Gyeongsang National University

2024 1st Semester

Artificial Intelligence

6. Logical Agents (2)

(지식표현과 추론 (2))

IDEALAB

Improving lives through learning

Gun-Woo Kim
School of Computer Science & Engineering
Gyeongsang National University (GNU)

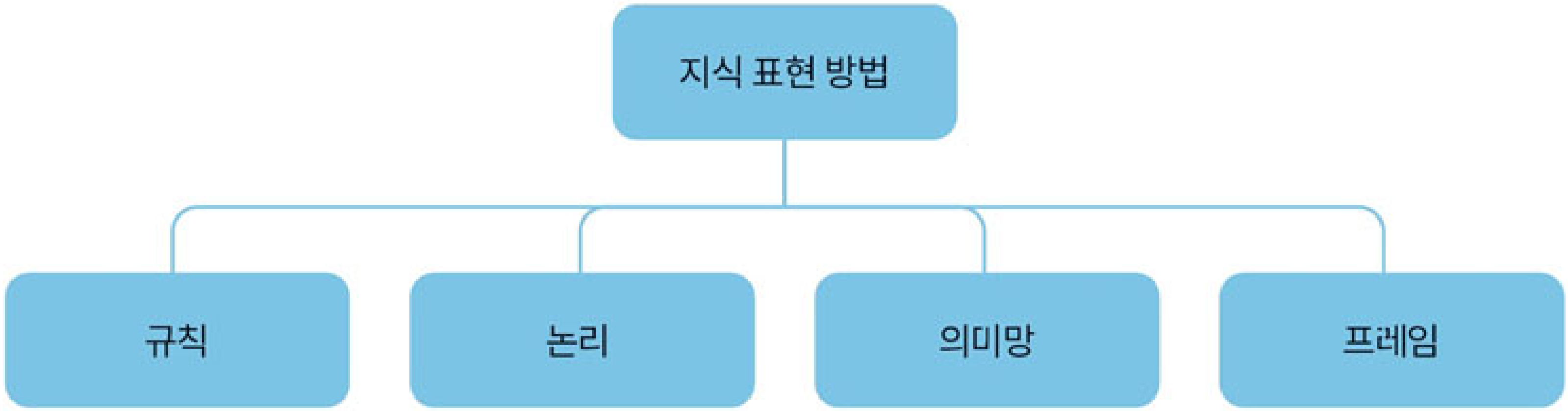
경상국립대학교 Gyeongsang National University

Contents

- Recap (지난주)
- 명제 논리 (Propositional Logic)
- 1차 논리 (First-order Logic)

Recap 지식기반 에이전트

- 지식베이스 에이전트(Knowledge Base Agent)
  - 지식베이스(KB): 형식언어(Formal Language)로 구성된 문장(sentence)들의 집합
    - Knowledge base = a set of sentences in a formal language



- 절차적 지식(procedural knowledge)
  - 지식베이스(KB)에 저장 및 학습(updating)된 상태들을 기반으로 추론엔진을 통해 동작 수행
- 선언적 지식(declarative knowledge)
  - 어떤 대상의 성질, 특성이나 관계 서술 (e.g. 논리, 의미망, 프레임)

Recap 지식기반 에이전트

- 지식베이스 에이전트(Knowledge Base Agent)
  - 지식베이스(KB) 에이전트 Pseudocode

```
function KB-AGENT(percept) returns an action
    static: KB, a knowledge base
            t, a counter, initially 0, indicating time

    TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
    action ← ASK(KB, MAKE-ACTION-QUERY(t))
    TELL(KB, MAKE-ACTION-SENTENCE(action, t))
    t ← t + 1
    return action
```

1) 에이전트는 인식한 현재 상태를 KB에게 알려줌 (TELL)

2) 이후 에이전트는 KB에게 해당 time에 어떠한 동작(action)을 취해야하는지 물어봄 (ASK)

→ 여기서 동작에 대한 Answer(action)는 KB에 저장되어 있는 것

3) 이후 에이전트는 KB에게 해당 time에 어떠한 동작(action)을 선택했는지 알려줌 (TELL)

→ 즉 에이전트는 KB로부터 선택된 동작들 중 추론 엔진을 통해 Best한 동작 선정

→ Counter를 통해 해당 동작이 몇 번째로 수행되었는지 기록을 위해 time update 실시

Recap 규칙 (Rule)

- 규칙(Rule)
  - IF 부분
    - 주어진 정보나 사실에 대응될 조건
    - 조건부(conditional part, antecedent)
    - 둘 이상의 조건을 AND 또는 OR로 결합하여 구성 가능
      - IF <조건1> AND <조건2> AND <조건3> THEN <결론>
      - IF <조건1> OR <조건2> OR <조건3> THEN <결론>
  - THEN 부분
    - 조건부가 만족될 때의 판단이나 행동
    - 결론부(conclusion, consequent)
    - 여러 개의 판단 또는 행동 포함 가능

- IF <조건>
  THEN <결론1>
      AND <결론2>
      AND <결론3>

Recap 논리 (Logic)

- 논리 종류 (Types of Logic)
  - 명제 논리 (Propositional Logic)
    - 명제를 기호로 표시하고, True/False를 판정하는 논리
      - Syntax: $P \lor (\neg Q \land R)$; $X_1 \Leftrightarrow (\text{Raining} \Rightarrow \text{Sunny})$
      - Possible world: {P=true,Q=true,R=false,S=true} or 1101
      - Semantics: $\alpha \land \beta$ is true in a world iff α is true and β is true (etc.)
  - 술어 논리 (First-order Logic, 1차논리)
    - 명제의 내용을 다루기 위해 변수, 함수 등을 도입하고, 이들 값에 따라 True/False를 판정하는 논리 (명제 논리 확장)
      - Syntax: $\forall x\, \exists y\, P(x,y) \land \neg Q(Joe,f(x)) \Rightarrow f(x)=f(y)$
      - Possible world: Objects o_1, o_2, o_3; P holds for $<o_1,o_2>$; Q holds for $< o_1, o_3>$; $f(o_1)=o_1$; $Joe=o_3$; etc.
      - Semantics: $\phi(\sigma)$ is true in a world if $\sigma=o_j$ and ϕ holds for o_j; etc.

Recap 논리 (Logic)

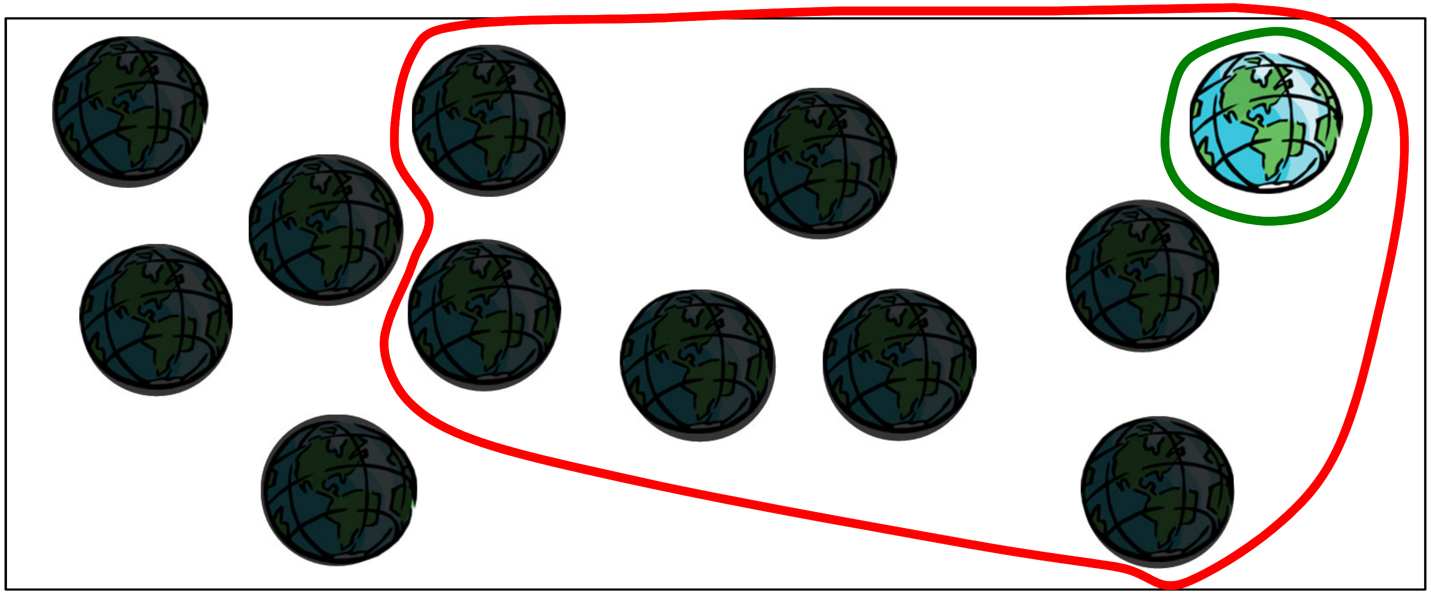
- 함의 (Entailment)
  - Entailment: $\alpha \vDash \beta$ ("α entails β" or "β follows from α")
    - 논리에서는 문장(sentence)의 logic 결과(True/False)가 다른 문장의 결과를 따를 수 있음
    - α가 β를 함의한다 = β는 α의 결과를 따른다
    - 즉 만약 α가 "true"이면 β도 "true"이다.

이때 α는 β의 subset 관계임 [*models*(α) $\subseteq$ *models*(β)]

$\alpha \vDash \beta$
(Say α is $\neg Q \wedge R \wedge S \wedge W$)
β is $\neg Q$)

β

α



Recap 논리 (Logic)

- 추론 (Inference)
  - KB $\vdash_i \alpha$ = 문장(sentence) α 는 추론 절차/알고리즘 i (inference procedures)를 통해 KB로 부터 결과가 도출 될 수 있음

→ 즉 α 는 추론 절차 i에 의해 KB로 부터 결과를 유도할 수 있음

e.g. "사막에서 바늘찾기" (Needle in haystack(건초))

- Haystack: KB의 결과 (무한함)
- Needle: α
- Entailment: Haystack에 있는 Needle
- inference: finding it(방법을 통해 찾아라!!)

- Soundness(건전성)

→ 추론 절차 i 가 sound하다는 것은, '만약 추론 절차 i를 통해 도출된 KB $\vdash_i \alpha$ 가 True이면 KB |= α 의 결과는 항상 True이다'를 보장할 수 있는 것

- Completeness(완전성)

→ 추론 절차 i 가 complet하다는 것은, 만약 KB |= α 가 True일 때 추론 절차 i를 통해 KB $\vdash_i \alpha$ 의 모든 결과도 항상 True임을 보장할 수 있는 것

Recap 명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - 명제기호 진리표 (Truth Table)

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

$\neg S$	is true iff	S	is false		
$S_1 \wedge S_2$	is true iff	S_1	is true **and**	S_2	is true
$S_1 \vee S_2$	is true iff	S_1	is true **or**	S_2	is true
$S_1 \Rightarrow S_2$	is true iff	S_1	is false **or**	S_2	is true
i.e.,	is false iff	S_1	is true **and**	S_2	is false
$S_1 \Leftrightarrow S_2$	is true iff	$S_1 \Rightarrow S_2$	is true **and**	$S_2 \Rightarrow S_1$	is true

  - Wumpus World 예제 대입 (e.g. P는 Pit를 의미, $P_{1,2}$는 (1,2)칸에 Pit가 있다.)

$$\neg P_{1,2} \wedge (P_{2,2} \vee P_{3,1}) = true \wedge (false \vee true) = true \wedge true = true$$

Recap 명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Equivalence Proof

Example: Show that $\neg(p \vee (\neg p \wedge q))$ is logically equivalent to $\neg p \wedge \neg q$

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - 명제논리 표현 (Translating English Sentences)
    - Steps to convert an English sentence to a statement in propositional logic
      - Identify atomic propositions and represent using propositional variables
      - Determine appropriate logical connectives

- "If I go to Harry's or to the country, I will not go shopping."
  - p: I go to Harry's
  - q: I go to the country.
  - r: I will go shopping.

If p or q then not r.

$$(p \vee q) \rightarrow \neg r$$

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - 명제논리 표현 (Translating English Sentences)
    - Example (Student)

"You can access the Internet from campus only if you are a computer science major or you are not a freshman."

One Solution: Let *a*, *c*, and *f* represent respectively "You can access the internet from campus," "You are a computer science major," and "You are a freshman."

$$a \rightarrow (c \vee \neg f)$$

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - 명제논리 표현 (Translating English Sentences)
    - Example (System Specifications)
      - System and Software engineers take requirements in English and express them in a precise specification language based on logic

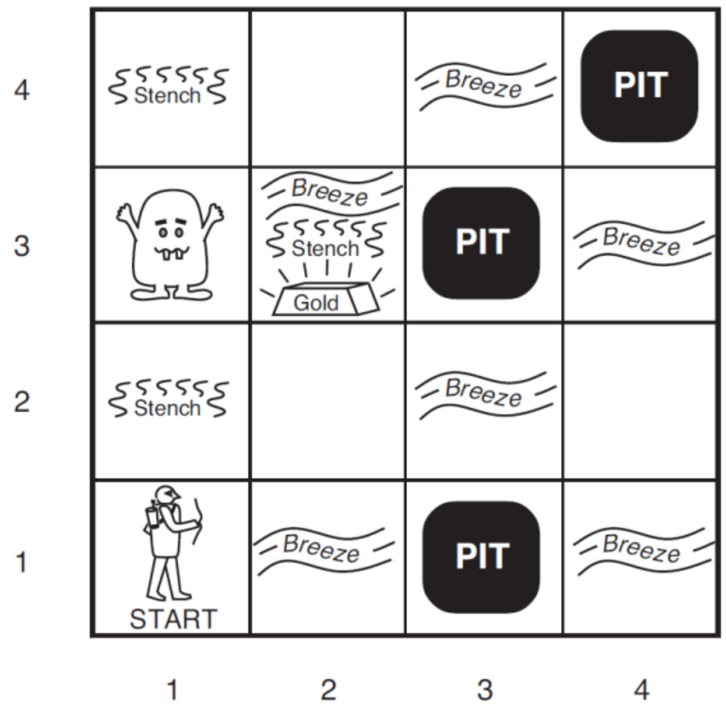
"The automated reply cannot be sent when the file system is full"

Solution: One possible solution: Let p denote "The automated reply can be sent" and q denote "The file system is full."

$$q \rightarrow \neg p$$

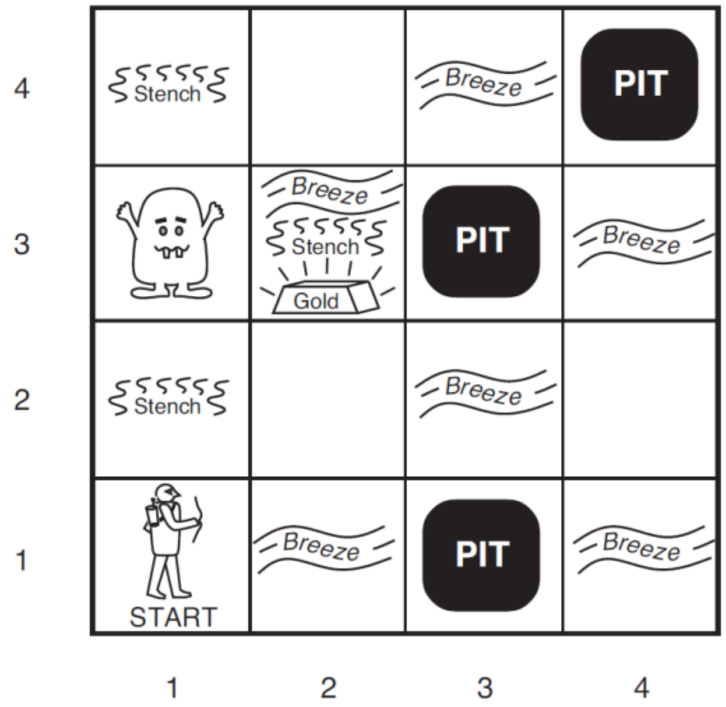
명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - 지식베이스(KB) 표현 (Wumpus world case)
    - $P_{x,y}$ = True
      - 만약 [x, y]지점에 Pit(구덩이)가 있을 경우 True이다
    - $B_{x,y}$ = True
      - 만약 [x, y]지점에서 Agent가 Breeze(미풍)을 느낄 경우 True이다
    - $S_{x,y}$ = True
      - 만약 [x, y]지점에서 Agent가 Stench(악취)를 느낄 경우 True이다
    - $W_{x,y}$ = True
      - 만약 [x, y]지점에 Wumpus (dead or alive)가 있을 경우 True이다



▪ 명제 논리 (Propositional Logic)

- 지식베이스(KB) 표현 (Wumpus world case)
  - [1,1]에는 Pit 가 없음
    - Rule 1 : $\neg P_{1,1}$
  - 만약 Pit에 인접한 칸이라면 Breeze를 감지
    - Rule 2 : $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
    - Rule 3 : $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
  - Breeze와 관련된 정보는 (1,1), (2,1)에서 감지되었음
    - Rule 4 : $\neg B_{1,1}$
    - Rule 5 : $B_{2,1}$



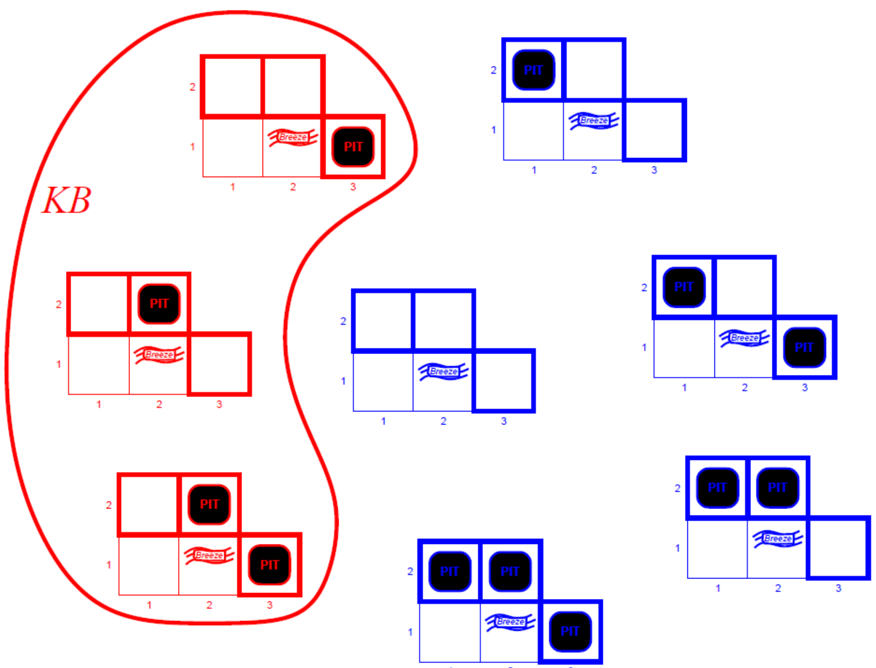
∴ KB는 $B_{1,1}, B_{2,1}, P_{1,1}, P_{1,2}, P_{2,1}, P_{2,2}, P_{3,1}$ (총 7개)에 대한 정보를 알고 있으며,
총 2^7 = 128개 조합 가능 (중복순열)

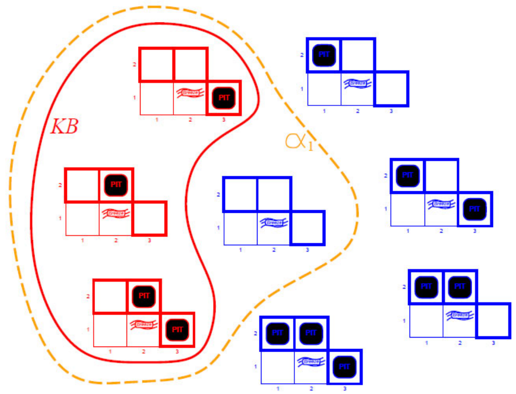
→ 이러한 결과들로 KB ⊨ $\neg P_{1,2}$ 를 entailment하는지 알 수 있을까?

(즉 KB가 true이면, $\neg P_{1,2}$ 도 true임을 추론할 수 있는가?)

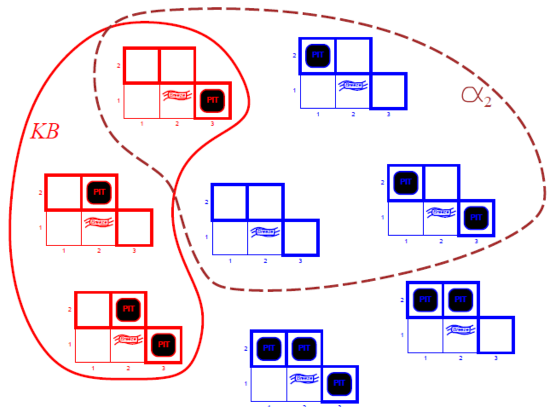
명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - 지식베이스(KB) 표현 (Wumpus world case)





명제 α1 = "[1,2]은 Safe함!!", KB |= α1, entailment 관계 형성



명제 α2 = "[2,2]는 safe함!!", KB |≠ α2

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Truth Table for Inference (Wumpus world case)

KB ⊨ ¬$P_{1,2}$ 를 entailment하는지 알 수 있을까?

- KB가 true이면, ¬$P_{1,2}$ 도 true임을 추론할 수 있는가?

- KB가 true이면, $P_{2,2}$ 도 true임을 추론할 수 있는가?

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	*false*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*false*	*false*
false	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*false*	*true*	*false*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*false*	*true*	*true*	*false*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*true*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*true*	*true*	*false*	*true*	*false*

2^7 = 128개

지식베이스 영역

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Inference Rules (추론 규칙)
    - 추론 규칙은 우리가 가지고 있는 지식(KB)과 이미 알고있는 사실(Observation)으로 부터 새로운 사실을 유추해내는 것 (결과에 대한 증명(Proof)을 이끌어 내는데 사용 가능)

1) Modus Ponens (전건긍정)

: A가 True인 것으로 정의하면, B도 True라는 결론을 얻는 논리 규칙

2) Modus Tollens (부정논법)

: 규칙의 결론이 False인 경우 해당 조건 또한 False로 추론하는 것

3) AND-Elimination (AND조건의 삭제)

: A가 True이며 A $\wedge$ B의 결론이 True인 경우, B 또한 True로 추론하는 것

4) Syllogism (삼단논법)

: 두개의 규칙을 연쇄적으로 작용시켜 새로운 규칙을 도출해내는 추론 방법

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Inference Rules (추론 규칙)
    - Modus Ponens (전건긍정)

      : A가 True인 것으로 정의하면, B도 True라는 결론을 얻는 논리 규칙

규칙 / A → B
사실 / A

결론 B

"홍길동이 세계 일주 중이라면 > 로또에 당첨된 것이다. "
"홍길동은 세계 일주 중이다."

∴ "홍길동은 로또에 당첨된 것이다. "

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Inference Rules (추론 규칙)
    - Modus Ponens (전건긍정)

Example)

$$\begin{array}{l} p \to q \\ p \\ \hline \therefore q \end{array}$$

Corresponding Tautology:

$(p \wedge (p \to q)) \to q$

p: Let p be "It is raining"

q: Let q be "I will study artificial intelligence"

p → q: If it is raining, then i will study artificial intelligence

"it is raining"

"Therefore, I will study artificial intelligence"

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Inference Rules (추론 규칙)
    - Modus Tollens (부정논법)

      : 규칙의 결론이 False인 경우 해당 조건 또한 False로 추론하는 것

규칙 / A → B
사실 / NOT B

결론 NOT A

"어떤 동물이 강아지라면 > 어떤 동물은 4개의 다리를 가지고 있다. "
"어떤 동물은 4개의 다리를 가지고 있지 않다. "

∴ "어떤 동물은 강아지가 아니다. "

- 명제 논리 (Propositional Logic)
  - Inference Rules (추론 규칙)
    - Modus Tollens (부정논법)

Example)

$$\begin{array}{l} p \rightarrow q \\ \neg q \\ \hline \therefore \neg p \end{array}$$

Corresponding Tautology:
$(\neg p \wedge (p \rightarrow q)) \rightarrow \neg q$

p: Let p be "It is raining"

q: Let q be "I will study artificial intelligence"

p → q: If it is raining, then i will study artificial intelligence

"I will not study artificial intelligence"

"Therefore, it is not raining"

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Inference Rules (추론 규칙)
    - AND-Elimination (AND조건의 삭제, Identify raw(항등법칙))

      : A가 True이며 A ∧ B의 결론이 True인 경우, B 또한 True로 추론하는 것

규칙 / A ∧ B
사실 / A

결론   B

"어떤 동물이 강아지이고 4개의 다리를 가지고 있다. "
" 어떤 동물은 강아지이다"

∴ "강아지는 4개의 다리를 가지고 있다. "

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Inference Rules (추론 규칙)
    - AND-Elimination

Example)

$$\frac{p \wedge q}{\therefore q}$$

Corresponding Tautology:
$(p \wedge q) \rightarrow p$

p: Let p be "I will study artificial intelligence"

q: Let q be "I will study data structure"

p ∧ q: I will study artificial intelligence, and data structure

"Therefore, I will study data structure"

- 명제 논리 (Propositional Logic)
  - Inference Rules (추론 규칙)
    - Syllogism (삼단논법)

: 두개의 규칙을 연쇄적으로 작용시켜 새로운 규칙을 도출해내는 추론 방법

규칙 / A → B
사실 / B → C

결론 A → C

"소크라테스는 인간이다"
"인간은 모두 죽는다"
--
∴ "소크라테스는 죽는다"

- 명제 논리 (Propositional Logic)
  - Inference Rules (추론 규칙)
    - Syllogism (삼단논법)

Example)

$$\begin{array}{l} p \to q \\ q \to r \\ \hline \therefore p \to r \end{array}$$

Corresponding Tautology:
$((p \to q) \land (q \to r)) \to (p \to r)$

p: Let p be "It snows"

q: Let q be "I will study artificial intelligence"

r: Let r be "I will get an A"

p → q: If it snows, then I will study artificial intelligence

q → r: If I study artificial intelligence, I will get an A

"Therefore, If it snows, I will get an A"

- 명제 논리 (Propositional Logic)
  - Inference Rules (추론 규칙)
    - 어떻게 $\neg P_{1,2}$가 true인지 증명(Proof) 해보자 !

① Rule 2에 대한 biconditional elimination 적용

$$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$$

- Rule 6 : $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

② Rule 6에 AND-Elimination 적용

- Rule 7 : $((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

③ Rule 7에 Contrapositive(모순) 적용 (Logically equivalence)

$$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$$

- Rule 8 : $(\neg B_{1,1} \Rightarrow \neg (P_{1,2} \vee P_{2,1}))$

④ Rule 8에 Modus Ponens를 적용 (i.e., Rule 4 : $\neg B_{1,1}$ 임을 알 수 있으므로 뒷부분도 True))

- Rule 9 : $\neg (P_{1,2} \vee P_{2,1})$

⑤ Rule 9에 De Morgan 법칙 적용, 이후 AND-Elimination 적용

- Rule 10 : $\neg P_{1,2} \wedge \neg P_{2,1}$ $\neg P_{1,2}$

- Rule 1 : $\neg P_{1,1}$
- Rule 2 : $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
- Rule 3 : $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
- Rule 4 : $\neg B_{1,1}$
- Rule 5 : $B_{2,1}$

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Inference Rules (추론 규칙)
    - 증명은 아래의 3가지 방법을 사용

      1) Forward Chaining (전방 연쇄/순방향 추론)
         - 알려진 사실로 부터 출발하여 결론을 이끌어내는 방법
         - LHS to RHS
         - Data-Driven

      2) Backward Chaining (후방 연쇄/역방향추론)
         - 목표를 설정하고, 이를 증명하는 증거를 찾는 방법
         - RHS to LHS
         - Goal-Driven

      3) Resolution Refutation (Theorem Proving)
         - CNF(논리곱) 표준형, Resolution(분해법칙)을 적용
         - 모순(Contradiction) 또는 Refutation(반박)을 통해 결론을 이끌어내는 방법

Model checking 방법

추론 규칙 적용 방법

명제 논리 (Propositional Logic)

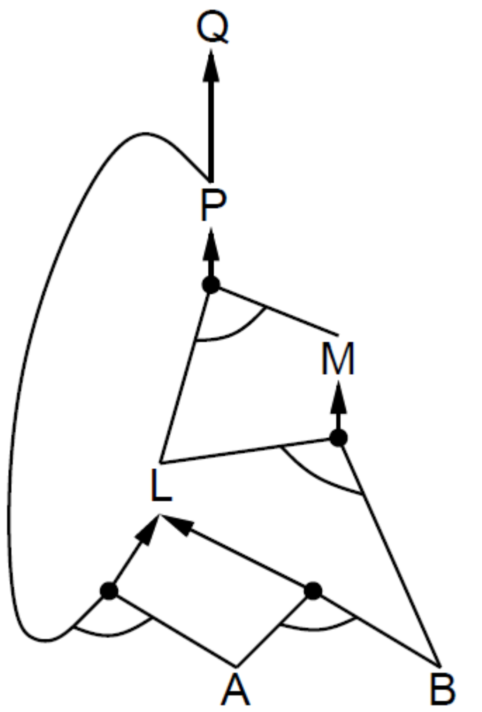
- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 1) Forward Chaining (전방 연쇄/순방향 추론)
      - 알려진 사실에 의거하여 KB에 있는 규칙들을 실행하고, 발견된 지식을 KB에 추가
      - A와 B라는 사실(Fact)를 알게 되면 $A \land B \Rightarrow L$ 라는 규칙을 통해 새로운 L이라는 사실을 알 수 있음

KB rules(지식베이스 규칙)

$P \Rightarrow Q$
$L \land M \Rightarrow P$
$B \land L \Rightarrow M$
$A \land P \Rightarrow L$
$A \land B \Rightarrow L$

사실(Fact (True))

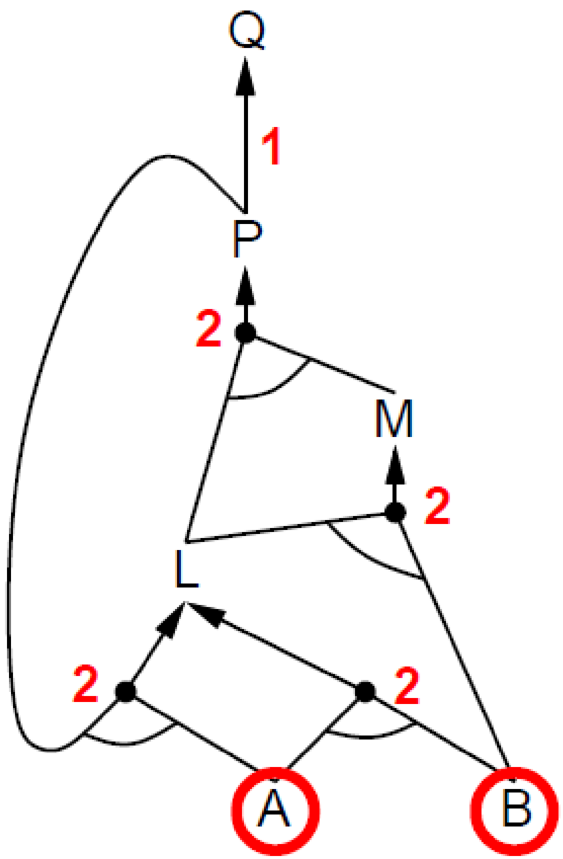
A
B



AND-OR 그래프

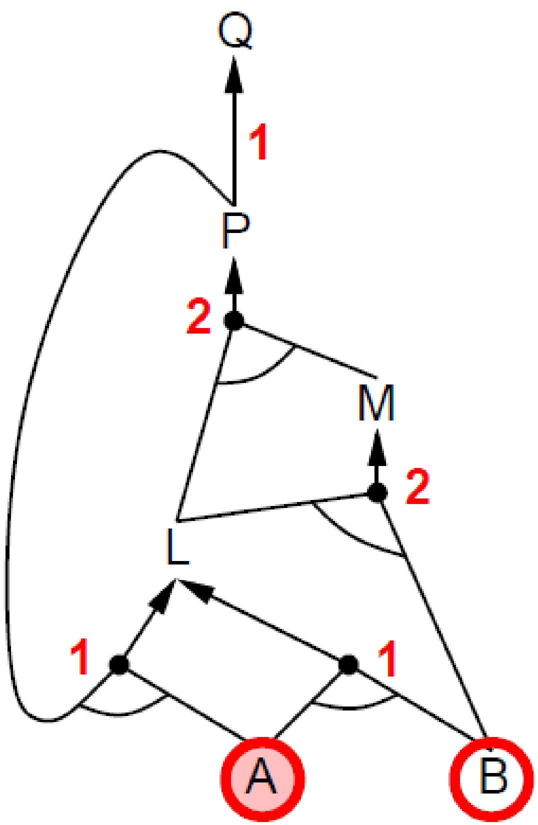
명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 1) Forward Chaining (전방 연쇄/순방향 추론)



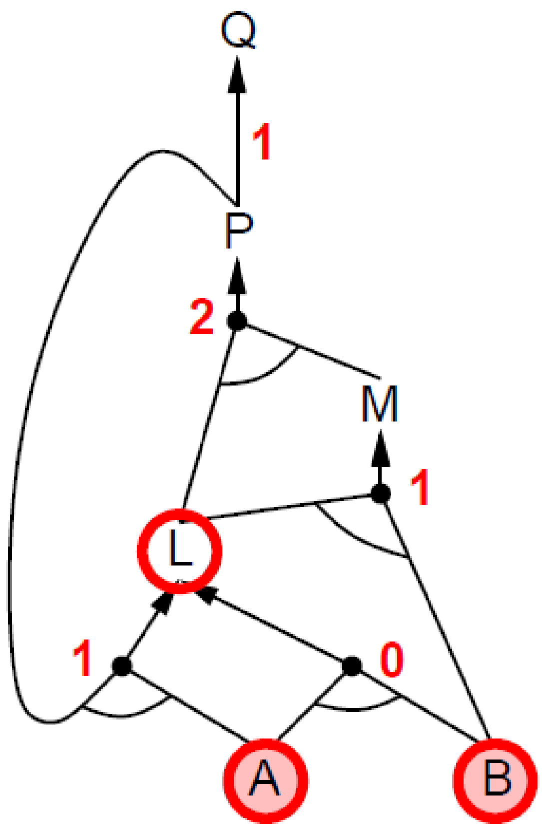
명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 1) Forward Chaining (전방 연쇄/순방향 추론)



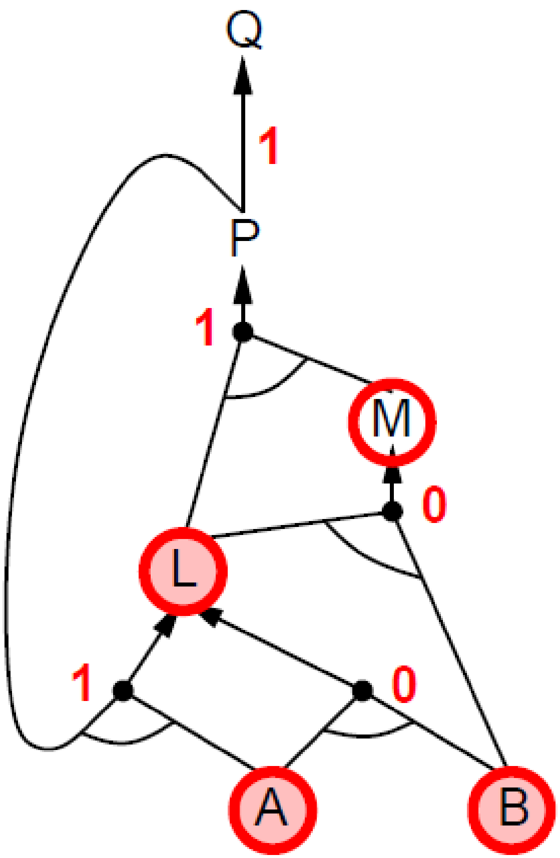
명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 1) Forward Chaining (전방 연쇄/순방향 추론)



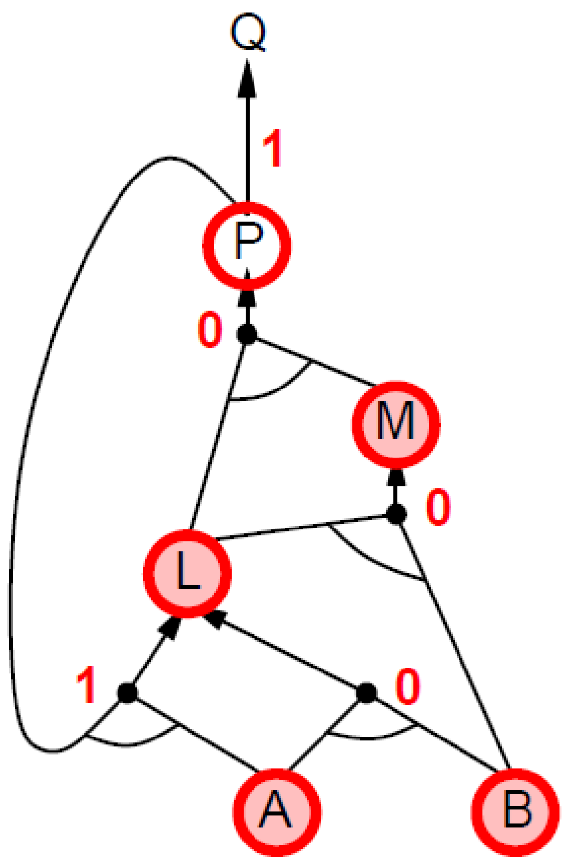
명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 1) Forward Chaining (전방 연쇄/순방향 추론)



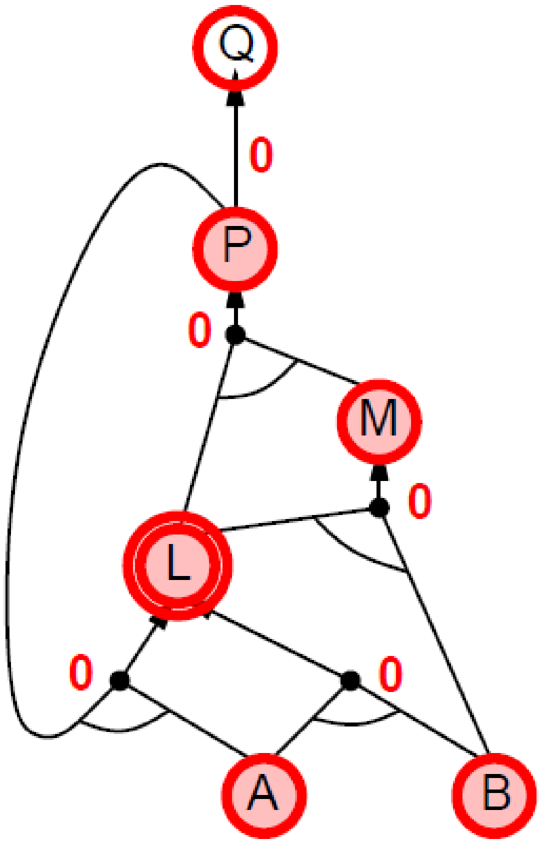
명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 1) Forward Chaining (전방 연쇄/순방향 추론)



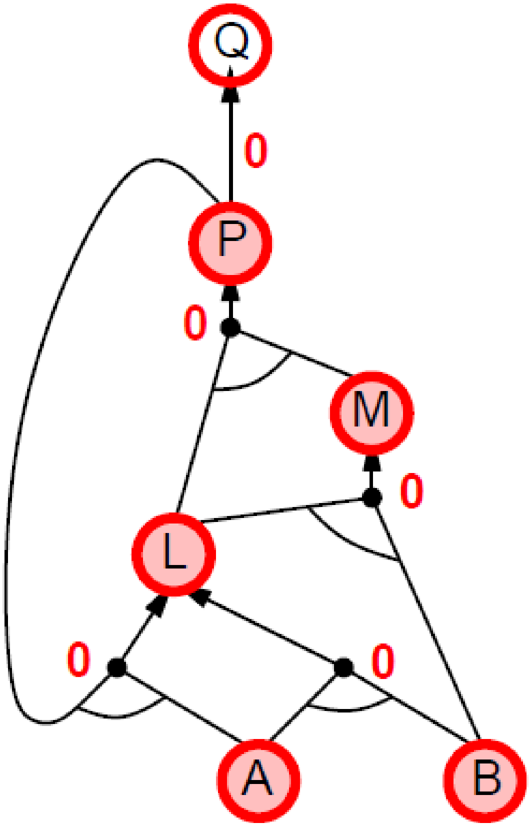
명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 1) Forward Chaining (전방 연쇄/순방향 추론)



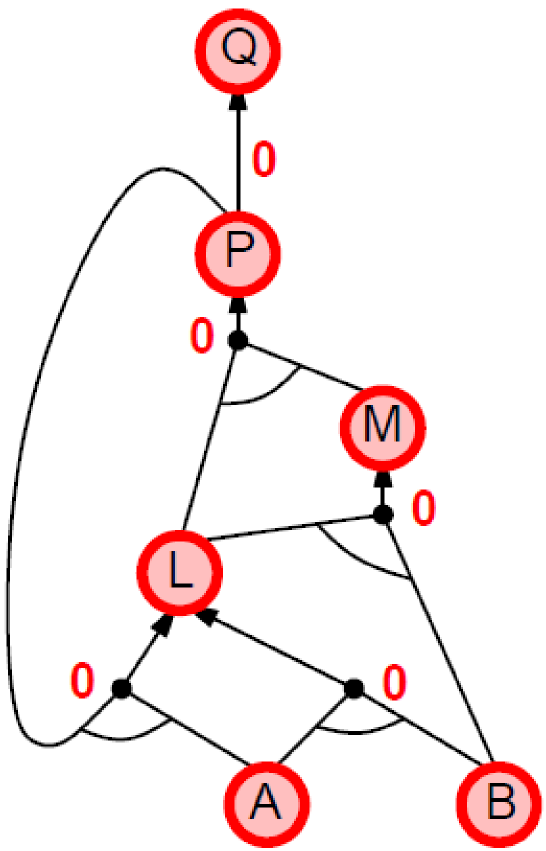
명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 1) Forward Chaining (전방 연쇄/순방향 추론)



명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 1) Forward Chaining (전방 연쇄/순방향 추론)



명제 논리 (Propositional Logic)

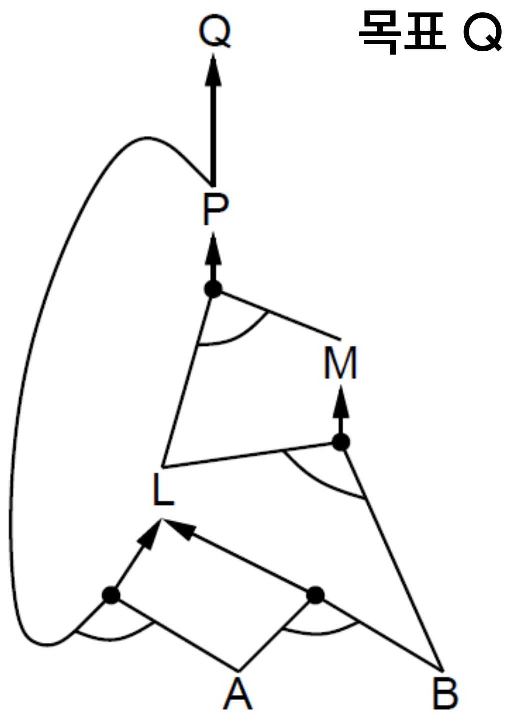
- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 2) Backward Chaining (후방 연쇄/역방향 추론)
      - 목표에서 시작하여 사실 데이터가 이러한 목표를 지원하는지 확인하는 방법

KB rules(지식베이스 규칙)

$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$

사실(Fact)

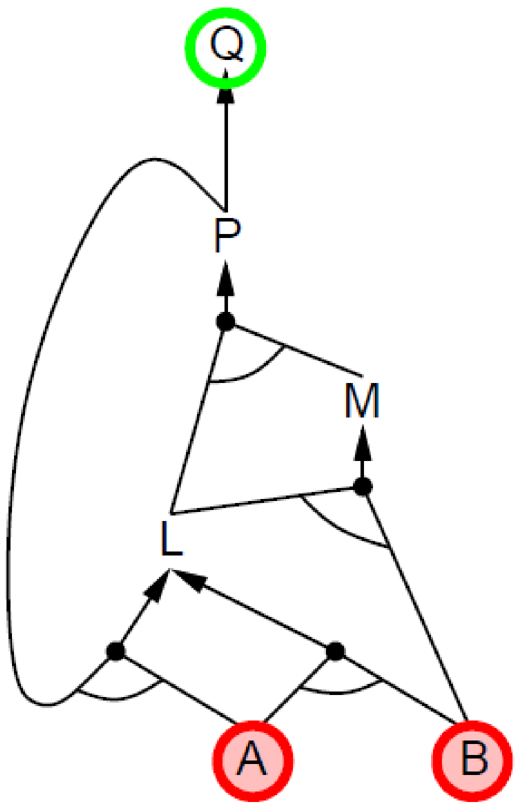
A
B



AND-OR 그래프

명제 논리 (Propositional Logic)

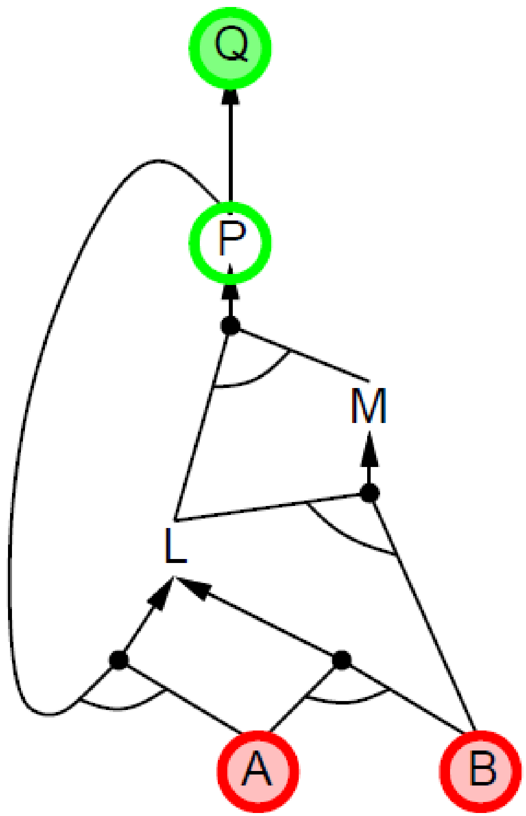
- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 2) Backward Chaining (후방 연쇄/역방향 추론)



목표 Q

명제 논리 (Propositional Logic)

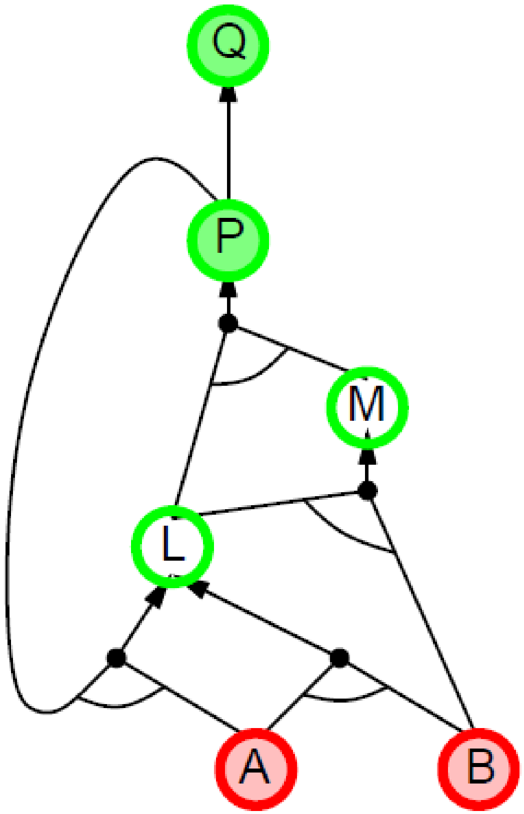
- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 2) Backward Chaining (후방 연쇄/역방향 추론)



목표 Q

명제 논리 (Propositional Logic)

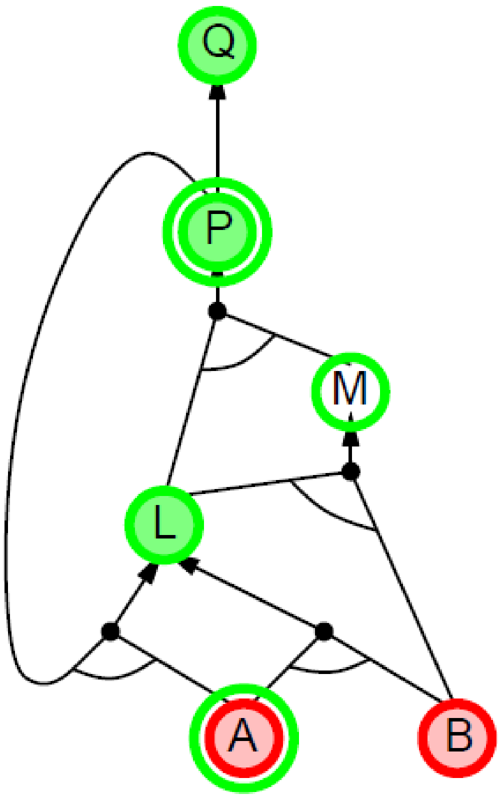
- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 2) Backward Chaining (후방 연쇄/역방향 추론)



목표 Q

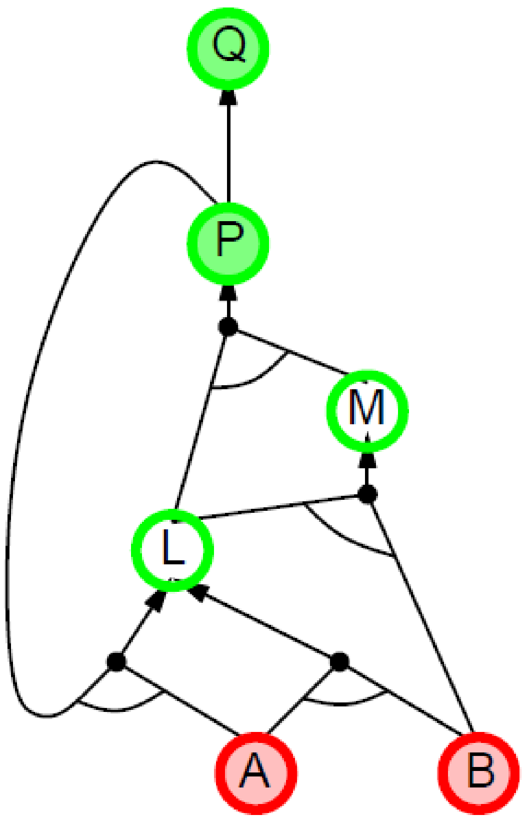
명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 2) Backward Chaining (후방 연쇄/역방향 추론)



목표 Q

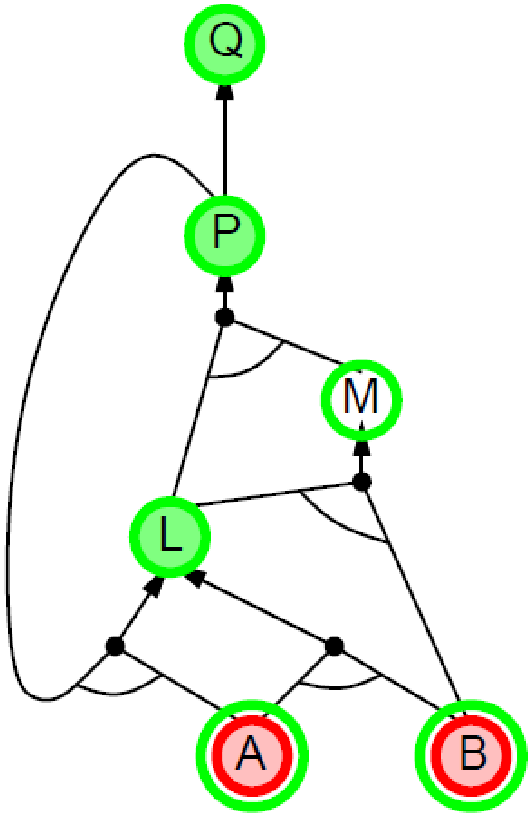
- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 2) Backward Chaining (후방 연쇄/역방향 추론)



목표 Q

명제 논리 (Propositional Logic)

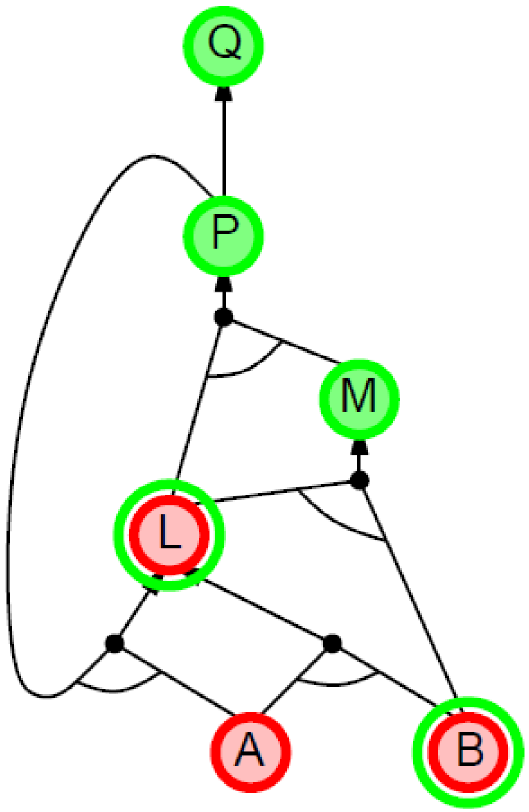
- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 2) Backward Chaining (후방 연쇄/역방향 추론)



목표 Q

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 2) Backward Chaining (후방 연쇄/역방향 추론)



목표 Q

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Proof Methods(증명 방법)
    - 연습문제) Forward Chaining vs. Backward Chaining (목표 Z)

KB(지식)

$A \Rightarrow X$
$X \wedge B \Rightarrow Y$
$Y \wedge C \wedge D \Rightarrow Z$
$Y \wedge C \wedge E \Rightarrow W$

사실(Fact)

A
B
C
D

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Forward Chaining vs. Backward Chaining
    - Forward Chaining은 Data-Driven 방법임 (즉 사실을 모은 후 이를 바탕으로 추론)
      - e.g.) object recognition, routine decision에 활용
    - Forward Chaining의 경우 목표(goal)과 관련 없는 규칙들에 대해서도 살펴봐야함 (BFS 방법과 유사)
    - Backward Chaining은 Goal-Driven 방법임 (즉 목표에서 시작하여, 데이터들이 목표를 지원하는지 확인)
      - e.g.) Problem-Solving에 활용 (where are my keys?, how do I get into a PhD program?)
    - Backward Chaining은 Forward Chaining 보다 Complexity가 낮음 (DFS 방법과 유사)
      - 즉 일반적으로 Backward Chaining은 Forward Chaining 보다 빠름(효과적)
      - 하지만 Backward Chaining의 경우 Complete(완전성), Optimal(최적성)을 보장 못함

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Resolution(분해법칙/도출원리)

> 추론규칙을 통해 증명(Proof)하기 위해서는 우선 주어진 지식이나 사실을 논리식으로 표현한 다음, 1) CNF(논리곱 표준형)으로 변환하고 2) Resolution(분해법칙) 및 논리적 동치 관계(Logically Equivalent)를 활용하여 논리식을 소거하면서 마지막으로 증명된 식이 참이거나 모순된 상황이 나오면 멈춤

- CNF(Conjunctive Normal Form, 논리곱 표준형)
  - 명제 P와 명제 부정의 ¬P를 리터럴(Literal)이라함 (P ∨ ¬P) = True (always)
  - 리터럴들이 논리합(∨)으로만 연결되거나, 논리곱(∧)으로 연결되면 절(Clause)라고 함
    - e.g.) P ∨ Q ∨ ¬ R (논리합 절)　　P ∧ Q ∧ ¬ R (논리곱 절)
  - 논리합 절들이 논리곱으로 연결되어 있으면 CNF라고 함

    → (P ∨ Q ∨ ¬R) ∧ (¬Q ∨ R ∨ S) ∧ (P ∨ R ∨ S)

Conjunctive normal form (CNF):
conjunction of clauses
$(\cdots \vee \cdots) \wedge (\cdots \vee \cdots) \wedge (\cdots \vee \cdots)$

KB is a conjunction of clauses (CNF)
$C \wedge (B \Rightarrow A) \wedge (C \wedge D \Rightarrow B)$

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - CNF 변환

$$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$$

- Eliminate $\Leftrightarrow$ replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
- $\left(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})\right) \wedge \left((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}\right)$
- Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$
- $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$
- Move $\neg$ inwards using de Morgan's rules and double-negation:
- $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge \left((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1}\right)$
- Apply distributivity law ($\vee$ over $\wedge$) and flatten:
- $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

- 명제 논리 (Propositional Logic)
  - CNF 변환
    - 연습문제) 아래의 식을 CNF로 변환

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

명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Resolution(분해법칙/도출원리)
    - 두 개의 논리합 절이 같은 기호의 긍정과 부정의 리터럴을 서로 포함하고 있을 때, 해당 리터럴들을 제외한 나머지 리터럴들의 논리합절을 만들어 내는 것

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

논리도출식(resolvent)

P	Q	R	¬P ∨ Q	P∨R	(¬P ∨ Q) ∧ (P ∨ R)	Q∨R
참	참	참	참	참	참	참
참	참	거짓	참	참	참	참
참	거짓	참	거짓	참	거짓	참
참	거짓	거짓	거짓	참	거짓	거짓
거짓	참	참	참	참	참	참
거짓	참	거짓	참	거짓	거짓	참
거짓	거짓	참	참	참	참	참
거짓	거짓	거짓	참	거짓	거짓	거짓

▪ 명제 논리 (Propositional Logic)

- Resolution(분해법칙/도출원리)

  - Resolution을 활용한 증명(Proof)

    - 다음 3가지 명제가 참이라고 주어져있다. 이때 R이 참이라는 것을 증명해 보자

$$P \vee Q \qquad P \rightarrow R \qquad Q \rightarrow R$$

① CNF 적용

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

② 모순(contradiction)을 유도 해내기 위해 증명할 명제 부정(R이 거짓)

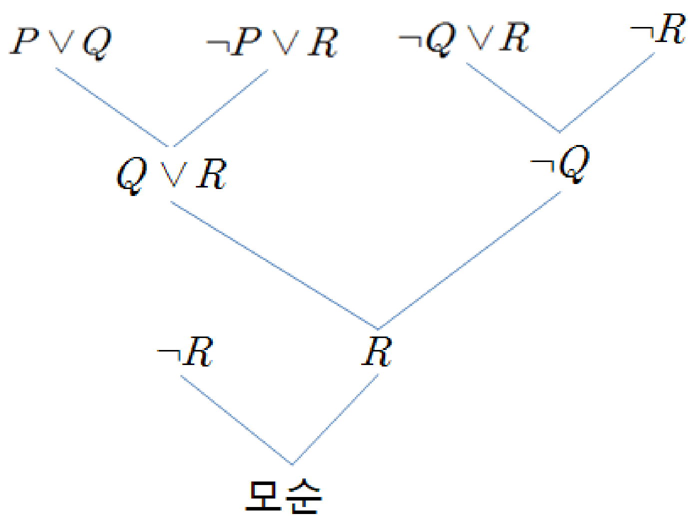
$$\neg R \qquad (P \vee Q) \wedge (\neg P \vee R) \wedge (\neg Q \vee R) \wedge \neg R$$

③ $P \vee Q$ 와 $\neg P \vee R$ 에 Resolution 적용

$$Q \vee R$$

④ $Q \vee R$ 와 $\neg Q \vee R$ 에 Resolution 적용

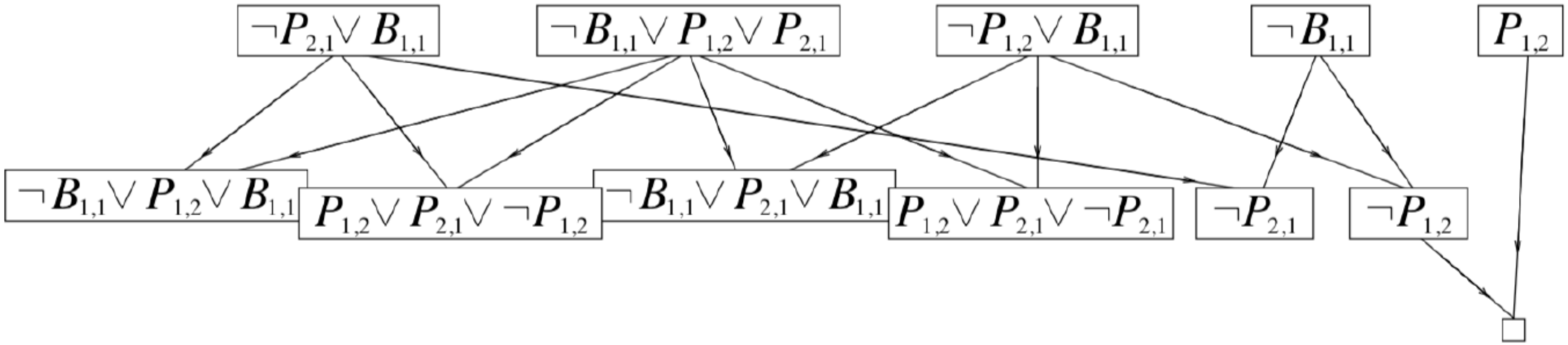
R 모순발생 (2번에서 명제를 부정)→ 즉 원래 명제 R은 참!!



- 명제 논리 (Propositional Logic)
  - Resolution(분해법칙/도출원리)
    - Resolution을 활용한 증명(Proof)
      - Wumpus World Cases (P$_{1,2}$에 Pit(구덩이)가 없음을 증명)

$$KB = \left(B_{1,1} \Leftrightarrow \left(P_{1,2} \vee P_{2,1}\right)\right) \wedge \neg B_{1,1} \qquad \neg\alpha = P_{1,2}$$

$$cf.\ B_{1,1} \Leftrightarrow \left(P_{1,2} \vee P_{2,1}\right) \rightarrow (\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$$



명제 논리 (Propositional Logic)

- 명제 논리 (Propositional Logic)
  - Pros and Cons (명제논리의 장단점)
    - Pros (장점)
      - 명제논리는 문맥이 독립적이고 모호하지 않은 선언적 의미론을 기반으로 함

        → 각 Syntax는 Fact(사실)에 해당
      - 명제논리는 (대부분의 자료구조 및 데이터베이스와는 달리) 부분(partial), 분리(disjunction), 부정(negation) 정보를 허용
      - 명제논리는 구성적(compositional)임 : 즉 $B_{1,1} \wedge P_{1,2}$의 의미는 $B_{1,1}$과 $P_{1,2}$의 의미에서 파생
    - Cons (단점)
      - 명제논리는 (자연어와는 달리) 표현력이 매우 제한적임 (limited expressive power)

        → "Pits(구덩이)와 인접한 칸에는 Breeze(미풍)이 분다"와 같은 문장을 표현하기 위해서는 각 칸마다 논리 형태로 표현해야 함 (e.g. $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$)

1차 논리 (First-order Logic)

- 1차 논리 (First-order Logic, FOL)란?
  - FOL, 술어논리(Predicate Logic) 이라고 함
    - 명제(proposition)의 내용을 다루기 위해 변수, 함수 등을 도입하고, 이들의 값에 따라 True, False가 결정되도록 명제논리(Propositional Logic)을 확장한 논리

      → 명제논리의 경우 Facts만 포함
    - FOL의 경우 아래의 요소들을 추가적으로 논리에 포함 가능
      - <u>Objects (객체)</u>: nouns(명사) and noun phrases(명사구)
      - e.g.) people, house, numbers, theories, Ronald McDonald, colors, baseball games, wars, centuries….
    - <u>Relations(Predicate, 관계)</u>: verbs(동사) and verb phrases(동사구)
      - e.g.) unary relations (properties(속성)): 1개의 객체가 가질 수 있는 관계: red, round, bogus, multi-storied,...
      - n-ary relations: 1개 이상의 객체가 가질 수 있는 관계: brother of, bigger than, inside, part of,….
    - <u>Functions(함수)</u>: 주어진 input에 대해 오직 한가지 value(값)을 가질 수 있는 Relations
      - e.g.) father of, best friend, third inning of, one more than, beginning of, …………..

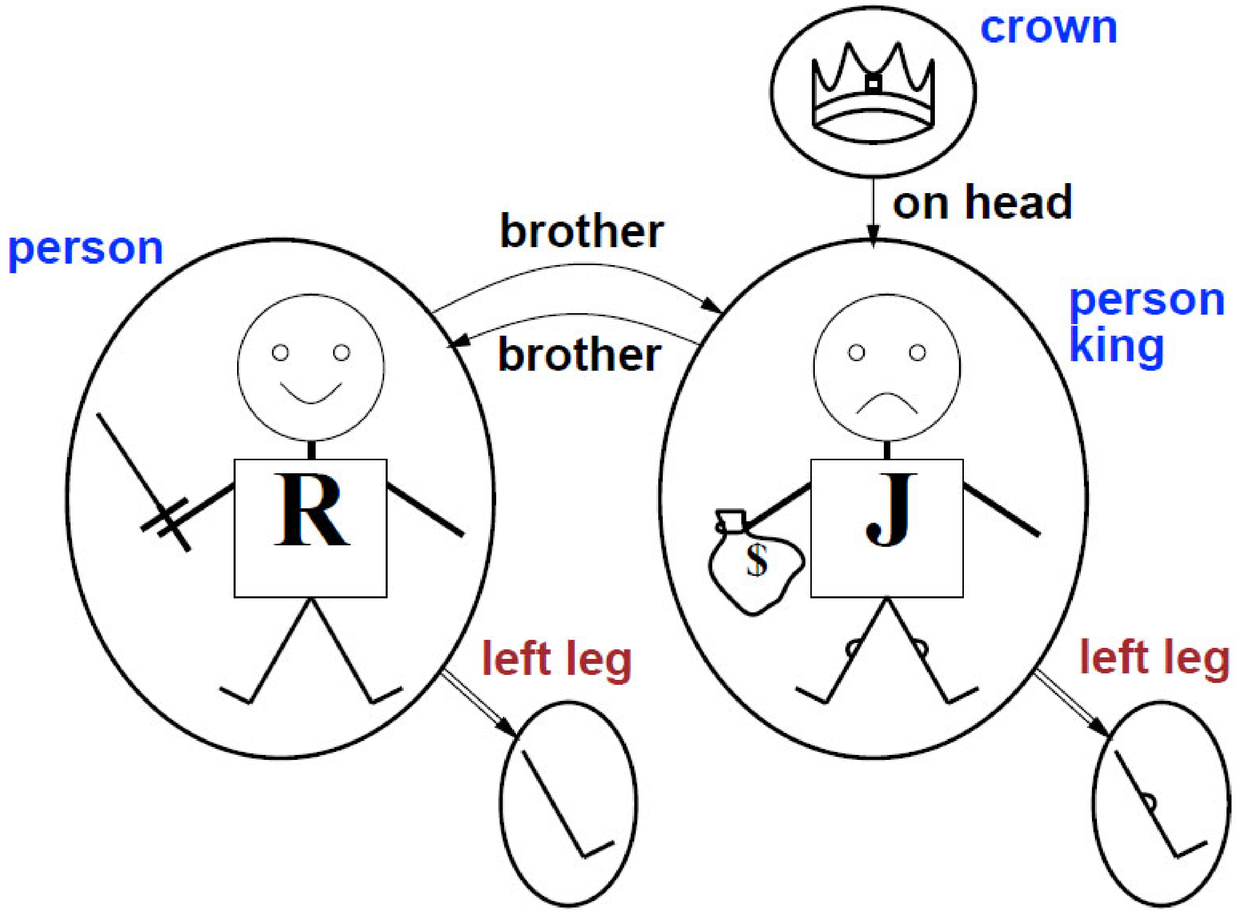
1차 논리 (First-order Logic)

- 1차 논리 (First-order Logic, FOL)란?
  - Logics in general (formal languages)

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
• Propositional logic	• facts	• true/false/unknown
• First-order logic	• facts, objects, relations	• true/false/unknown
• Temporal logic	• facts, objects, relations, times	• true/false/unknown
• Probability theory	• facts	• degree of belief
• Fuzzy logic	• facts + degree of truth	• known interval value

1차 논리 (First-order Logic)

- FOL의 Syntax와 Semantics
  - FOL 모델
    - Objects(객체)
      - *Richard, John*
    - Binary relations(관계)
      - *brother, on head*
    - Unary relations(관계)
      - *person, king, crown*
    - Functions (함수)
      - *LeftLeg*



1차 논리 (First-order Logic)

- FOL의 Syntax와 Semantics
  - FOL의 Syntax(구문)
    - Constants (상수) : *Richard, KingJohn, 2,*⋯.
    - Predicates (관계) : *Brother(x, y), →,* ⋯⋯
    - Functions (함수) : *Sqrt(x), LeftLegOf(x),* ⋯.
    - Variables (변수) : $x, y, a, b \cdots$
    - Connectives (논리접속사) : $\wedge,\ \vee,\ \neg,\ \Rightarrow,\ \Leftrightarrow,\ \cdots$
    - Equality (상등) : $=$
    - Quantifiers (한정사) : $\forall, \exists$

- Let "$x + y = z$" be denoted by $R(x, y, z)$ and U (for all three variables) be the integers. Find these truth values:

  R(2,-1,5)

  Solution: F

  R(3,4,7)

  Solution: T

  R(x, 3, z)

  Solution: Not a Proposition

- FOL의 Syntax와 Semantics
  - FOL의 Syntax(구문)
    - Terms (항)
      - 하나의 Object(객체)를 지칭하는 논리식 (constants(상수)의 경우 Term(항)임)
      - 항이 될 수 있는 것: 상수, 변수, 함수

        e.g.) Richard, KingJohn, 2, x, LeftLeg(Richard)

        (1) 개체상수, 변수는 항이다.

        (2) $t_1, t_2, \cdots, t_n$이 모두 항이고, f가 n개의 인자를 갖는 함수 기호일 때,

        $f(t_1, t_2, \cdots, t_n)$은 항이다.

        (3) (1)과 (2)에 의해 만들어질 수 있는 것만 항이다.
    - Atomic Sentences (원자적 문장)
      - Sentence is a predicate which can be evaluated as true or false (객체를 지칭하는 term과 관계를 지칭하는 predicate이 갖추어 졌다면, 조합해서 원자적 문장(atomic sentence)를 만들 수 있음)

        e.g.) Brother(Richard, John), Married(Father(Richard), Mother(John))

1차 논리 (First-order Logic)

- FOL의 Syntax와 Semantics
  - FOL의 Syntax(구문)
    - Complex Sentences (복합 문장)
      - 논리접속사(logical connectives)를 이용해서 복합 문장 생성

e.g.) $Brother(Richard, John) \land Brother(John, Richard)$

$\neg Brother(LeftLeg(Richard), John)$

$King(Richard) \lor King(John)$

$\neg King(Richard) \Rightarrow King(John)$

- FOL의 Syntax와 Semantics
  - FOL의 Syntax(구문)
    - Quantifiers (한정사)

– 전칭 한정사(universal quantifier) $\forall x\ P(x)$ is read as "For all x, P(x)" or "For every x, P(x)"

- For all (모든 ….에 대해)

e.g.) $\forall x\ King(x) \Rightarrow Person(x)$: 모든 x에 대해, 만일 x가 왕이면 x는 사람이다

$\forall x\ At(x, GNU) \Rightarrow Smart(x)$: 모든 x에 대해, 만일 x가 GNU에 다니면 x는 Smart하다

- Syntax(구문)

∀ <variables> <sentences>

- Semantics(의미)

P가 임의의 논리식일 때, '∀x P(x)라는 문장은 모든(all) 객체 x에 대해 P가 참이다'라는 것을 의미

1차 논리 (First-order Logic)

- FOL의 Syntax와 Semantics
  - FOL의 Syntax(구문)
    - Quantifiers (한정사)
      - 전칭 한정사(universal quantifier)의 주요실수 (common mistake)

- 일반적으로, ⇒ (implication)은 ∀ 기호와 주로 연결되어 있음

- Common mistake: <u>∀ 기호와 ∧ 기호를 연결해서 사용</u>

$\forall x\ At(x, GNU) \wedge Smart(x)$:

모든 x에 대해, x는 GNU에 다닌다 ∧ (그리고) 모든 x는 Smart하다

"Everyone is at GNU and everyone is smart"

x= {*Richard, John, LeftLeg(Richard), Crown,* …..}

→ x는 다음 중 하나가 될 수 있음

너무 강한 주장이 됨!!

1차 논리 (First-order Logic)

- FOL의 Syntax와 Semantics
  - FOL의 Syntax(구문)
    - Quantifiers (한정사)

– 존재 한정사(existential quantifier)

$\exists x\, P(x)$ is read as “For some x, P(x)”, or as “There is an x such that P(x),” or “For at least one x, P(x).”

- For some…, or There exists (어떤 ….에 대해)

e.g.) $\exists\, x\ Crown(x) \wedge OnHead(x, John)$: 어떤 x에 대해, x는 왕관이며 x는 John 머리위에 있다

$\exists\, x\ At(x, GNU) \Rightarrow Smart(x)$: 어떤 x에 대해, x가 GNU에 다니면 x는 Smart하다

- Syntax(구문)

∃ <variables> <sentences>

- Semantics(의미)

P가 임의의 논리식일 때, ‘∃ x P(x)라는 문장은 일부(some) 객체 x에 대해 P가 참이다’라는 것을 의미

1차 논리 (First-order Logic)

- FOL의 Syntax와 Semantics
  - FOL의 Syntax(구문)
    - Quantifiers (한정사)
      - 존재 한정사(existance quantifier)의 주요실수 (common mistake)

        - 일반적으로 $\wedge$은, $\exists$ 기호와 주로 연결되어 있음

        - Common mistake: <u>$\Rightarrow$ (implication) 기호와 $\wedge$ 기호를 연결해서 사용</u>

$\exists\, x\, At(x, GNU) \Rightarrow Smart(x)$:

어떤 x에 대해, 만약 x는 GNU에 다닌다면 x는 Smart하다

→ 위의 의미는 만약 GNU에 다니지 않는 x가 있으면,
x는 smart하다 라는 의미가 됨

너무 약한 주장이 됨!!

1차 논리 (First-order Logic)

- FOL의 Syntax와 Semantics
  - FOL의 Syntax(구문)
    - Quantifiers (한정사)
      - 아래의 식은 어떻게 해석해야할까?

1) $\exists x \ \mathrm{Friend}(\mathrm{John},\ x)$

2) $\forall x \exists y \ \mathrm{Friend}(x, y)$

    - Equality (상등)
      - 2개의 terms(항)이 같다는 것을 보임 (원자적 문장)

$$Father(John) \Rightarrow Henry$$

1차 논리 (First-order Logic)

- FOL의 Syntax와 Semantics
  - BNF(Bacus-Naur Form)

$$
\begin{array}{rcl}
\mathit{Sentence} & \rightarrow & \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} & \rightarrow & \mathit{Predicate} \mid \mathit{Predicate}(\mathit{Term},\ldots) \mid \mathit{Term} = \mathit{Term} \\
\mathit{ComplexSentence} & \rightarrow & (\ \mathit{Sentence}\ ) \mid [\ \mathit{Sentence}\ ] \\
& \mid & \neg\ \mathit{Sentence} \\
& \mid & \mathit{Sentence} \land \mathit{Sentence} \\
& \mid & \mathit{Sentence} \lor \mathit{Sentence} \\
& \mid & \mathit{Sentence} \ \Rightarrow\ \mathit{Sentence} \\
& \mid & \mathit{Sentence} \ \Leftrightarrow\ \mathit{Sentence} \\
& \mid & \mathit{Quantifier}\ \mathit{Variable},\ldots\ \mathit{Sentence} \\
\\
\mathit{Term} & \rightarrow & \mathit{Function}(\mathit{Term},\ldots) \\
& \mid & \mathit{Constant} \\
& \mid & \mathit{Variable} \\
\\
\mathit{Quantifier} & \rightarrow & \forall \mid \exists \\
\mathit{Constant} & \rightarrow & A \mid X_1 \mid \mathit{John} \mid \cdots \\
\mathit{Variable} & \rightarrow & a \mid x \mid s \mid \cdots \\
\mathit{Predicate} & \rightarrow & \mathit{True} \mid \mathit{False} \mid \mathit{After} \mid \mathit{Loves} \mid \mathit{Raining} \mid \cdots \\
\mathit{Function} & \rightarrow & \mathit{Mother} \mid \mathit{LeftLeg} \mid \cdots \\
\textsc{Operator Precedence} & : & \neg, =, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

1차 논리 (First-order Logic)

- FOL의 Syntax와 Semantics
  - BNF(Bacus-Naur Form)

(1) $t_1, t_2, \cdots, t_n$이 모두 항이고, p가 n개의 인자를 갖는 술어 기호일 때, $p(t_1, t_2, \cdots, t_n)$은 정형식이다.

(2) p와 q가 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, x가 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.

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

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

$Horse(Harry)$

$Rabbit(Ralph)$

$\forall y\ Greyhound(y) \rightarrow Dog(y)$

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

1차 논리 (First-order Logic)

- FOL의 Syntax와 Semantics
  - Quantifier(한정사)의 속성
    - $\forall x \ \forall y$ is the same as $\forall y \ \forall x$
    - $\exists x \ \exists y$ is the same as $\exists y \ \exists x$
    - $\exists x \ \forall y$ is **not** the same as $\forall y \ \exists x$
      - $\exists x \ \forall y \ \ Loves(x, y)$
        "There is a person who loves everyone in the world"
      - $\forall y \ \exists x \ \ Loves(x, y)$
        "Everyone in the world is loved by at least one person"
    - Quantifier duality: each can be expressed using the other

$$\forall x \ \ Likes(x, IceCream) \qquad \neg\exists x \ \ \neg Likes(x, IceCream)$$

$$\exists x \ \ Likes(x, Broccoli) \qquad \neg\forall x \ \ \neg Likes(x, Broccoli)$$

1차 논리 (First-order Logic)

- FOL의 Syntax와 Semantics
  - Quantifier(한정사)를 이용한 문장 표현

Example 1: Translate the following sentence into predicate logic: "Every student in this class has taken a course in Java."

Solution:

First decide on the domain U.

Solution 1: If U is all students in this class, define a propositional function J(x) denoting "x has taken a course in Java" and translate as $\forall x\, J(x)$.

Solution 2: But if U is all people, also define a propositional function S(x) denoting "x is a student in this class" and translate as $\forall x\, (S(x) \rightarrow J(x))$.

$\forall x\, (S(x) \wedge J(x))$ is not correct. What does it mean?

- FOL의 Syntax와 Semantics
  - Quantifier(한정사)를 이용한 문장 표현

Example 2: Translate the following sentence into predicate logic: "Some student in this class has taken a course in Java."

Solution:

First decide on the domain *U*.

Solution 1: If *U* is all students in this class, translate as

$$\exists x\, J(x)$$

Solution 1: But if *U* is all people, then translate as

$\exists x\, (S(x) \wedge J(x))$

$\exists x\, (S(x) \rightarrow J(x))$ is not correct. What does it mean?

1차 논리 (First-order Logic)

- FOL의 Syntax와 Semantics
  - **Quantifier(한정사)를 이용한 문장 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)

경상국립대학교
Gyeongsang National University

Improving lives through learning

IDEALAB