

# 논리 에이전트

## 개요

- I. 지식 기반 에이전트(Knowledge-based agents)
- II. 웬푸스 환경(Wumpus world)
- III. 논리(Logic)

# I. 지식 기반 에이전트

- ♣ 문제 해결 에이전트는 일반 상식이 부족함

8 조각 맞추기 퍼즐 에이전트는 두개의 타일이 하나의 공간을 차지 할 수 없다는 당연한 사실을 모르고 있음(코드 내에서 명시해 주어야 함)

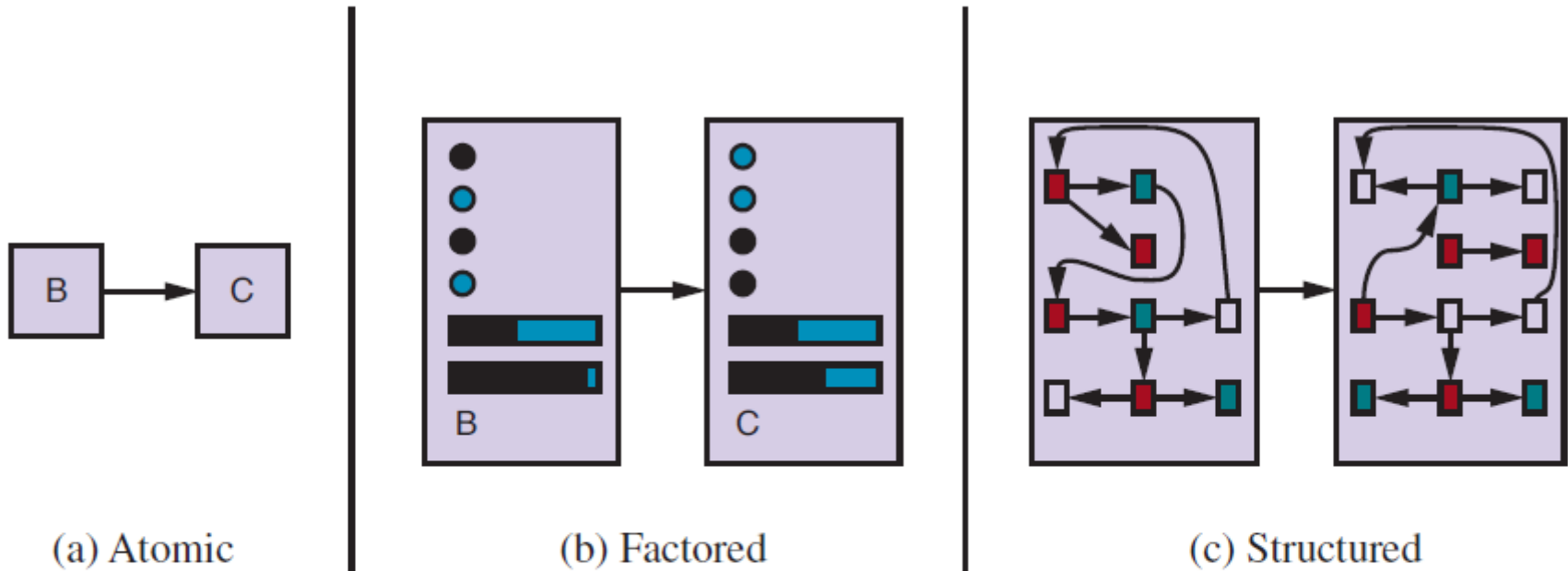
- ♣ 문제 해결 에이전트의 원자형 표현이 갖는 정보는 제한적임(당면한 문제에만 국한된 정보)

예) 모든 가능한 구체적 상태의 목록

- ◆ 지능형 에이전트는 올바른 의사결정을 위해 추론을 수행할 수 있도록 *상식(세상에 대한 지식)*이 필요하다.

- 상태, 행동 등에 대한 표현 방법
- 새로 감지한 정보를 통합하는 방법
- 외부 환경에 대한 내적 표현의 갱신
- 외부 환경 중 숨겨진 부분 추론하는 방법
- 적절한 행동을 추론하는 방법

# Agent Types Classified by Complexity and Expressive Power

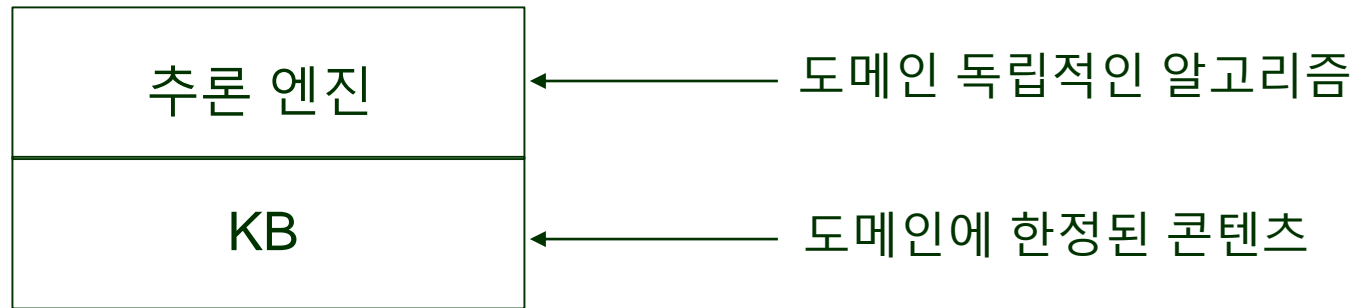


**Figure 2.16** Three ways to represent states and the transitions between them. (a) Atomic representation: a state (such as B or C) is a black box with no internal structure; (b) Factored representation: a state consists of a vector of attribute values; values can be Boolean, real-valued, or one of a fixed set of symbols. (c) Structured representation: a state includes objects, each of which may have attributes of its own as well as relationships to other objects.

# 지식 기반

**지식 기반(knowledge base, KB):** 외부 환경에 대한 주장을 표현하는 문장(sentence) 집합

**공리(axiom):** 다른 문장으로부터 유도되지 않고 그 자체로 참이라고 받아들여지는 문장(sentence)



**TELL:** KB에 새로운 문장을 추가

**ASK:** KB에 질의를 수행.

**추론:** 이전 지식을 이용하여 신규 문장 도출

# 범용 지식 기반 에이전트

**function** KB-AGENT(*percept*) **returns** an *action*

**persistent:** *KB*, a knowledge base

*t*, a counter, initially 0, indicating time

TELL(*KB*, MAKE-PERCEPT-SENTENCE(*percept*, *t*))

*action*  $\leftarrow$  ASK(*KB*, MAKE-ACTION-QUERY(*t*))

TELL(*KB*, MAKE-ACTION-SENTENCE(*action*, *t*))

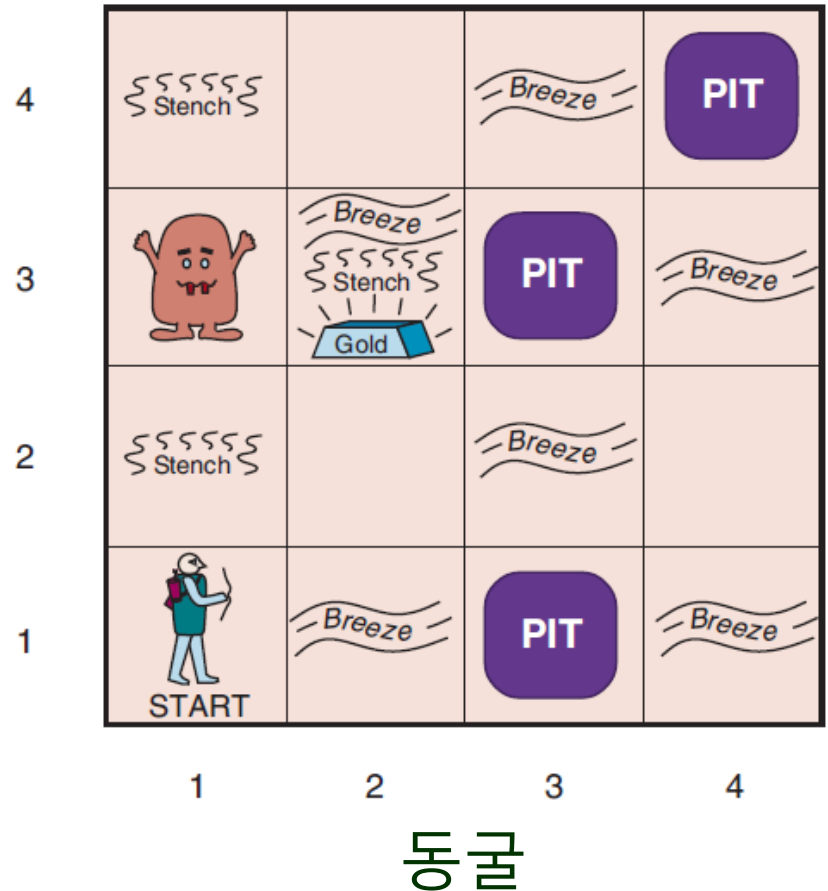
*t*  $\leftarrow$  *t* + 1

**return** *action*

## II. 워프스 환경

### 방으로 연결된 동굴 환경

- 들어 가면 잡아 가두는 구렁이 함정이 있는 방
- 방에 숨어서 그 방에 들어오는 사람을 잡아먹으려 하는 워프스
- 에이전트는 워프스를 쏴아 죽일 수 있는 단 하나의 화살을 가짐
- 금덩어리는 워프스가 숨어 있는 곳과 다른 방에 있습니다.



**목표:** 죽지 않고 금을 찾아 출발 지점으로 가져오기

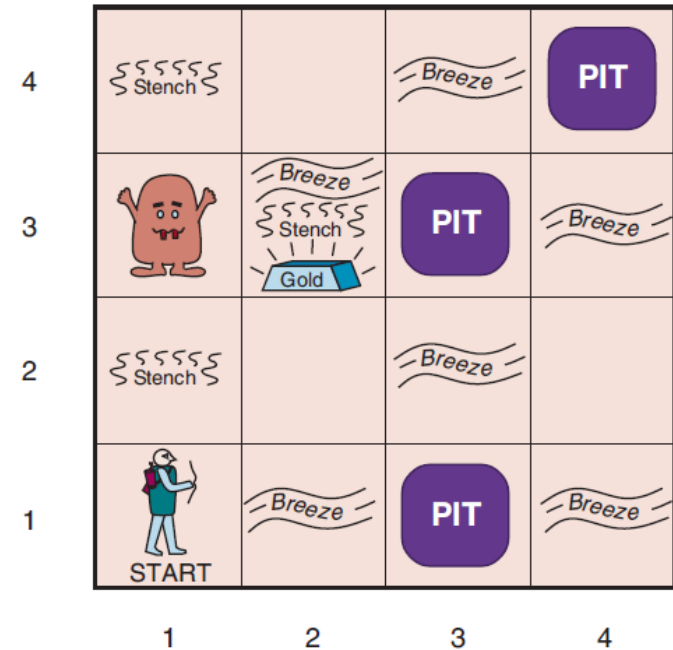
# 작업 환경

## 성능 측정 기준 (Performance Measurement)

- +1000 (금을 가지고 동굴을 빠져 나온 경우)
- -1000 (구덩이에 빠지거나 움푸스에게 잡아 먹힘)
- -1 (액션 한 번 취할 때 마다)
- -10 (화살 한 번 쏜 경우)

## 환경(Environment)

- 4 × 4 행렬 격자. 각 방은 벽으로 구분
- [1,1]: 에이전트 출발 지점. 동쪽으로 향함
- 금덩어리와 움푸스의 위치:
  - $\neq [1, 1]$
  - 균일 분포를 따라 무작위로 위치 생성
- 금괴나 움푸스가 없는 방 중에서 0.2의 확률로 함정이 있는 방 결정



# 액추에이터

## 액추에이터:

### 1) 전진, 왼쪽으로 90도 회전, 오른쪽으로 90도 회전

- 구덩이나 살아있는 뱀프스가 있는 칸에 들어가면 에이전트 사망
- 벽에 부딪히면 이동 불가(제자리)

### 2) 잡기(획득)

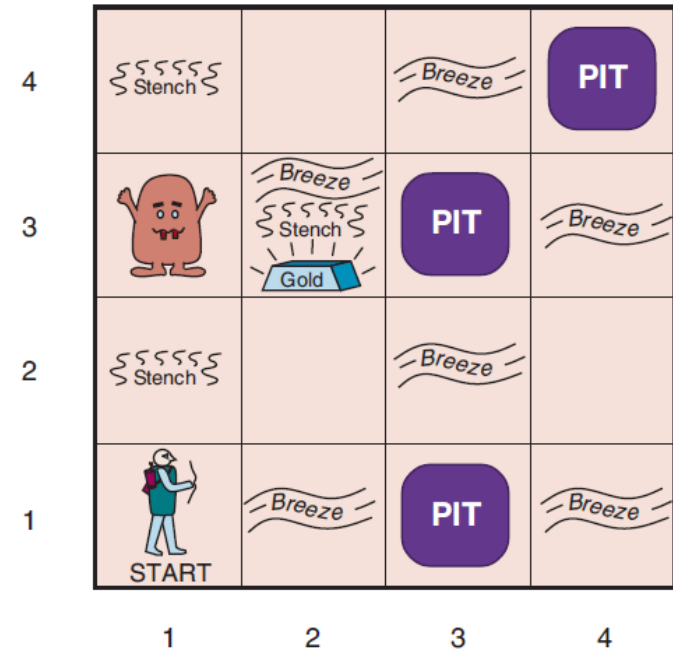
- 금이 있는 칸에 도착했을 때 금 획득

### 3) 쏘기

- 에이전트가 향하고 있는 방향으로 화살 발사
- 화살은 뱀프스를 맞혀서 결과적으로 그를 죽이거나, 또는 벽에 부딪힐 때까지 계속 날아감

### 4) 기어 오르기

- [1, 1] 위치로 돌아 오면 동굴에서 기어 올라 올





# 센서

5 가지 센서, 각 센서 별로 1 비트 정보 제공:

1) 악취

- 뽕푼스가 있는 방과 바로 인접한 방에서 악취가 남

2) 바람

- 구덩이가 있는 방과 인접한 방에는 바람이 불어옴

3) 반짝임

- 금덩이가 있는 방에는 반짝임이 감지됨

5차원 벡터 형식으로 인지:

e.g., [악취, 바람, None, None, None]

4) 충돌

- 에이전트가 벽 쪽으로 계속 걸으려고 하면 충돌이 감지됨

5) 비명

- 뽕푼스가 화살에 맞아 죽을 때 비명이 감지됨

# 웜프스 세계의 특성

- ◆ 결정론적, 이산적, 정적, 단일 에이전트

확률적인 요소가 배제됨

웜프스가 움직이지 않음

- ◆ 부분적으로 관측 가능

구덩이와 웜프스의 위치가 알려져 있지 않음

**도전 과제:** 에이전트는 처음에는 환경의 구성을 알지 못함 => 논리적 추론을 사용하여 이를 해결

# 지식 기반 에이전트에 의한 해법

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2	3,2	4,2
1,1 A OK	2,1 OK	3,1	4,1

**A** = Agent  
**B** = Breeze  
**G** = Glitter, Gold  
**OK** = Safe square  
**P** = Pit  
**S** = Stench  
**V** = Visited  
**W** = Wumpus

Forward



[Stench, Breeze, Glitter, Bump, Scream]

인지: [None, None, None, None, None]



[1,2] 와 [2, 1] 은 위험하지 않는 방

[None, Breeze, None, None, None]



[1,1] , [2,2] , 또는 [3, 1]에 구덩이



[1,1] 은 이전에 출발한 곳

[2,2] 또는 [3, 1]에 구덩이

# 다음 단계 진행

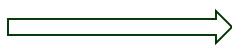
1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2 P?	3,2	4,2
1,1 V OK	2,1 A B OK	3,1 P?	4,1

[Stench, Breeze, Glitter, Bump, Scream]

방문하지 않은 방 중 [1,2] 번 방만 OK.

안전한 행보: 뒤 돌아서, [1,1]방으로  
돌아간 뒤 [1,2]번 방으로 이동

**A** = Agent  
**B** = Breeze  
**G** = Glitter, Gold  
**OK** = Safe square  
**P** = Pit  
**S** = Stench  
**V** = Visited  
**W** = Wumpus



1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

[Stench, None, None, None, None]

웬푸스 가능한 방 [1,1] , [2,2] , [1, 3]

[1,1] is OK  
 [2,2] is impossible because no  
 stench was detected at [2, 1].

그렇다면 웬푸스는 [1, 3]에 있음

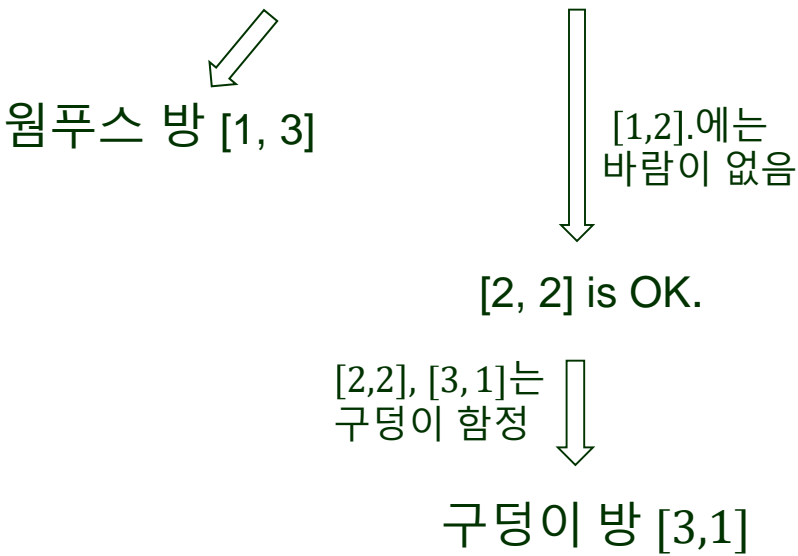
- A** = Agent
- B** = Breeze
- G** = Glitter, Gold
- OK** = Safe square
- P** = Pit
- S** = Stench
- V** = Visited
- W** = Wumpus

# 다음 단계 진행

1,4	2,4 P?	3,4	4,4
1,3 W!	2,3 <b>A</b> S G B ↑	3,3 P?	4,3
1,2 S V OK →	2,2 V OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

수집한 정보가 정확하다면(명제들이 참이라면) 도출된 결론도 참이 보장됨

[Stench, None, None, None, None]



- (2, 2)로 이동
- (2,2)에서 수집한 정보를 근거로 왼쪽으로 돌아서 (2.3)으로 이동
- 금괴를 집어서 출발지점으로 복귀

# III. 논리

- 추론 규칙에 대한 체계적인 연구
  - 결론을 도출할 수 있도록 정보를 표현하는 공식 언어.
- ◆ **구문** - 어떤 표현이 합법적인지 (올바르게 구성된 문장인지)

“ $x + y = 4$ ” 는 올바른 문장, “ $x4y +=$ ” 는 아님

- ◆ **의미론** - 문장의 "의미"가 무엇인지.

각각의 가능한 세계(**모델**)에서의 문장의 진실성.

“ $x+y=4$ ”는  $x$ 가 2이고  $y$ 가 2인 세계에서는 참이지만,  
 $x$ 가 1이고  $y$ 가 1인 세계에서는 거짓

모든 문장은 각 가능한 세계에서 참 또는 거짓, 둘 중 하나

# 모델과 추론

모델  $m$ : 변수에 값을 할당함

모델  $m$ 에서  $\alpha$ 가 참인 경우,  $m$ 은 문장  $\alpha$ 를 만족. 또는  $m$ 은  $\alpha$ 의 모델

$M(\alpha)$ :  $\alpha$ 의 모든 모델의 집합.

논리적 함의

$\alpha \models \beta$   $\alpha$ 의 모든 모델이  $\beta$ 의 모델일 때에만

“문장  $\alpha$ 는 문장  $\beta$ 를 함의한다.”




이와 동치로,

$\alpha \models \beta$ 은  $M(\alpha) \subseteq M(\beta)$ 일 때에만 성립

$\Downarrow$   
 $\alpha$ 는  $\beta$ 보다 더 강한 주장임

예  $x = 0$  는  $xy = 0$ 를 함의

# 웜푸스 환경 재고찰

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2 P? 	3,2	4,2
1,1 V OK 	2,1  B OK	3,1 P? 	4,1

[Stench, Breeze, Glitter, Bump, Scream]

지식 기반(Knowledge base, KB):

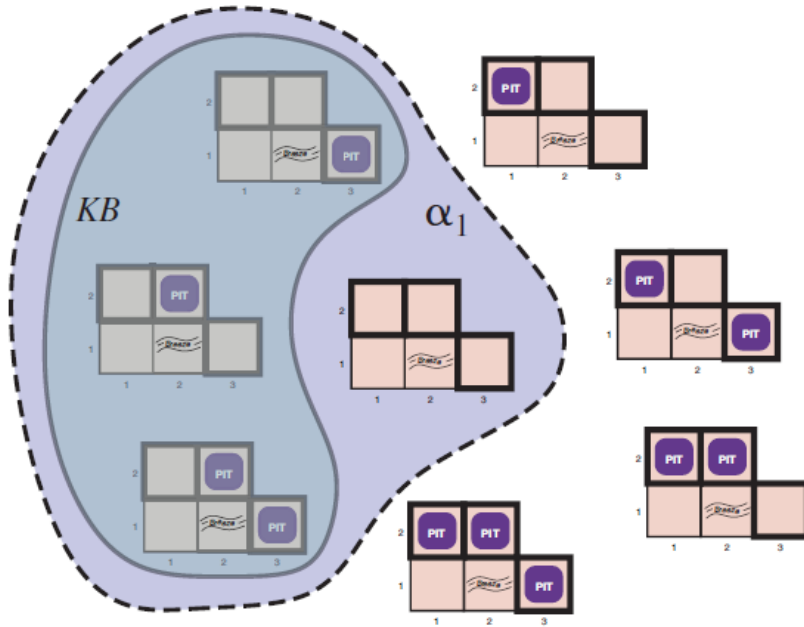
- 모든 규칙들
- [2,1] 방문 시점까지 획득한 인지 정보:  
*[None, None, None, None, None]* in [1,1]  
*[None, Breeze, None, None, None]* in [2,1]

**Q.** 다음 세 방 중에 구덩이 함정이 있는 곳은?  
[1,2], [2,2], [3, 1]

KB 를 이용하지 않을 경우 8가지 가능성에 대해 조사 필요



# 이웃하는 방 중에서 함정이 있는 곳은?



[1, 2], [2, 2], [3, 1] 위치에 구덩이가 있을 수 있는 모델은 8개  
 2가지 인지정보가 주어진 경우 KB가 참인 3가지 모델 존재 [None, None, None, None, None] in [1,1]  
[None, Breeze, None, None, None] in [2,1]

$\alpha_1$  = "[1, 2]에는 함정이 없다."

4개 모델에서 참

모델 체크

$KB \models \alpha_1$  since  $M(KB) \subseteq M(\alpha_1)$

$\alpha_2$  = "[2, 2]에는 함정이 없다."

4개 모델에서 참

$KB \not\models \alpha_2$  since  $M(KB) \not\subseteq M(\alpha_2)$

# 도출(Derivation)과 함의(Entailment)

추론은 건초 더미(KB) 안에 있는 바늘(함의되는 내용)을 찾는 것과 유사함

$KB \vdash_i \alpha$  " $\alpha$ 는 (추론 알고리즘)  $i$ 에 의해  $KB$ 에서 **도출**됨"

" $i$ 는  $KB$ 로부터  $\alpha$ 를 **도출**함"

♣ "무오류(**sound**)" 또는 "진실 보존(**truth-preserving**)"

추론 알고리즘  $i$ 가 오직 함의된 문장들만 도출하는 경우

즉,  $KB \vdash_i \alpha$  일 경우 항상  $KB \models \alpha$

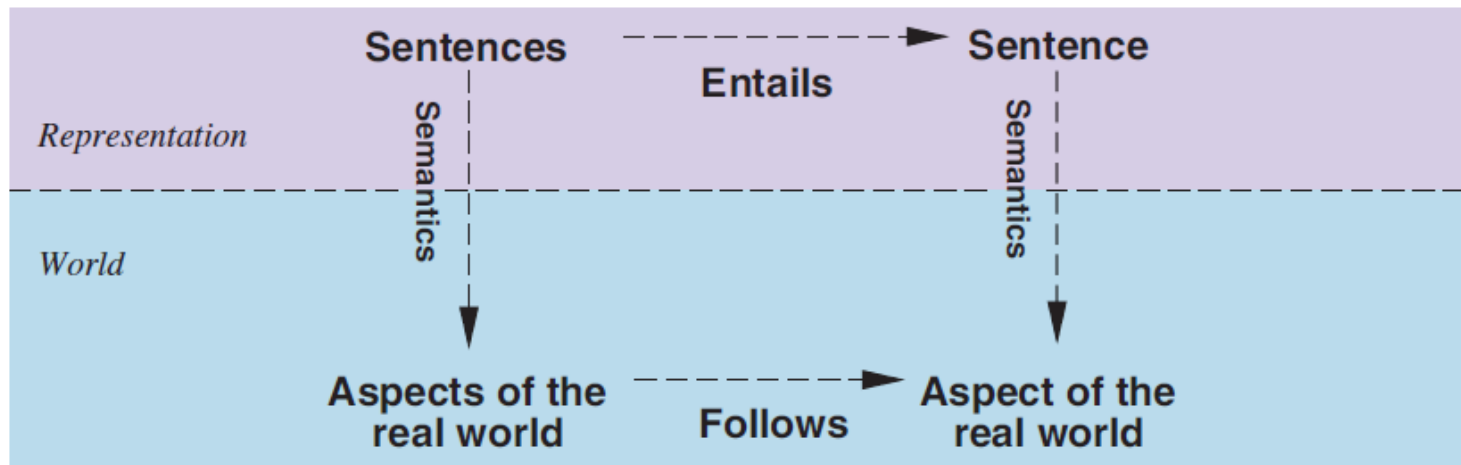
♣ "완결성(**complete**)"

함의된 모든 문장을 도출할 수 있다는 의미

즉,  $KB \models \alpha$  일 경우 항상  $KB \vdash_i \alpha$

# 논리적 추론

전제, 즉 KB가 참인 것들로 구성된 임의의 세계(환경)에서 어떤 주장이 참이라고 결론 지어지는 과정



세계와 표현 간의 대응 관계