics...
<u>Shopping</u> Clothing, Food, Gifts...	<u>Society</u> People, Religion, Issues...	<u>Sports</u> Baseball, Soccer, Basketball...

온톨로지

❖ 온톨로지의 사례 – cont.

- **UNSPSC**(United Nations Standard Products and Services Code)
 - 제품 및 서비스 용어
- **WordNet**(wordnet.princeton.edu)
 - 영어 단어의 어휘목록과 어휘목록 사이의 다양한 의미 관계를 기록

WordNet Search - 3.1

- [WordNet home page](#) - [Glossary](#) - [Help](#)

Word to search for:

Display Options:

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- **S: (n) ontology** ((computer science) a rigorous and exhaustive organization of some knowledge domain that is usually hierarchical and contains all the relevant entities and their relations)
 - [domain category](#)
 - **S: (n) computer science, computing** (the branch of engineering science that studies (with the aid of computers) computable processes and structures)
 - [direct hypernym](#) / [inherited hypernym](#) / [sister term](#)
- **S: (n) ontology** (the metaphysical study of the nature of being and existence)
 - [direct hypernym](#) / [inherited hypernym](#) / [sister term](#)
 - [derivationally related form](#)
 - **W: (adj) ontological** [Related to: [ontology](#)] (of or relating to ontology) "ontological speculations"

온톨로지

❖ 온톨로지의 사례 – cont.

- **UMLS**(Unified Medical Language System)
 - 의료영역의 여러 용어체계를 총괄
- **SNOMED-CT**(Systematized Nomenclature of Medicine - Clinical Terms)
 - 의료용어, 유사어, 코드
- **LOINC**(Logical Observation Identifiers Names and Codes)
 - 병원 검사 명칭 및 임상 용어 표준 체계
- **GO**(Gene Ontology)
 - 유전자와 유전자 산물
- **FreeBase**
 - Google Knowledge Cloud에서 사용
 - 사람들이 만든 RDF 형태의 지식

온톨로지

❖ 온톨로지의 사례 – cont.

- 의료 용어 온톨로지 SNOMED-CT의 감기(common cold)에 대한 항목

Parent(s):

(Select a parent to make it the "Current Concept".)

[Viral upper respiratory tract infection \(disorder\)](#)

Current Concept:

[Common cold \(disorder\)](#)

Child(ren):

(N=0) (Select a child to make it the "Current Concept".)

Current Concept:

Fully Specified Name: Common cold (disorder)

ConceptId: 82272006

Defining Relationships:

Is a [Viral upper respiratory tract infection \(disorder\)](#)
Causative agent [Virus \(organism\)](#)
Finding site [Upper respiratory tract structure \(body structure\)](#)
Pathological process [Infectious process \(qualifier value\)](#)

This concept is primitive.

Qualifiers:

[View Qualifying Characteristics and Facts](#)

Descriptions (Synonyms):

Fully Specified Name: Common cold (disorder)

Preferred: Common cold

Synonym: Acute coryza

Synonym: Acute nasal catarrh

Synonym: Acute rhinitis

Synonym: Infective rhinitis

Synonym: Acute nasopharyngitis

Synonym: Infective nasopharyngitis

Synonym: Head cold

Synonym: Acute infective rhinitis

Synonym: Cold

Synonym: Acute nasopharyngitis, NOS

Synonym: Infective nasopharyngitis, NOS

온톨로지

❖ 온톨로지의 지식표현

- 의미망과 비슷하게 방향성이 있는 그래프로 표현 가능
- **RDF**(Resource Description Framework)
 - **자원**에 대한 **메타데이터**(metadata, 데이터에 대한 데이터)를 기술하는 명세(specification)

온톨로지

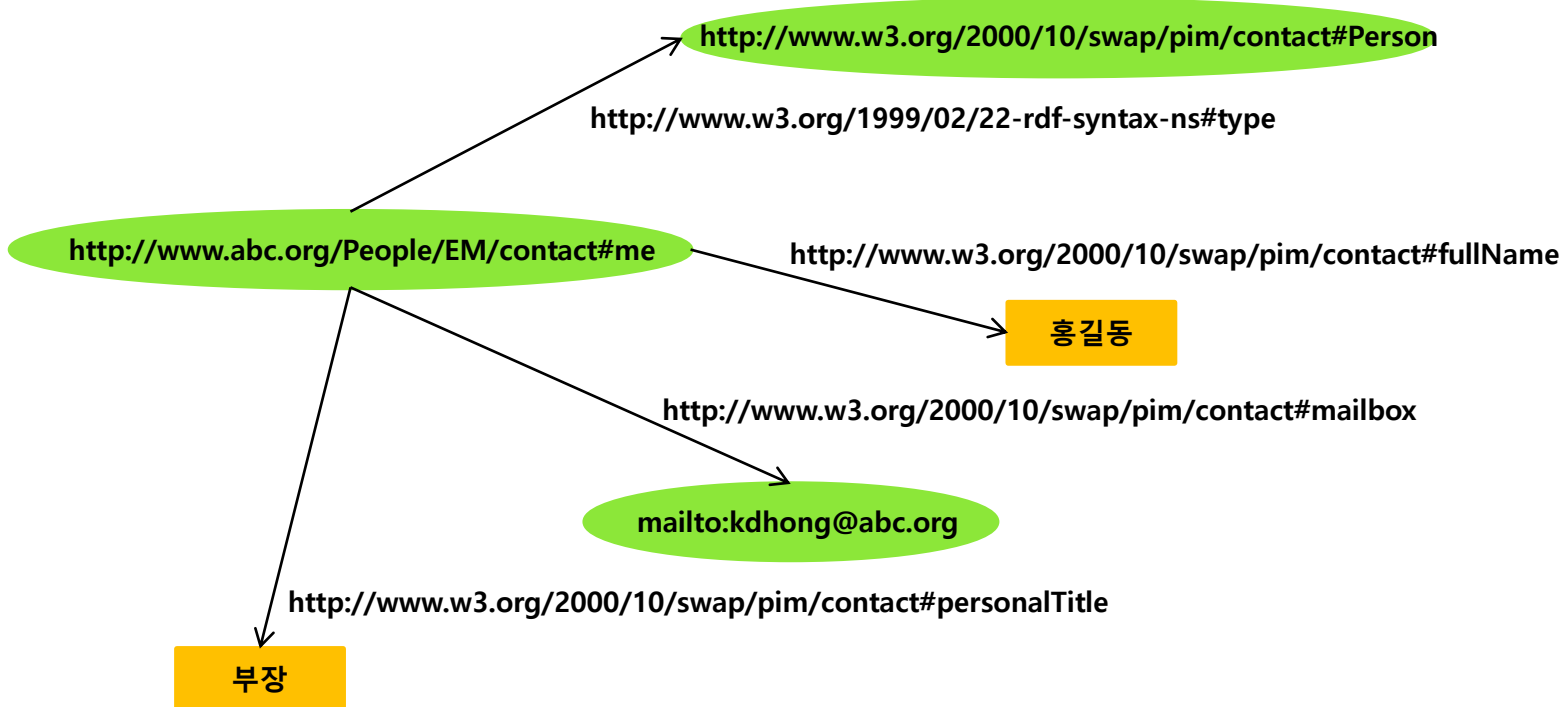
❖ 온톨로지의 지식표현

▪ RDF(Resource Description Framework)

• 특정 대상을 기술하는 간단한 문장(sentence)

- '주어(subject)+서술어(predicate)+목적어(object)'로 파악
- 자원(대상; resource) - 속성(특징, 주어와 목적어 관계, attribute) - 값(value)로 분해

• abc 기관의 이메일이 kdhong@abc.org인 홍길동 부장이라는 사람



온톨로지

❖ 온톨로지의 지식표현

▪ RDF(Resource Description Framework) – Conf.

• 튜플 형태의 표현

<http://www.abc.org/People/EM/contact#me>, <http://www.w3.org/2000/10/swap/pim/contact#fullName>, "홍길동"
<http://www.abc.org/People/EM/contact#me>, <http://www.w3.org/2000/10/swap/pim/contact#mailbox>,
<mailto:kdhong@abc.org>
<http://www.abc.org/People/EM/contact#me>, <http://www.w3.org/2000/10/swap/pim/contact#personalTitle>, "부장"
<http://www.abc.org/People/EM/contact#me>, <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>,
<http://www.w3.org/2000/10/swap/pim/contact#Person>

• XML 형태의 표현

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
  <contact:Person rdf:about="http://www.abc.org/People/EM/contact#me">
    <contact:fullName> 홍길동 </contact:fullName>
    <contact:mailbox rdf:resource="mailto:kdhong@abc.org"/>
    <contact:personalTitle> 부장 </contact:personalTitle>
  </contact:Person>
</rdf:RDF>
```

온톨로지

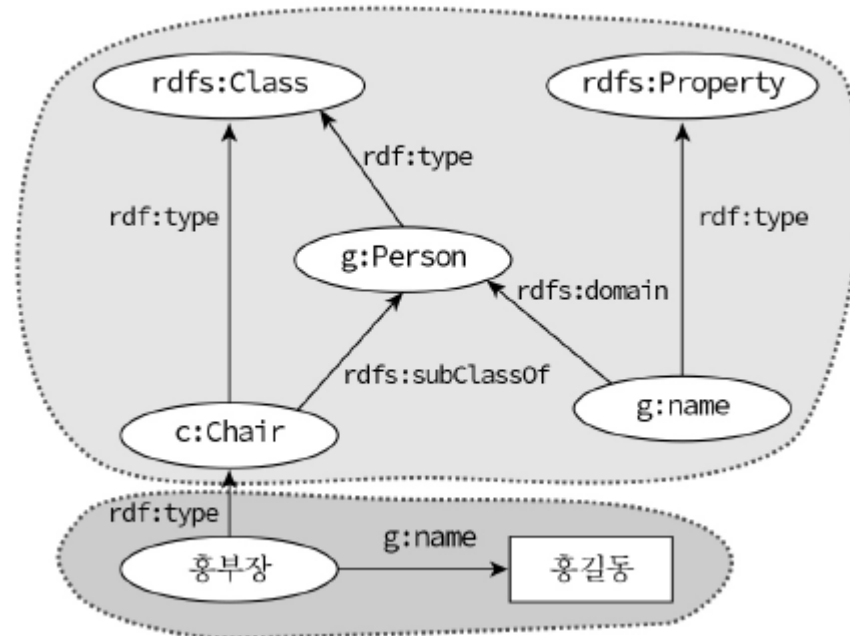
❖ 온톨로지의 지식표현 – cont.

- **RDFS**(Resource Description Framework Schema, RDF 스키마)
 - RDF를 사용하여 온톨로지를 표현할 때 사용할 관련 어휘 온톨로지를 정의하는 언어

```
<rdfs:Class rdf:ID="http://schema.org/given#Person">  
  <rdf:type> <rdfs:Class> </rdf:type>  
</rdfs:Class>  
<rdfs:Class rdf:ID="http://schema.org/com#Chair">  
  <rdfs:subClassOf rdf:resource="http://schema.org/given#Person">  
</rdfs:Class>  
<rdfs:Property rdf:ID="http://schema.org/given#name">  
  <rdfs:domain rdf:resource="http://schema.org/given#Person">  
</rdfs:Property>
```

온톨로지

- ❖ 온톨로지의 지식표현 – cont.
 - RDFS(RDF 스키마)



```
<rdf:RDF
  xmlns:g="http://schema.org/given" xmlns:c="http://schema.org/com">
  <c:Chair rdf:ID="홍부장">
    <g:name> "홍길동" </g:name>
  </c:Chair>
</rdf:RDF>
```

온톨로지

❖ 온톨로지의 지식표현

▪ SPARQL

- RDF 형태로 저장된 데이터에 대한 질의어
- SQL 유사한 문법

PREFIX contact: <http://www.w3.org/2000/10/swap/pim/contact#>

PREFIX rdf: < http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?name ?email

WHERE { ?person **rdf:type** **contact:Person**.

 ?person **contact:name** ?name.

 ?person **contact:mbox** ?email. }

온톨로지

❖ 온톨로지의 지식표현

- RIF (Rule Interchange Format)
 - 규칙을 정의하고 교환하기 위한 규약

```
Prefix(ex <http://example.com/2008/prd1#>)  
(* ex:rule_1 *)  
Forall ?customer ?purchasesYTD (  
  If And ( ?customer#ex:Customer  
           ?customer[ex:purchasesYTD->?purchasesYTD]  
           External(pred:numeric-greater-than(?purchasesYTD 5000)) )  
  Then Do ( Modify(?customer[ex:status->"Gold"]) ) )
```

당해년도 누적 구매금액이 \$5,000이상이면 "Gold" 등급으로 조정

온톨로지

❖ OWL (Web Ontology Language)

- 웹 상의 자원과 이들의 속성에 대한 지식을 표현하기 위한 온톨로지 언어
- 예
 - Vintage라는 자원이 하나의 hasVintageYear라는 속성을 하나만 갖는다는 것을 표현

```
<owl:Class rdf:ID="Vintage">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasVintageYear"/>
      <owl:cardinality
        rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

온톨로지

❖ 온톨로지의 활용

▪ 시맨틱 웹(semantic web)

- 웹의 데이터를 **소프트웨어 에이전트**가 이해하여 지능적으로 활용할 수 있도록 하는 것
- 웹의 처음 설계한 **Tim Bernes-Lee**가 주창한 아이디어
- 의미있는 태그(tag)를 정의하여 문서를 기술하기 위해 **XML** 사용
- 태그 및 데이터의 의미 해석을 위해 **RDF** 사용
- **온톨로지** 구축을 통해 태그 및 메타 데이터의 의미 해석

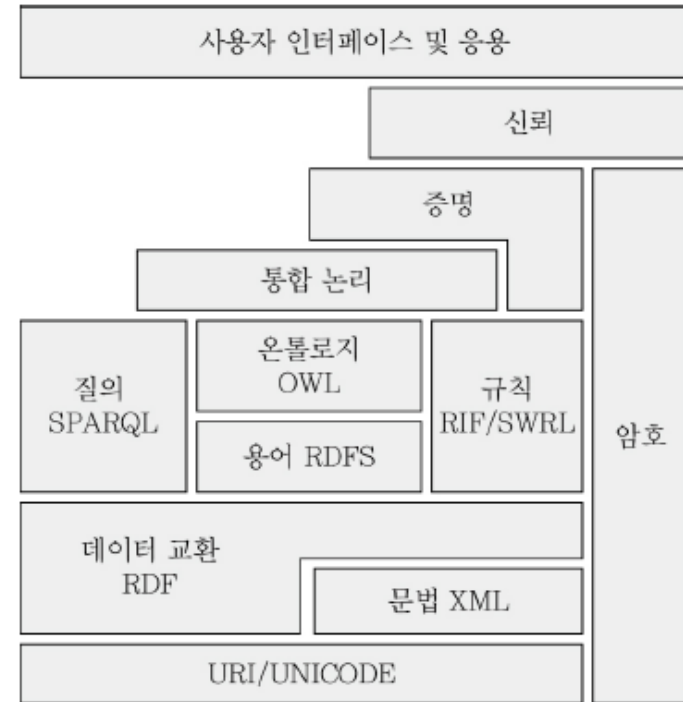


그림 3.15 W3C 시맨틱 웹 기술 스택

온톨로지

❖ 의미망과 온톨로지

- 그래프 구조를 이용 지식 모델링
- 의미망
 - 대상, 관계 등의 표현에 사용되는 용어가 임의적
 - 통일된 표현 관련 규정 부재
- 온톨로지
 - 다른 시스템과 공유와 상호운영성(interoperability)를 위해 명확한 지침에 따라 표현
 - 구축된 정보 및 지식의 재사용에 관심

온톨로지와 RDF

❖ 온톨로지 구축 도구

▪ Protégé 등

- 지식 획득 및 표현을 위한 GUI 기반의 오픈 소스 온톨로지 편집기

The image shows two windows from the Protégé ontology editor. The left window, titled 'Classes', displays a hierarchical tree of ontology classes. The right window, titled 'Class description for annulate lamellae', provides detailed information for the selected class.

Classes Window:

- owl:Thing
 - biological_process
 - cellular_component
 - axon part
 - axoneme part
 - bacterial-type flagellum part
 - cell
 - cell cortex part
 - cell division site part
 - cell junction
 - cell part
 - annulate lamellae**
 - apical complex
 - apical part of cell
 - ascus epiplasm
 - aspartate carbamoyltransferase complex
 - basal labyrinth
 - basal part of cell
 - basal ring of apical complex
 - CD20-Lck-Fyn complex
 - CD20-Lck-Lyn-Fyn complex
 - cell body
 - cell division site
 - cell division site part
 - cell envelope Sec protein transport complex
 - cell leading edge
 - cell periphery
 - cell pole
 - cell projection
 - cell projection part
 - cell septum
 - cell septum part



지식표현과 추론

Part III

충북대학교 소프트웨어학과
이건명

8. 함수에 의한 지식 표현

❖ 기호 기반의 지식 표현

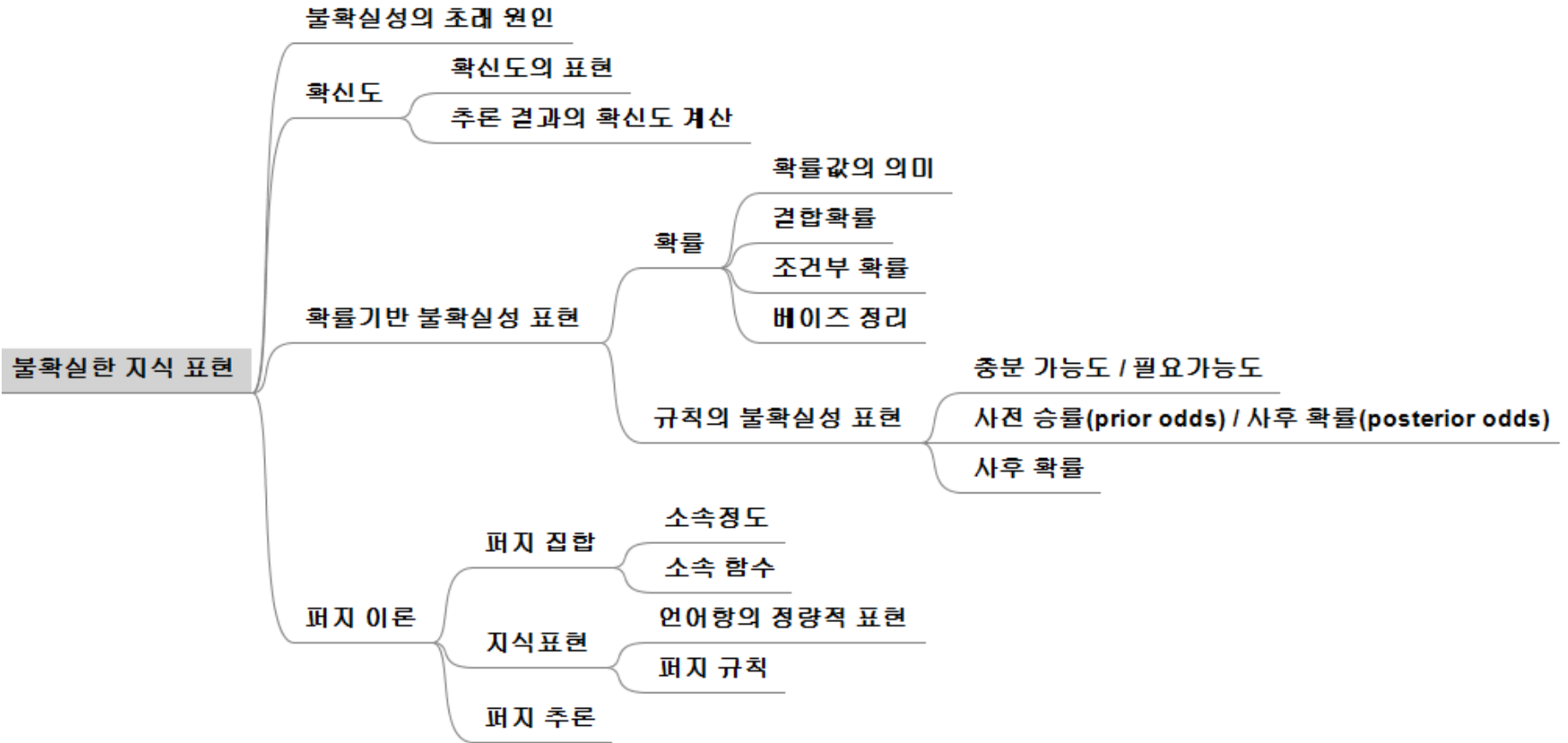
- 기호를 사용하여 대상 표현
- 대상 간의 관계 표현
- 규칙, 프레임, 의미망, 논리 등

❖ 함수 기반의 지식 표현

- 비기호적 지식표현
 - 기호 대신 수치값과 수치값을 계산하는 함수를 사용하여 지식을 표현
- 신경망과 딥러닝
 - 퍼셉트론(Perceptron) : 함수식 계산 출력 생성
 - 다층 퍼셉트론
- 서포트 벡터 머신(SVM)
- 회귀(regression) 모델 등

$$f(x) = \frac{1}{1 + e^{-0.5 + 1.2x_1 - 0.3x_2 + 2.4x_3}}$$

9. 불확실한 지식 표현



불확실한 지식 표현

❖ 불확실성의 원인

▪ 약한 관련성의 지식

- 약한 인과성(weak implication)이나 애매한 연관관계(vague association)인 지식의 표현

ex. IF(조건)와 THEN(취해야 할 행동) 사이의 연관성의 강도

⇒ 확신도(certainty degree) 사용 표현

⇒ 베이즈 정리(Bayesian theorem) 사용 표현

▪ 부정확한 언어 사용

- 자연어(natural language)는 본질적으로 모호하고(vague) 부정확(imprecise)

ex. 자주(frequently), 크다(big), 무겁다(heavy)

⇒ 퍼지이론(fuzzy theory) 사용 표현

불확실한 지식 표현

❖ 불확실성의 원인 - cont.

- 불완전하거나(incomplete) 결손된(missing) 데이터에 기반한 지식
⇒ '알려지지 않은 것(unknown)'으로 간주하고, 근사적인 추론 (approximate reasoning) 진행
- 상충되는 지식의 통합
 - 모순된 견해(contradictory opinion)와 상충된 지식(conflicting knowledge)의 통합
⇒ 지식 소스 별로 가중치 부여

8.1 확신도

❖ 확신도(certainty factor)

- 규칙(rule)과 사실(fact)의 신뢰정도를 $[-1,1]$ 구간의 값으로 표현
 - **1** (단정적 신뢰), **-1** (단정적 불신)

- 규칙과 사실에 **확신도 cf** 부여

- 규칙 : IF A THEN B $cf(r)$
- 사실 : A $cf(A)$
- 추론 결과: B $cf(B)$

```
IF      the sky is clear
THEN   the forecast is sunny {cf 0.8}
```

- **확신도 값에 따른 대응 단어**

- -1.0 : 절대 아니다 (definitely not)
- -0.8 : 거의 확실히 아니다 (almost certainly not)
- -0.6 : 아마 아니 것이다 (probably not)
- -0.4 : 어쩌면 아닐 것이다 (maybe not)
- -0.2 ~ 0.2 : 모르겠다 (unknown)
- 0.4 : 어쩌면 그럴 것이다 (maybe)
- 0.6 : 아마 그럴 것이다 (probably)
- 0.8 : 거의 확실하다 (almost certainly)
- 1.0 : 확실하다 (definitely)

확신도

❖ 규칙에 대한 추론 결과의 확신도

IF A THEN B	$cf(A \rightarrow B)$
A	$cf(A)$
<hr/>	
B	$cf(B)$

$$cf(B) = cf(A) \times cf(A \rightarrow B)$$

IF A and B THEN C	$cf(A \rightarrow B)$
A	$cf(A)$
B	$cf(B)$
<hr/>	
C	$cf(C)$

$$cf(C) = \min\{cf(A), cf(B)\} \times cf(A \rightarrow B)$$

IF A or B THEN C	$cf(A \rightarrow B)$
A	$cf(A)$
B	$cf(B)$
<hr/>	
C	$cf(C)$

$$cf(C) = \max\{cf(A), cf(B)\} \times cf(A \rightarrow B)$$

확신도

❖ 규칙에 대한 추론 결과의 확신도 – cont.

IF sky is clear

AND the forecast is sunny

THEN the action is 'wear sunglasses' $\{cf = 0.8\}$

'sky is clear' $cf = 0.9$,

'the forecast is sunny' $cf = 0.7$

'wear sunglasses' $cf = \min\{0.9, 0.7\} \times 0.8 = 0.7 \times 0.8 = 0.56$

IF sky is overcast

OR the forecast is rain

THEN the action is 'take an umbrella' $\{cf = 0.9\}$

'sky is overcast' $cf = 0.6$

'the forecast is rain' $cf = 0.8$

'take an umbrella' $cf = \max\{0.6, 0.8\} \times 0.9 = 0.8 \times 0.9 = 0.72$

확신도

❖ 규칙에 대한 추론 결과의 확신도 – cont.

- 여러 규칙에 의한 동일 사실 추론의 확신도 결합

IF A THEN B	$cf(A \rightarrow B)$
A	$cf(A)$
<hr/>	
B	$cf(B)$

$$cf_1(B) = cf(A) \times cf(A \rightarrow B)$$

IF C THEN B	$cf(C \rightarrow B)$
C	$cf(C)$
<hr/>	
B	$cf(B)$

$$cf_2(B) = cf(C) \times cf(C \rightarrow B)$$

$$cf(cf_1, cf_2) = \begin{cases} cf_1 + cf_2 \cdot (1 - cf_1) & \text{if } cf_1 \geq 0 \text{ and } cf_2 \geq 0 \\ \frac{cf_1 + cf_2}{1 - \min\{|cf_1|, |cf_2|\}} & \text{if } cf_1 \text{와 } cf_2 \text{중에서 하나만 음수} \\ cf_1 + cf_2 \cdot (1 + cf_1) & \text{if } cf_1 < 0 \text{ and } cf_2 < 0 \end{cases}$$

$$cf(cf_1, cf_2) = \frac{0.8 - 0.6}{1 - \min[0.8, 0.6]} = 0.5$$

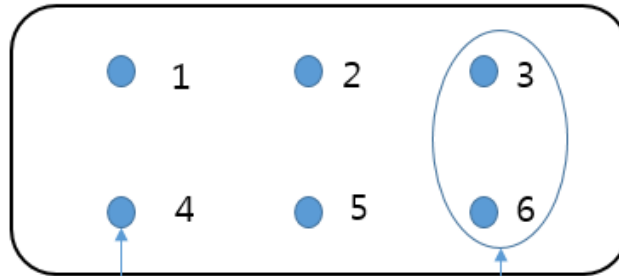


9.2 확률기반 불확실성 표현

❖ 확률 (probability)

- 어떤 사건이 일어날 가능성
- 확률의 의미
 - 상대빈도 확률(relative frequency probability)
 - 빈도주의자 확률(frequentist probability)
 - 전체 실험 회수 대비 관심 사건의 상대적 빈도

표본 공간 S



확률실험 결과

사건

- 주관적 확률(subjective probability)
 - 확신 또는 믿음의 정도(degree of belief)



확률기반 불확실성 표현

❖ 결합 확률(joint probability)

- $P(A, B)$, $P(A \cap B)$, $P(AB)$
- 사건 A 와 B 가 **동시에 일어날 확률**
- 예.
 - A : 첫번째 주사위가 짝수, B : 두번째 주사위가 홀수
 - $P(A, B) = \frac{9}{36} = 0.25$



(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)
(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)

확률기반 불확실성 표현

❖ 조건부 확률(conditional probability)

- $P(A|B)$
- B 가 주어질 때 A 가 일어날 확률

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(AB)}{P(B)} = \frac{P(A, B)}{P(B)} \quad \text{where } P(B) > 0.$$



A : 두 주사위의 합이 8이다
 B : 첫 번째 주사위는 3이다

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)
(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)

$$P(A|B) = \frac{P(A, B)}{P(B)} = \frac{1/36}{6/36} = \frac{1}{6}$$

확률기반 불확실성 표현

❖ 베이즈 정리 (Bayesian theorem)

사후 확률 가능도 사전 확률

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

증거

$$P(A|B) = \frac{P(A, B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

- 사후 확률 (posterior probability)
- 가능도 (likelihood)
- 사후 확률 (posterior probability)
- 증거 (evidence)

확률기반 불확실성 표현

❖ 확률을 이용한 규칙의 불확실성 표현

- 전문가에 의한 각 규칙에 대한 충분 가능성도 LS , 필요 가능성도 LN 값 부여
- 규칙 : $A \rightarrow B$

IF today is rain
THEN tomorrow is rain

- 충분 가능성도 (likelihood of sufficiency)

- $LS = \frac{P(A|B)}{P(A|\neg B)}$

$$LS = \frac{p(\text{today is rain} | \text{tomorrow is rain})}{p(\text{today is rain} | \text{tomorrow is dry})}$$

- 필요 가능성도 (likelihood of necessity)

- $LN = \frac{P(\neg A|B)}{P(\neg A|\neg B)}$

$$LN = \frac{p(\text{today is dry} | \text{tomorrow is rain})}{p(\text{today is dry} | \text{tomorrow is dry})}$$

- 사실 또는 추론 결과에 대한 사전확률 (prior probability) 부여

Rule: 1

IF today is rain {LS 2.5 LN .6}
THEN tomorrow is rain {prior .5}

Rule: 2

IF today is dry {LS 1.6 LN .4}
THEN tomorrow is dry {prior .5}

확률기반 불확실성 표현

❖ 확률을 이용한 규칙의 불확실성 표현

- 규칙 : $A \rightarrow B$

Rule: 1

IF today is rain {LS 2.5 LN .6}

THEN tomorrow is rain {prior .5}

- 사전 승률(prior odds)

- $O(B) = \frac{P(B)}{1-P(B)}$

$$O(\text{tomorrow is rain}) = \frac{0.5}{1 - 0.5} = 1.0$$

- 사후 승률(posterior odds)

- $O(B|A) = LS \times O(B)$

- $O(B|\neg A) = LN \times O(B)$

$$O(\text{tomorrow is rain} | \text{today is rain}) = 2.5 \times 1.0 = 2.5$$

- 사후 확률

- $P(B|A) = \frac{o(B|A)}{1+o(B|A)}$

- $P(B|\neg A) = \frac{o(B|\neg A)}{1+o(B|\neg A)}$

$$p(\text{tomorrow is rain} | \text{today is rain}) = \frac{2.5}{1 + 2.5} = 0.71$$



9.3 퍼지 이론

❖ 집합론

- 자연어의 **단어(word)**는 집합의 궁극적인 표현
 - '자동차'는 자동차의 집합
 - '자동차 한 대'는 자동차 집합의 원소 하나
- **일반 집합(crisp set, classical set)** X
 - 원소 x 는 X 에 속하거나($x \in X$), X 에 속하지 않거나($x \notin X$) 둘 중 하나
 - 집합에 **명확한 경계**를 긋고,
집합의 원소에는 1, 원소가 아닌 것에는 0의 **소속**(membership)



퍼지 이론

- ❖ 개념이나 범주가 항상 이분적(二分的)이지는 않다
 - 자전거 vs 오토바이



- 정도(degree)의 문제
⇒ 퍼지 집합(fuzzy set) 도입

퍼지 이론

❖ 퍼지집합(Fuzzy Set)

- 원소가 모임(collection)에 어느 정도 속한다는 것
- 명제는 참 또는 거짓이 아니라 어느 정도는 부분적으로 참(이거나 부분적으로 거짓)
- 소속정도(membership degree)는 $[0,1]$ 범위의 실수값으로 표현

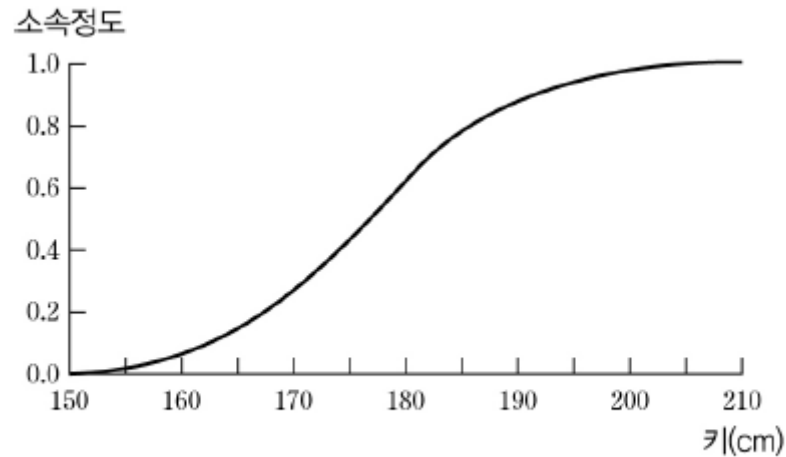
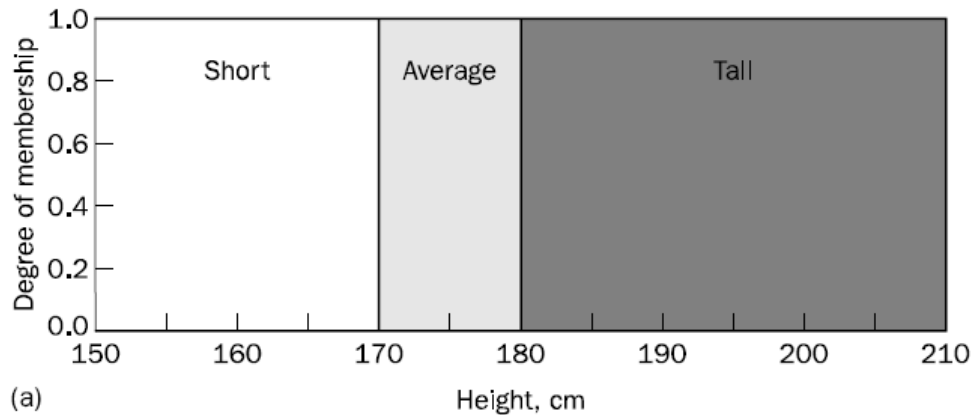


그림 3.16 '크다'를 나타내는 소속함수의 예

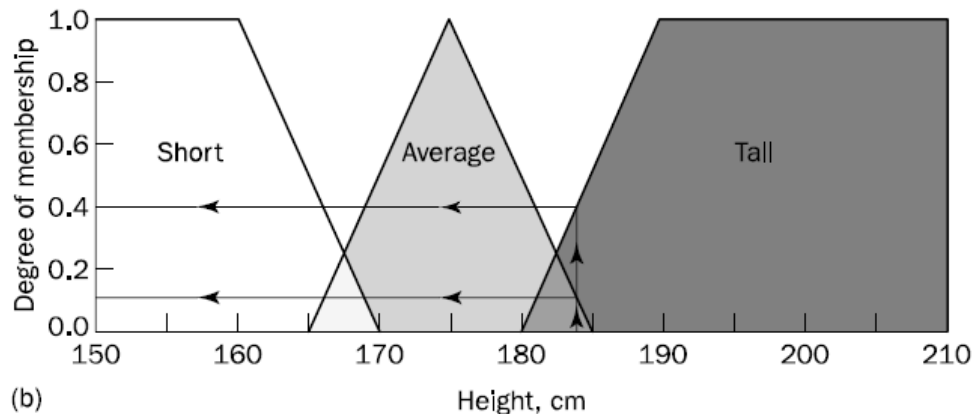
퍼지 이론

❖ 소속정도를 사용한 언어항 (linguistic term)의 표현

- 예. 작다(short), 평균이다(average), 크다(tall)의 표현
 - 일반 집합



- 퍼지 집합



퍼지 이론

❖ 언어항을 포함한 지식 표현

- 퍼지 규칙(fuzzy rule) 사용

- 소속함수로 표현된 언어항을 포함하는 규칙

IF 서비스가 나쁘거나 음식이 별로이다 THEN 팁을 적게 준다

IF 서비스가 좋다 THEN 팁을 보통으로 준다

IF 서비스가 훌륭하거나 음식이 맛있다 THEN 팁을 많이 준다

- 언어항

- '나쁘다', '좋다', '훌륭하다', '별로이다', '맛있다', '적다', '보통이다', '많다'

퍼지 이론

- ❖ 언어항을 포함한 지식 표현
 - 언어항을 표현하는 소속함수

IF service = 나쁘다 OR food = 별로이다 THEN tip = 적다
IF service = 좋다 THEN tip = 보통이다
IF service = 훌륭하다 OR food = 맛있다 THEN tip = 많다

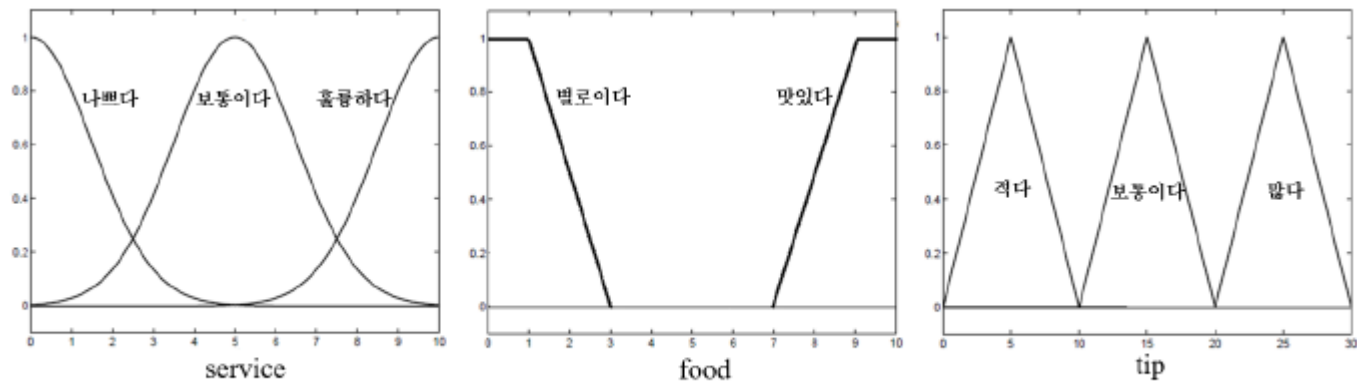


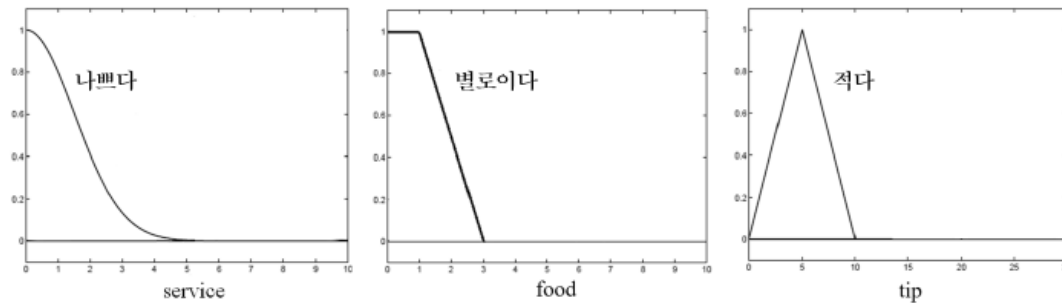
그림 3.17 언어항과 소속함수

퍼지 이론

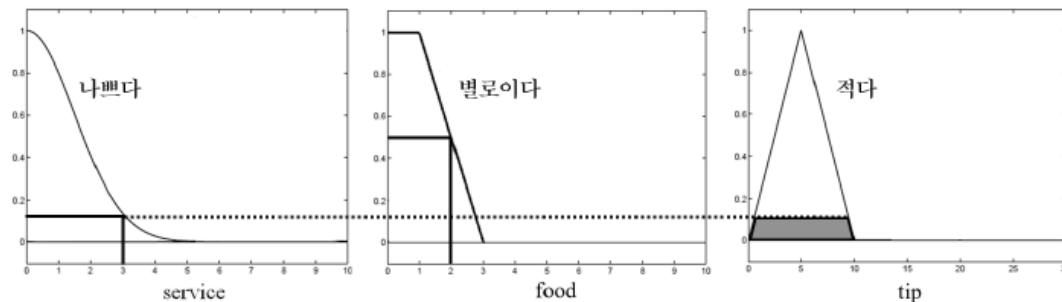
❖ 퍼지 추론(fuzzy inference)

- 소속함수로 표현된 언어항을 사용하는 퍼지 규칙들의 모음
- 언어항의 기호적인 대응을 통한 추론 대신, 수치적인 추론이 가능해짐
 - 수치값 입력에 대해 수치값 출력을 생성

IF service = 나쁘다 OR food = 별로이다 THEN tip = 적다



service = 3, food = 2

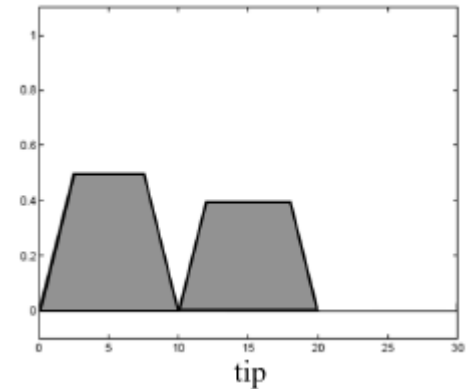


퍼지 이론

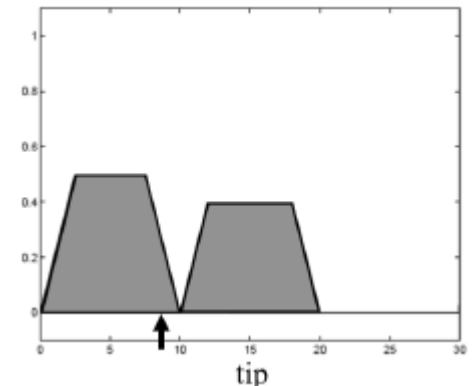
❖ 퍼지 추론(fuzzy inference)

- 세 개의 퍼지 규칙에 대한 추론 결과

IF service = 나쁘다 OR food = 별로이다 THEN tip = 적다
IF service = 좋다 THEN tip = 보통이다
IF service = 훌륭하다 OR food = 맛있다 THEN tip = 많다



- 비퍼지화(defuzzification)
 - 퍼지 추론의 결과를 실수 값으로 변환하는 것
 - 예. 무게중심(center of gravity) 법

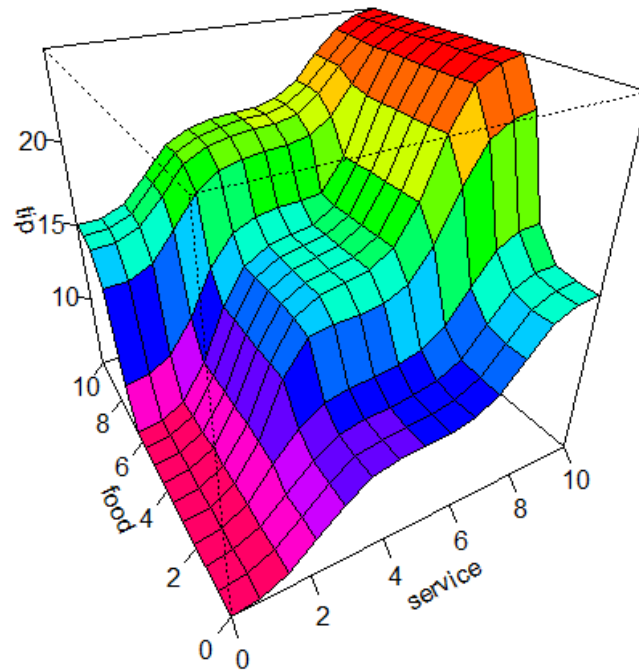


퍼지 이론

❖ 예. 레스토랑 팁 계산

- 서비스 평가값(service)과 음식 평가값(food)에 따른 팁(tip)의 크기

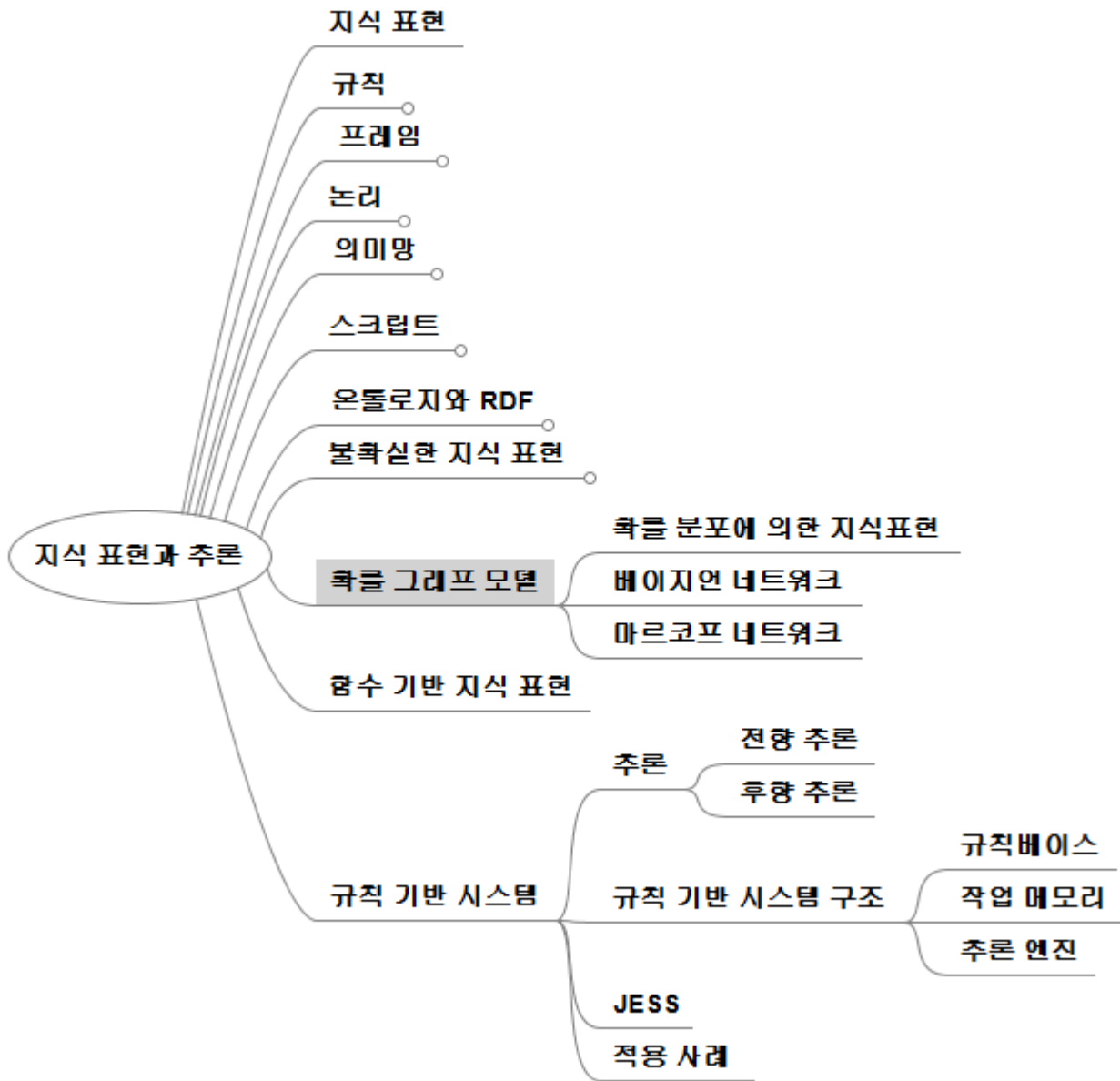
IF 서비스가 나쁘거나 음식이 별로이다 THEN 팁을 적게 준다
IF 서비스가 좋다 THEN 팁을 보통으로 준다
IF 서비스가 훌륭하거나 음식이 맛있다 THEN 팁을 많이 준다



지식표현과 추론

Part IV

충북대학교 소프트웨어학과
이건명



9. 확률 그래프 모델

❖ 확률 그래프 모델(probabilistic graph model)

- 확률 이론과 그래프 이론을 결합하여 **확률분포**(probability distribution)를 **표현**하고,
관심있는 대상(**확률변수**)에 대한 **확률**을 **계산**할 수 있는 모델

- 베이지언 망
- 마르코프 랜덤 필드
- 조건부 랜덤 필드
- 로그-선형 모델

확률 그래프 모델

❖ 확률 그래프 모델(probabilistic graph model)

▪ 예. 절도 경보 문제

- 절도가 발생하거나 지진이 발생하면 경보 발생
- 경보가 울리면 이웃이 전화

• 불확실한 요소가 있어 확률로 표현

- 확률변수(random variable)
 - » 경보 작동(A; alarm)
 - » 절도 발생(B; burglary)
 - » 지진 발생(E; earthquake)
 - » 이웃 전화(N; neighbor call)



확률 그래프 모델

- 절도 경보 문제의 확률분포에 의한 지식표현
 - 지진 발생(E), 절도 발생(B), 경보 생성(A), 이웃전화 (N)
 - 결합확률 분포로 표현

<i>E</i>	<i>B</i>	<i>A</i>	<i>N</i>	확률
F	F	F	F	0.56133
F	F	F	T	0.06237
F	F	T	F	0.00126
F	F	T	T	0.00504
F	T	F	F	0.0243
F	T	F	T	0.0027
F	T	T	F	0.0486
F	T	T	T	0.1944
T	F	F	F	0.0189
T	F	F	T	0.0021
T	F	T	F	0.0098
T	F	T	T	0.0392
T	T	F	F	0.00027
T	T	F	T	0.00003
T	T	T	F	0.00594
T	T	T	T	0.02376



경보가 울릴 때 이웃이 전화할 확률은?

$$P(N = T | A = T) = ?$$

이웃이 전화했을 때 도둑이 들었을 확률은?

$$P(B = T | N = T) = ?$$

확률 그래프 모델

❖ 조건부 독립과 확률분포의 인수분해(factorization)

- 사건의 독립(independence)
 - $P(E, B) = P(E)P(B)$
- 조건부 독립(conditional independence)성질 이용
 - $P(R, A|E) = P(R|E)P(A|E)$
- 확률분포의 인수분해
 - $P(A, B) = P(A|B)P(B)$
 - $P(A_1, A_2, A_3, A_4) = P(A_1|A_2, A_3, A_4)P(A_2|A_3, A_4)P(A_3|A_4)P(A_4)$

확률 그래프 모델

❖ 조건부 독립을 이용한 확률분포의 인수 분해

$$\blacksquare P(N, A, E, B) = P(N|A, E, B)P(A|E, B)P(E|B)P(B)$$

만족하는 조건부 독립 성질

$$P(N|A, E, B) = P(N|A)$$

$$P(E|B) = P(E)$$

$$= P(N|A)P(A|E, B)P(E)P(B)$$

<i>A</i>	<i>N</i>	
	F	T
F	0.9	0.1
T	0.2	0.8

<i>E</i>	<i>B</i>	<i>A</i>	
		F	T
F	F	0.99	0.01
F	T	0.1	0.9
T	F	0.3	0.7
T	T	0.01	0.99

<i>E</i>	
F	T
0.9	0.1

<i>B</i>	
F	T
0.7	0.3

$$P(N = T, A = T, E = F, B = T)$$

$$= P(N = T|A = T)P(A = T|E = F, B = T)P(E = F)P(B = T)$$

$$= 0.8 \times 0.9 \times 0.9 \times 0.3$$

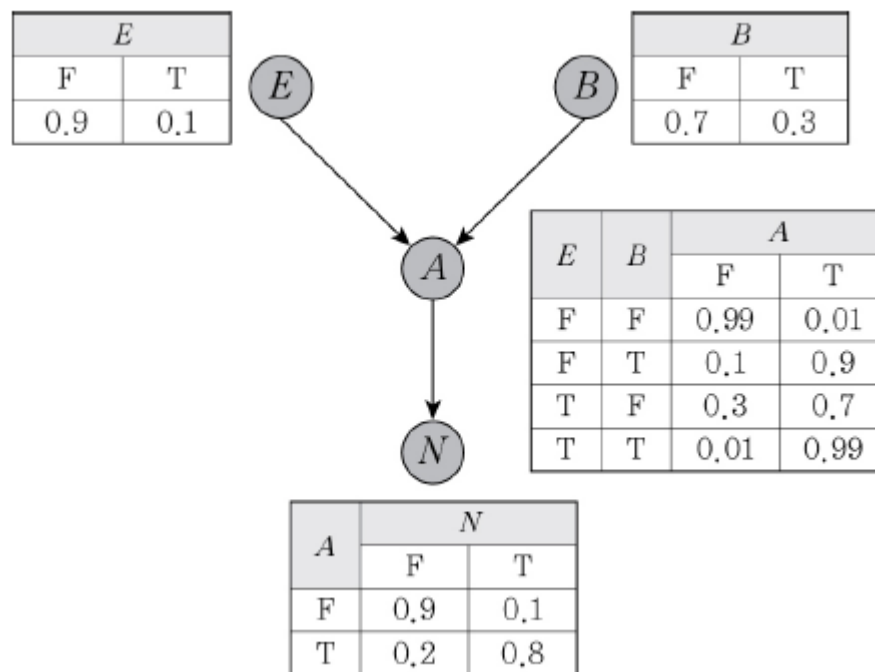
$$= 0.1944$$

확률 그래프 모델

❖ 베이저안 망(Bayesian network)

- 조건부 확률의 곱으로 표현된 확률분포를 방향성 그래프(directed graph)로 표현한 것
 - 노드 : 확률 변수
 - 에지 : 의존관계

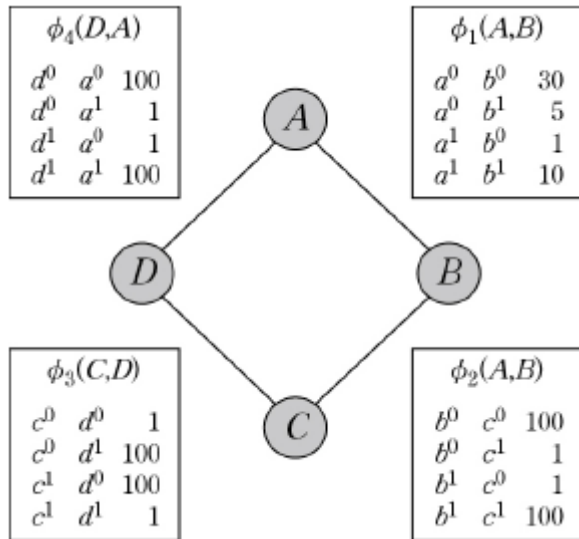
$$P(N, A, E, B) = P(N|A, E, B)P(A|E, B)P(E|B)P(B) = P(N|A)P(A|E, B)P(E)P(B)$$



확률 그래프 모델

❖ 마르코프 랜덤 필드(Markov network)

- 확률분포를 무방향 그래프(undirected graph)를 사용하여 표현
- 마르코프 랜덤 필드(Markov random field)라고도 함
- 확률분포를 요소함수(factor)들의 곱으로 표현



A	B	C	D	확률
a^0	b^0	c^0	d^0	0.04
a^0	b^0	c^0	d^1	0.04
a^0	b^0	c^1	d^0	0.04
a^0	b^0	c^1	d^1	$4.1 \cdot 10^{-6}$
a^0	b^1	c^0	d^0	$6.9 \cdot 10^{-5}$
a^0	b^1	c^0	d^1	$6.9 \cdot 10^{-5}$
a^0	b^1	c^1	d^0	0.69
a^0	b^1	c^1	d^1	$6.9 \cdot 10^{-5}$
a^1	b^0	c^0	d^0	$1.4 \cdot 10^{-5}$
a^1	b^0	c^0	d^1	0.14
a^1	b^0	c^1	d^0	$1.4 \cdot 10^{-5}$
a^1	b^0	c^1	d^1	$1.4 \cdot 10^{-5}$
a^1	b^1	c^0	d^0	$1.4 \cdot 10^{-6}$
a^1	b^1	c^0	d^1	0.014
a^1	b^1	c^1	d^0	0.014
a^1	b^1	c^1	d^1	0.014



확률 그래프 모델

❖ 연속인 확률변수가 포함된 확률 분포

- 표(table)를 사용한 표현 곤란
- 함수식을 이용한 표현
 - 베이지안 망 : 조건부 확률 값 출력 함수 사용
 - 마르코프 랜덤 필드 : 지수함수와 같은 함수식 사용

$$\phi(X_i, X_j, X_k) = \exp(f(X_i, X_j, X_k)) : \text{팩터(factor)}$$

X_i, X_j, X_k : 확률 변수

$f(X_i, X_j, X_k)$: 특정 특징의 유무나 정도 등을 계산하는 함수식

확률 그래프 모델

❖ 조건부 랜덤 필드(conditional random field, CRF)

- 조건부 확률분포를 표현하는 마르코프 랜덤 필드
- \mathbf{X} : 관측되는 대상이나 입력을 나타내는 확률변수들의 집합
- \mathbf{Y} : 추정하거나 예측하는 대상을 나타내는 확률변수들의 집합
- $\{\phi_1(D_1), \phi_2(D_2), \dots, \phi_n(D_n)\}$: \mathbf{Y} 의 확률변수를 하나라도 포함한 팩터의 집합
- 조건부 확률 정의

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})} \tilde{P}(\mathbf{X}, \mathbf{Y})$$

$$\tilde{P}(\mathbf{X}, \mathbf{Y}) = \prod_{i=1}^m \phi_i(D_i)$$

$$Z(\mathbf{X}) = \sum_{\mathbf{Y}} \tilde{P}(\mathbf{Y}, \mathbf{X}) \quad \text{분할 함수(partition function)}$$

확률 그래프 모델

❖ 로그-선형 모델(log-linear model)

▪ 팩터가 지수 함수로 표현되는 마르코프 랜덤 필드 모델

- $\mathbf{X} = \{X_1, X_2, \dots, X_m\}$: 확률변수의 집합
- 팩터의 형태

$$\phi_i = \exp(-w_i f_i(D_i))$$

- 확률 분포의 표현

$$P(X_1, X_2, \dots, X_n) = \frac{1}{Z} \exp\left[-\sum_{i=1}^K w_i f_i(D_i)\right]$$

10. 규칙 기반 시스템

❖ 규칙 기반 시스템(rule-based system)

- 지식을 규칙의 형태로 표현
- 주어진 문제 상황에 적용될 수 있는 규칙들을 사용하여 문제에 대한 해를 찾으려 하는 **지식 기반 시스템(knowledge-based system)**
- **전문가 시스템(expert system)**을 구현하는 전형적인 형태
 - 특정 문제영역에 대해서 전문가 수준의 해를 찾아주는 시스템

11.1 추론

❖ 추론

- 구축된 지식과 주어진 데이터나 정보를 이용하여 새로운 사실을 생성하는 것
- **전향 추론**(forward chaining, **순방향 추론**)
 - 규칙의 조건부와 만족하는 사실이 있을 때 규칙의 결론부를 실행하거나 처리
- **후향 추론**(backward chaining, **역방향 추론**)
 - 어떤 사실을 검증하거나 확인하고 싶은 경우에 관심 대상 사실을 결론부에 가지고 있는 규칙을 찾아서 조건부의 조건들이 모두 만족하는지 확인

전향 추론의 예

R1: IF ?x는 체모가 있다 THEN ?x는 포유류이다.

R2: IF ?x는 수유를 한다 THEN ?x는 포유류이다.

R3: IF ?x는 깃털이 있다 THEN ?x는 조류이다.

R4: IF ?x는 난다 AND ?x는 알을 낳는다 THEN ?x는 조류이다.

R5: IF ?x는 포유류이다 AND ?x는 고기를 먹는다 THEN ?x는 육식동물이다.

R6: IF ?x는 포유류이다 AND ?x는 되새김질한다 THEN ?x는 유제류이다.

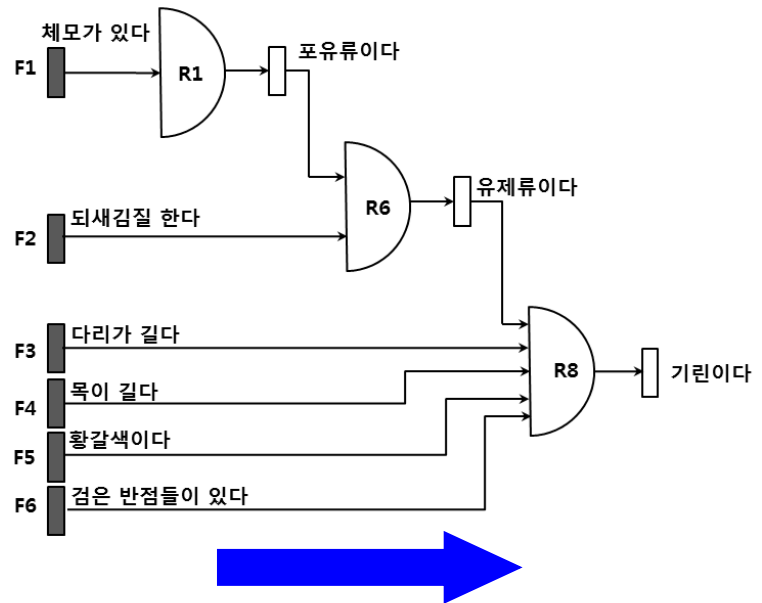
R7: IF ?x는 육식동물이다 AND ?x는 황갈색이다 AND ?x는 검은 반점들이 있다 THEN ?x는 치타이다.

R8: IF ?x는 유제류이다 AND ?x는 다리가 길다 AND ?x는 목이 길다 AND ?x는 검은 반점들이 있다 THEN ?x는 기린이다.

R9: IF ?x는 포유류이다 AND ?x는 눈이 앞을 향해있다 AND ?x는 발톱이 있다 AND ?x는 이빨이 뾰족하다 THEN ?x는 육식동물이다.

- F1: 래더는 체모가 있다.
- F2: 래더는 되새김질을 한다.
- F3: 래더는 다리가 길다.
- F4: 래더는 목이 길다.
- F5: 래더는 황갈색이다.
- F6: 래더는 검은 반점들이 있다

래더는 뭘까?

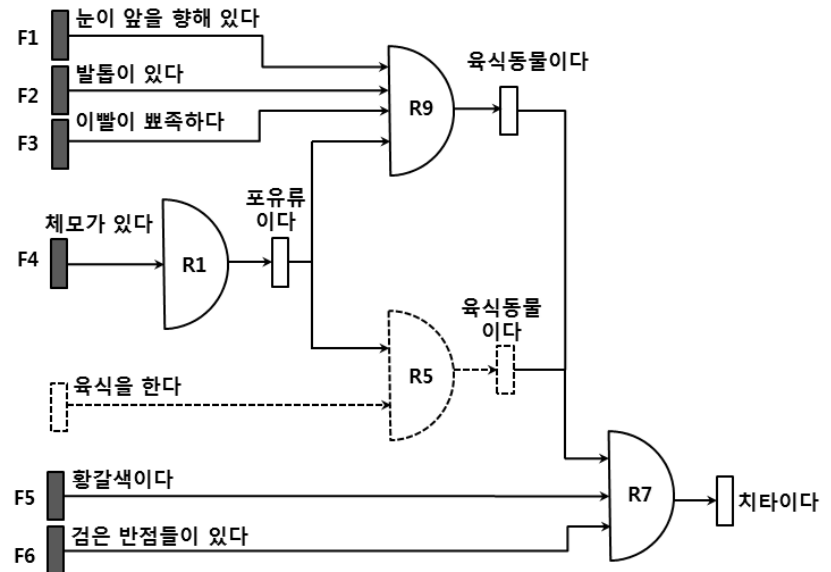


후향 추론의 예

- R1: IF ?x는 체모가 있다 THEN ?x는 포유류이다.
- R2: IF ?x는 수유를 한다 THEN ?x는 포유류이다.
- R3: IF ?x는 깃털이 있다 THEN ?x는 조류이다.
- R4: IF ?x는 난다 AND ?x는 알을 낳는다 THEN ?x는 조류이다.
- R5: IF ?x는 포유류이다 AND ?x는 고기를 먹는다 THEN ?x는 육식동물이다.
- R6: IF ?x는 포유류이다 AND ?x는 되새김질한다 THEN ?x는 유제류이다.
- R7: IF ?x는 육식동물이다 AND ?x는 황갈색이다 AND ?x는 검은 반점들이 있다 THEN ?x는 치타이다.
- R8: IF ?x는 유제류이다 AND ?x는 다리가 길다 AND ?x는 목이 길다 AND ?x는 검은 반점들이 있다 THEN ?x는 기린이다.
- R9: IF ?x는 포유류이다 AND ?x는 눈이 앞을 향해 있다 AND ?x는 발톱이 있다 AND ?x는 이빨이 뾰족하다 THEN ?x는 육식동물이다.

- F1: 스프린터는 눈이 앞을 향해 있다.
- F2: 스프린터는 발톱이 있다.
- F3: 스프린터는 이빨이 뾰족하다.
- F4: 스프린터는 체모가 있다.
- F5: 스프린터는 황갈색이다.
- F6: 스프린터는 검은 반점들이 있다.

스프린터는 치타인가?

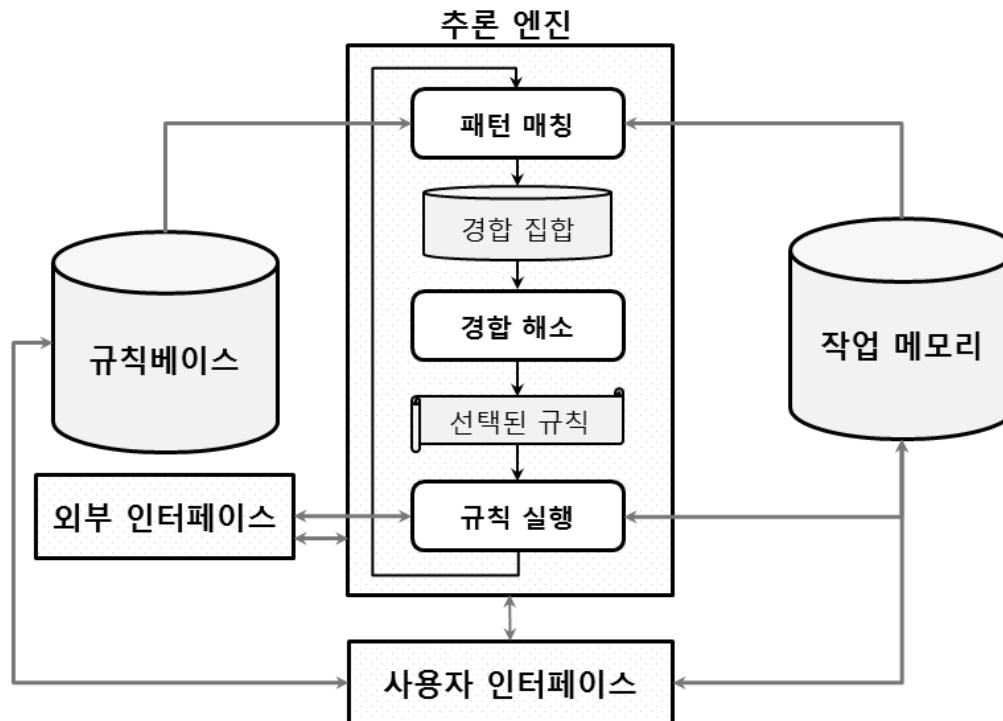


10.2 규칙 기반 시스템 구조

❖ 규칙 기반 시스템 구조

▪ 지식

- 규칙과 사실로 기술
- **규칙(rule)** : 문제 해결을 위한 지식
- **사실(fact)** : 문제 영역에 대해 알려진 데이터나 정보



규칙 기반 시스템 구조

❖ 규칙 기반 시스템 구조

- 규칙베이스(rule base)
 - 전체 규칙의 집합을 관리하는 부분
 - 생성 메모리(production memory)라고도 함
- 작업 메모리(working memory)
 - 사용자로부터 받은 문제에 대한 정보를 관리
 - 추론과정의 중간결과를 저장하고, 유도된 최종해 저장
 - 작업 메모리에 저장되는 모든 것을 사실이라 함

규칙 기반 시스템 구조

❖ 추론 엔진(inference engine)

- 실행할 수 있는 규칙을 찾아서, 해당 규칙을 실행하는 역할
- 패턴 매칭 - 결합 해소 - 규칙 실행의 과정 반복

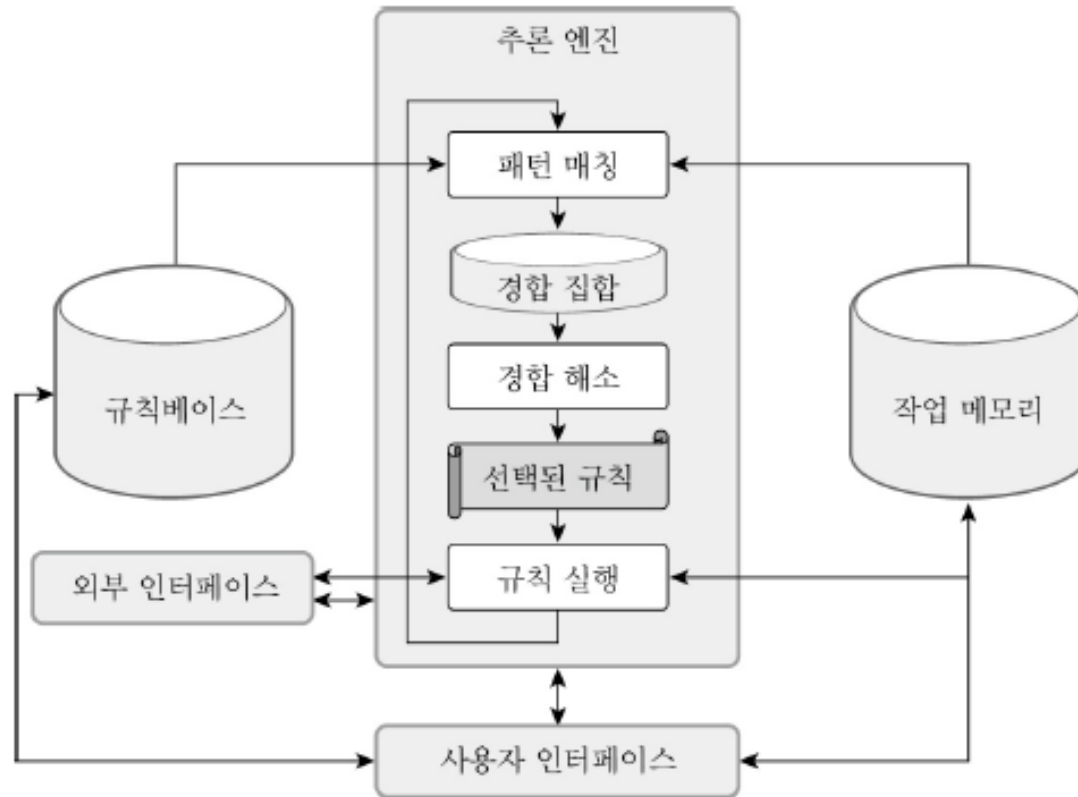
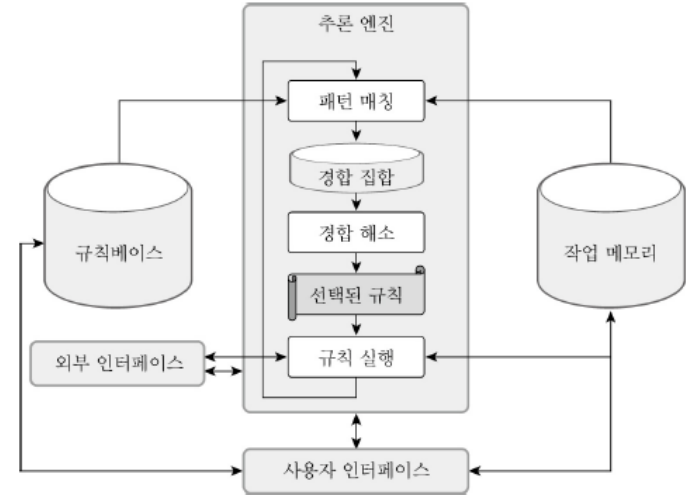


그림 3.26 규칙 기반 시스템의 구조

규칙 기반 시스템 구조

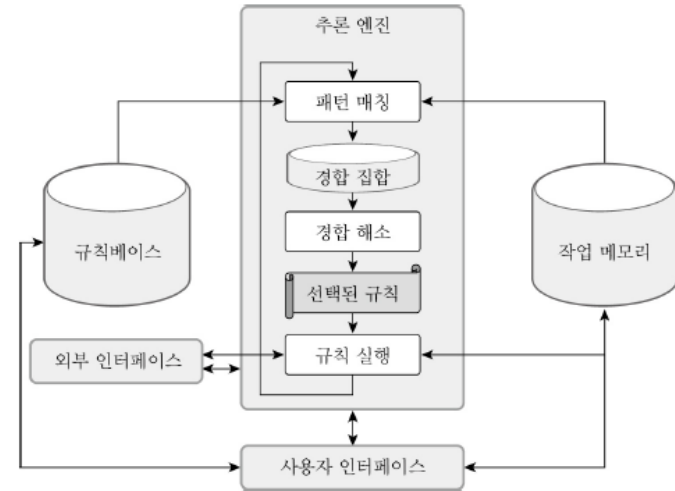
❖ 추론 엔진(inference engine)



- **패턴 매칭**(pattern matching)
 - 작업 메모리의 사실과 규칙베이스에 있는 규칙의 조건부를 대조하여 일치하는 규칙을 찾는 과정
- **경합 집합**(conflict set)
 - 규칙들의 집합, 실행 가능한 규칙들의 집합
- **경합 해소**(conflict resolution)
 - 경합 집합에서 하나의 규칙을 선택
- **사용자 인터페이스**(user interface)
 - 규칙베이스 및 작업 메모리 관리 및 추론 엔진 조작
- **외부 인터페이스**(external interface)
 - 외부 데이터나 함수의 기능 사용 지원

규칙 기반 시스템 구조

❖ 추론 엔진(inference engine)



▪ 경합 해소 전략

- **규칙 우선순위**(rule priority)
 - 미리 각 규칙에 우선순위 부여
 - 경합 집합에서 우선순위가 가장 높은 규칙 선택
- **최신 우선**(recency, depth)
 - 가장 최근에 입력된 데이터와 매칭된 규칙 선택
- **최초 우선**(first match, breath)
 - 경합 집합에서 가장 먼저 매칭된 규칙 선택
- **상세 우선**(specificity)
 - 가장 상세한 조건부를 갖는 규칙 선택
 - 규칙의 조건부가 가장 복잡하게 기술된 것 선택

규칙 기반 시스템

❖ 경합 해소 전략 – cont.

▪ 규칙 우선순위(rule priority)

- 규칙 1: 뇌막염 처방전 1 (우선순위 100)
IF 감염이 뇌막염이다
AND 환자가 어린이이다
THEN 처방전은 Number_1이다
AND 추천 약은 암피실린(ampicillin)이다
AND 추천 약은 겐타마이신(gentamicin)이다
AND 뇌막염 처방전 1을 보여준다
- 규칙 2: 뇌막염 처방전 2(우선순위 90)
IF 감염이 뇌막염이다
AND 환자가 어른이다
THEN 처방전은 Number_2이다
AND 추천 약은 페니실린(penicillin)이다
AND 뇌막염 처방전 2를 보여준다

규칙 기반 시스템

❖ 경합 해소 전략 – cont.

▪ 상세 우선(specificity)

- 가장 특수한 규칙 선택

• 규칙 1

IF 가을이다

AND 하늘이 흐리다

AND 일기예보에서는 비가 온다고 한다

THEN 조언은 '집에 머무르시오'

• 규칙 2

IF 가을이다

THEN 조언은 '우산을 가져가시오'

규칙 기반 시스템

❖ 경합 해소 전략 – cont.

- 최신 우선(recency, depth)

- 가장 최근에 입력된 데이터(data most recently entered) 사용 규칙 선택
- 각 사실에 시간 태그 부여

- 규칙 1

IF 일기예보에서는 비가 온다고 한다 [03/25 08:16 PM]
THEN 조언은 '우산을 가져가시오'

- 규칙 2

IF 비가 온다 [03/26 10:18 AM]
THEN 조언은 '집에 머무르시오'

규칙 기반 시스템

❖ 지식 표현

- 개발 도구에 따라 고유한 형식 사용
- 사실(fact)
 - 객체(object)나 프레임(frame)처럼 여러 개의 속성 포함 가능
 - 예. '이름이 멍키인 원숭이가 나이가 세 살이고 거실에 있다'
(**monkey (name 멍키) (age 3) (room 거실)**)

▪ 규칙

- Jess의 규칙 표현 예

```
(defrule ruleBirthday
  ?c <- (monkey (name cheetah) (age ?old) (room ?where) (birthdate ?day))
  (calendar (date ?day))
  =>
  (bind ?newAge (+ ?old 1))
  (retract ?c)
  (assert (monkey cheetah ?newAge ?where)))
```

규칙 기반 시스템

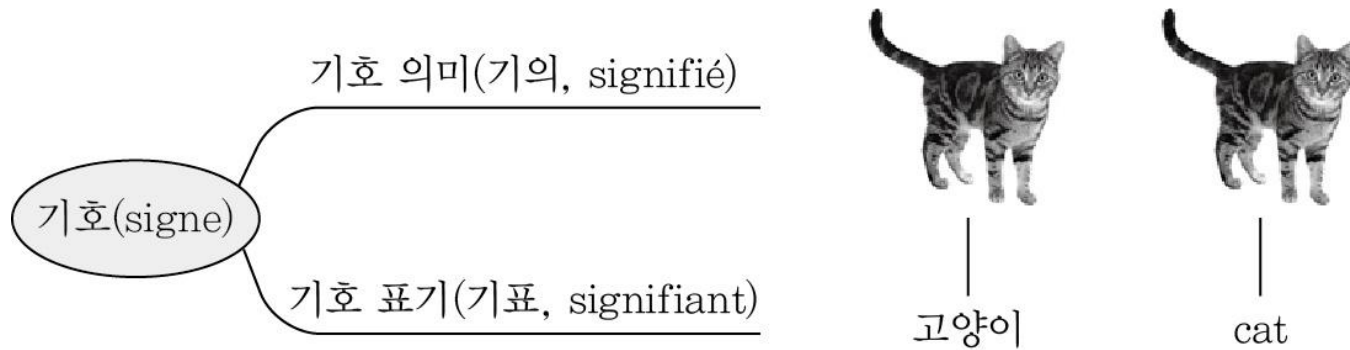
❖ 규칙 기반 시스템 개발 도구

- 규칙 기반 시스템의 기본 컴포넌트들을 미리 제공하여 규칙 기반 시스템을 쉽게 구현할 수 있게 하는 소프트웨어
- 문제 영역의 지식을 잘 획득하여 정해진 형태로 표현만 하면 규칙 기반 시스템을 비교적 쉽게 구현 가능
- **Jess**, CLIPS, EXSYS, JEOPS 등

11. 심볼 그라운드링 문제와 프레임 문제

❖ 심볼 그라운드링 문제

- '고양이는 귀엽다'라는 지식 표현
 - '고양이'와 '귀엽다'는 대상 또는 개념을 가리키는 기호(symbol) 사용



- 기호의 표기와 의미가 자의적인 관계
- 기호를 이해하는 문화 체계 속에서 필연화
- 심볼 그라운드링(symbol grounding)
 - 기호 표기를 실제 세계의 의미와 연결시키는 것

심볼 그라운드링 문제

❖ 심볼 그라운드링 문제 – cont.

- 기호 표기로 표현되어 있는 지식에 대해서, 컴퓨터는 심볼 그라운드링을 할 수 있는 능력이 없음
- 실제 세계와 컴퓨터의 기호 표기 사이의 심볼 그라운드링을 인간이 대신 수행
- 기호 표기를 실제 세계의 의미와 직접 연결시킬수 없다는 것
- 딥러닝 기술 발전은 심볼 그라운드링 문제에 해결에 기여 예상

프레임 문제

❖ 프레임(frame) 문제

- 사고범위 문제(思考範圍問題)
- 어떤 작업을 수행할 때 관계있는 지식만 꺼내서 사용한다는 것은 지극히 자연스럽게 당연하지만, 인공지능에서는 이러한 일이 쉽지 않다는 것

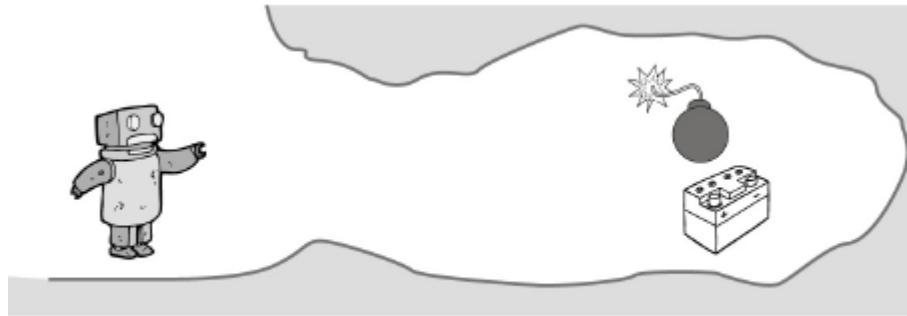


그림 3.28 프레임 문제

12. CYC 프로젝트

❖ 상식의 필요성

- 추론 등을 위해 상식의 활용 중요
 - 예. 기계번역

He saw a girl in the garden with a telescope.

그는 망원경으로 정원에 있는 소녀를 보았다.

그는 정원에서 망원경으로 소녀를 보았다.

그는 소녀가 정원에서 망원경을 들고 있는 것을 보았다.

정원에서 그는 망원경을 들고 있는 소녀를 보았다.

CYC 프로젝트

❖ CYC 프로젝트

- 상식적인 추론을 하는데 필요한 방대한 지식을 추출하여 표현하는 프로젝트
- 일차 술어 논리를 사용 지식 표현
- 1984년 르넷(Douglas Lenat) 시작

- 예. CYC의 예

(`#$isa #$DonaldTrump #$UnitedStatesPresident`) ; 도널드트럼프는 미국 대통령이다.

(`#$capitalCity #$France #$Paris`) ; 프랑스 수도는 파리이다.

(`#$implies (#$and (#$isa ?OBJ ?SUBSET) (#$genls ?SUBSET ?SUPERSET))`)

(`#$isa ?OBJ ?SUPERSET`) ; OBJ가 SUBSET의 사례이고, SUBSET이 SUPERSET에 속하면, OBJ는 SUPERSET의 사례이다. (규칙표현의 예)

(`#$relationAllExists #$biologicalMother #$ChordataPhylum #$FemaleAnimal`)

; 모든 ChordataPhylum에 속하는 개체에게는 어머니(biologicalMother)인

; 여성(FemaleAnimal)이 있다. (존재한정사가 있는 문장)

요약

1. 지식
2. 규칙
3. 프레임
4. 논리
5. 의미망
6. 스크립트
7. 온톨로지
8. 함수에 의한 지식 표현
9. 불확실한 지식 표현
10. 규칙 기반 시스템
11. 심볼 그라운드링 문제와 프레임 문제
12. CYC 프로젝트

요약

- ❖ 지식
- ❖ 규칙
- ❖ 프레임
- ❖ 논리
 - 명제 논리, 술어 논리
- ❖ 의미망
- ❖ 스크립트
- ❖ 온톨로지
 - 온톨로지, 시맨틱웹, RDF, RDFS, OWL
- ❖ 함수에 의한 지식 표현
- ❖ 불확실한 지식 표현
 - 확률, 확신도, 퍼지이론
- ❖ 확률 그래프 모델
 - 베이지언 망, 마르코프 랜덤 필드, 조건부 랜덤 필드, 로그-선형 모델
- ❖ 규칙 기반 시스템
- ❖ 심볼 그라운드링 문제와 프레임 문제
- ❖ CYC 프로젝트