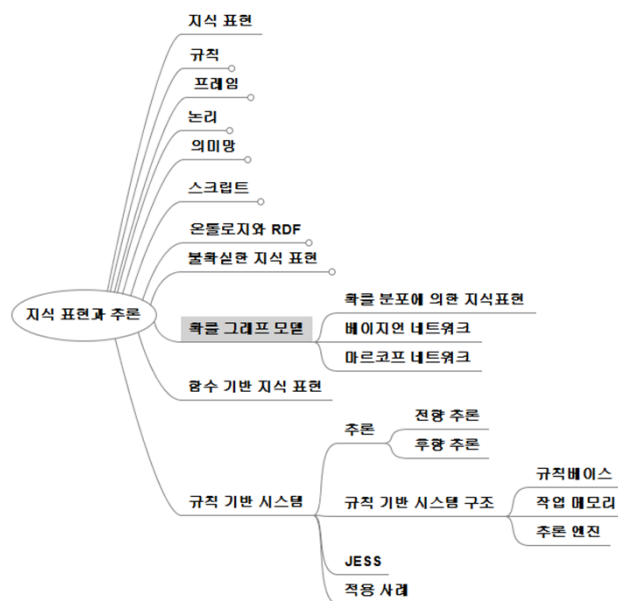


# 지식표현과 추론

## Part IV

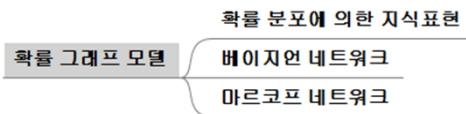
충북대학교 소프트웨어학과  
이건명



## 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 계산할 수 있는 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)



Image : www.thefreedictionary.com

## 확률 그래프 모델

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

E	B	A	N	확률
F	F	F	F	0.01
F	F	F	T	0.04
F	F	T	F	0.05
F	F	T	T	0.01
F	T	F	F	0.02
F	T	F	T	0.07
F	T	T	F	0.2
F	T	T	T	0.1
T	F	F	F	0.01
T	F	F	T	0.07
T	F	T	F	0.13
T	F	T	T	0.04
T	T	F	F	0.06
T	T	F	T	0.05
T	T	T	F	0.1
T	T	T	T	0.05



경보가 울릴 때 이웃이 전화할 확률은?

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

이웃이 전화했을 때 도둑이 들었을 확률은?

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

Image : www.thefreedictionary.com

## 확률 그래프 모델

### ❖ 조건부 독립과 확률분포의 인수분해(factorization)

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

## 확률 그래프 모델

### ❖ 조건부 독립을 이용한 확률분포의 인수 분해

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

만족하는 조건부 독립 성질

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

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

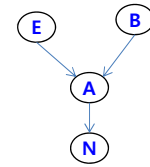
$$= P(N|A)P(A|E, B)P(E)P(B)$$

A	N	
	F	T
F	0.9	0.1
T	0.2	0.8

E	B	A	
		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

E	
F	T
0.9	0.1

B	
F	T
0.7	0.3



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

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

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

$$= 0.1944$$

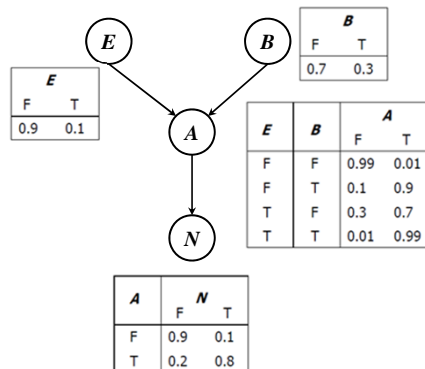
## 확률 그래프 모델

### ❖ 베이저안 네트워크(Bayesian network)

- 조건부 확률의 곱으로 표현된 확률분포를 방향성 그래프(directed graph)로 표현한 것

- 노드 : 확률 변수
- 에지 : 의존관계

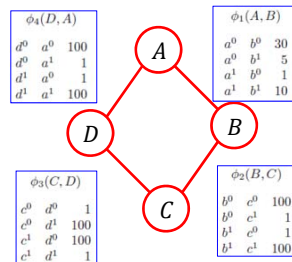
$$P(N, A, E, B) = P(N|A, E, B)P(A|E, B)P(E|B)P(B) = P(N|A)P(A|E, B)P(E)P(B)$$



## 확률 그래프 모델

### ❖ 마르코프 네트워크(Markov network)

- 확률분포를 무방향 그래프(undirected graph)를 사용하여 표현
- 마르코프 랜덤필드(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

## 10. 함수 기반 지식 표현

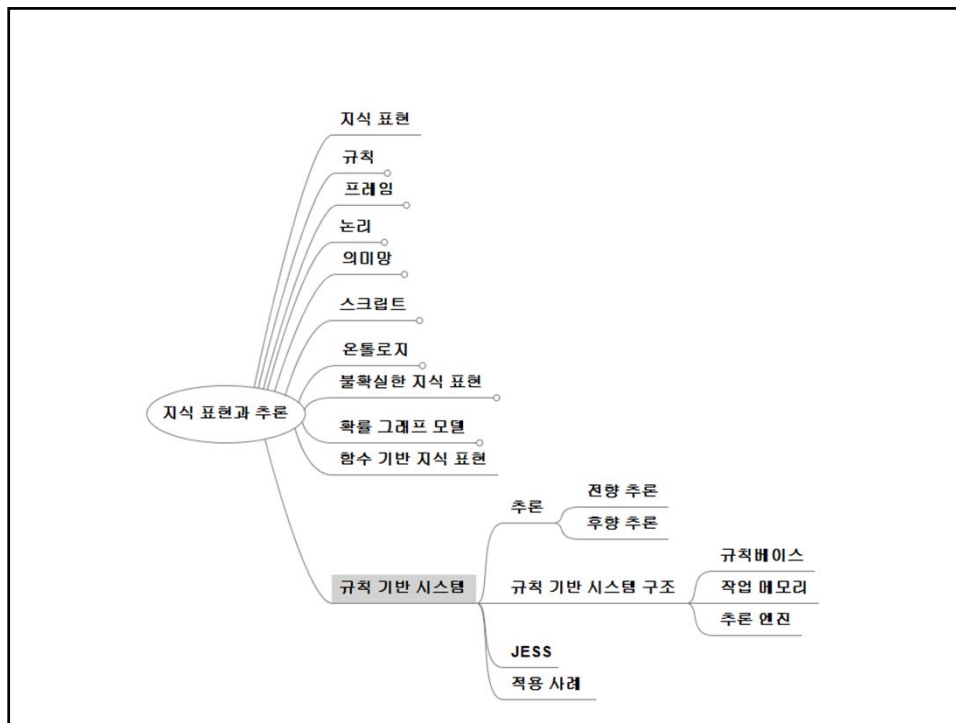
### ❖ 함수 기반의 지식

- 기호 대신 수치값과 수치값을 계산하는 함수를 사용하여 지식을 표현

$$y = f(x)$$

$$y = \frac{1}{1 + e^{-0.5 + 1.2x_1 - 0.3x_2 + 2.4x_3}}$$

- 회귀분석(regression analysis)
- 신경망(neural network)
- SVM(support vector machine)
- 딥러닝(deep learning)



## 11. 규칙 기반 시스템

### ❖ 규칙 기반 시스템(rule-based system)

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

## 11.1 추론

### ❖ 추론

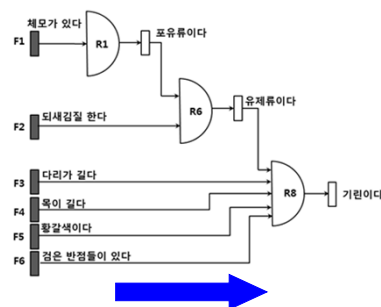
- 구축된 지식과 주어진 데이터나 정보를 이용하여 새로운 사실을 생성하는 것
- **전향 추론(forward chaining, 순방향 추론)**
  - 규칙의 조건부와 만족하는 사실이 있을 때 규칙의 결론부를 실행하거나 처리
- **후향 추론(backward chaining, 역방향 추론)**
  - 어떤 사실을 검증하거나 확인하고 싶은 경우에 관심 대상 사실을 결론부에 가지고 있는 규칙을 찾아서 조건부의 조건들이 모두 만족하는지 확인

## 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 되새김질을 한다 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?

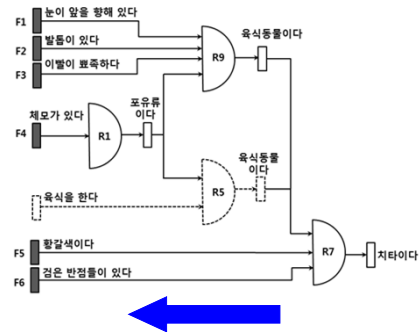


## 후향 추론의 예

- 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 향해있다 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.

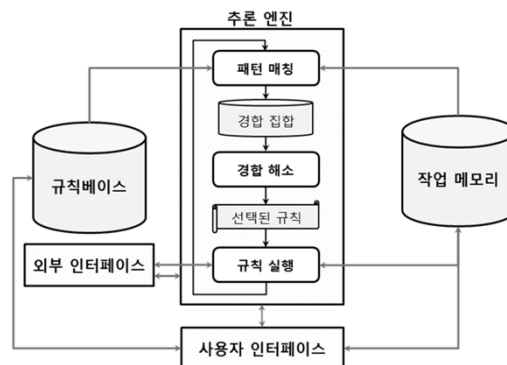
스프린터는 치타인가?



## 11.2 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

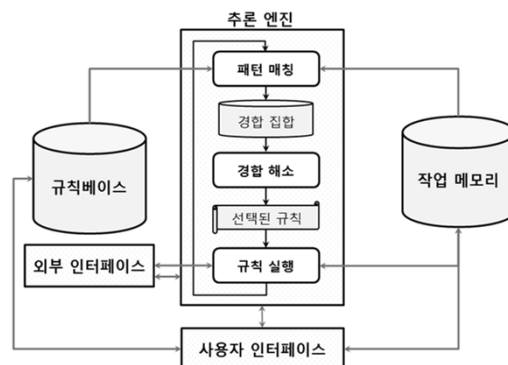
### ❖ 규칙 기반 시스템 구조

- 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 **사실**이라 함

## 규칙 기반 시스템 구조

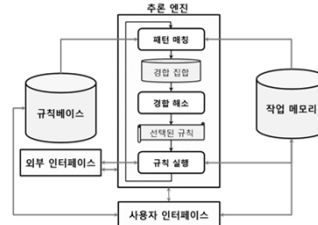
### ❖ 추론 엔진(inference engine)

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



## 규칙 기반 시스템 구조

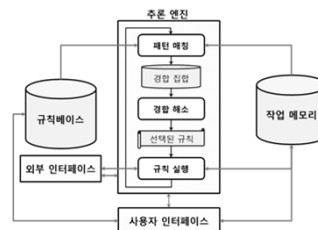
### ❖ 추론 엔진(inference engine)



- **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 경합 집합에 포함된 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 조연은'집에 머무르시오'
- 규칙 2
  - IF 가을이다
  - THEN 조연은'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 조연은 '우산을 가져가시오'

- 규칙 2

IF 비가 온다 [03/26 10:18 AM]  
THEN 조연은 '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

### ❖ JESS (Java Expert System Shell)

- 규칙 기반 전문가 시스템의 개발 환경
- 상용이지만, 비영리 목적으로는 무료 사용 가능
- Rete 알고리즘을 구현한 매우 빠른 경량(lightweight) 규칙 엔진 사용
- Jess의 규칙에서 **Java의 API 사용** 가능
- **Java 프로그램**에 Jess를 컴포넌트로 **사용**한 지식 기반 서비스 개발 가능
- Eclipse 환경과 통합된 편리한 개발환경 제공

## 규칙 기반 시스템

### ❖ JESS – cont.

- Java에서 Jess engine 사용 가능

```
import jess.*;

public class PricingEngine {
    private Rete engine;
    private WorkingMemoryMarker marker;
    private Database database;

    public PricingEngine(Database aDatabase) throws JessException {
        // Create a Jess rule engine
        engine = new Rete();
        engine.reset();

        // Load the pricing rules
        engine.batch("pricing.clp");

        // Load the catalog data into working memory
        database = aDatabase;
        engine.addAll(database.getCatalogItems());

        // Mark end of catalog data for later
        marker = engine.mark();
    }
    ...
}
```

## 규칙 기반 시스템 적용 사례

### ❖ 게임

- NPC (non-player character) 동작 제어



IF enemy visible and my health is < very-low-health-value (20%)  
 OR  
 his weapon is much better than mine  
 THEN propose retreat

image : Quake 2

## 규칙 기반 시스템 적용 사례

### ❖ 기업 정보 시스템

- 현업에서 비즈니스 로직(business logic)에 대응하는 업무 규칙(Business Rule)을 시스템에 직접 등록함으로써 새로운 업무규칙 적용 가능
  - 프로그램으로 코딩을 할 경우,  
 프로그래밍 부담, 테스트 및 재설치 비용
- 시간이 대폭 감소
- 비즈니스 로직 변경에 따라 오류 최소화
- 신속한 시장 대응 가능

