Introduction to Artificial Intelligence

Chapter 3: Kwoledge Representation and Reasoning (1) Logic Agents

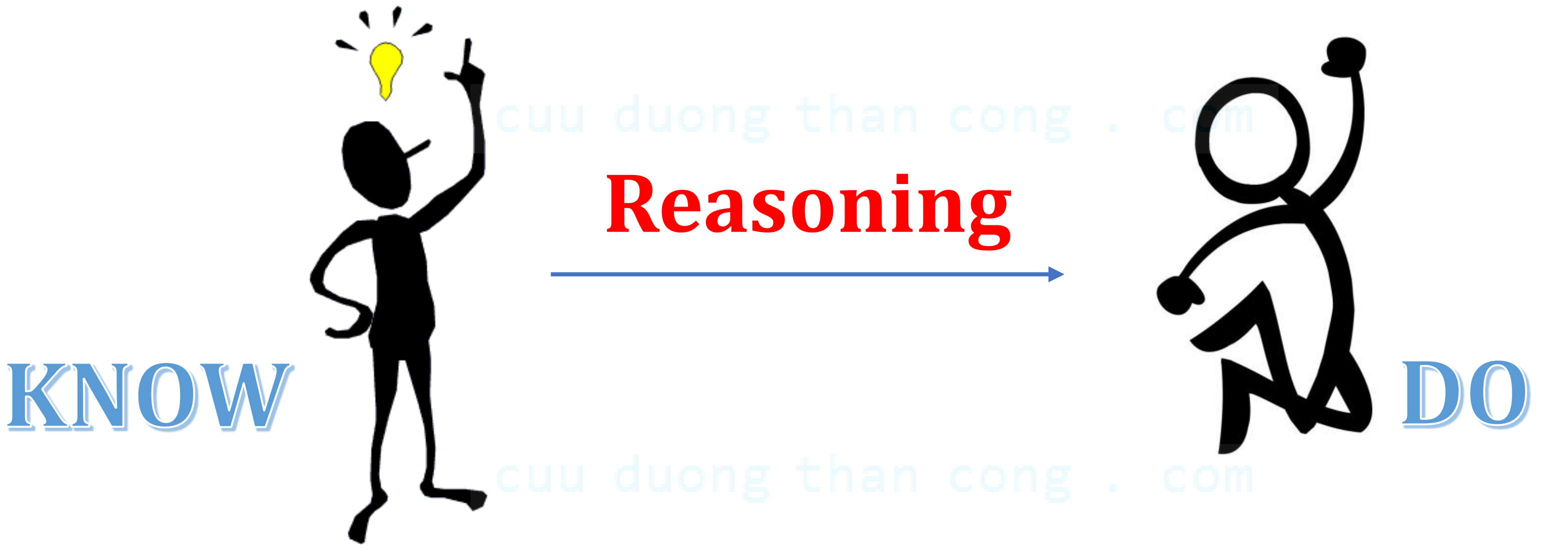
Nguyễn Hải Minh, Ph.D
nhminh@fit.hcmus.edu.vn

Outline

1. Knowledge-Based Agents
2. The Wumpus World

Knowledge-Based Agents

- Human intelligence is achieved not purely by reflex mechanisms but by reasoning that operate on internal representation of knowledge



In AI, this approach to intelligence is embodied in **knowledge-based agents**.

Knowledge-Based Agents

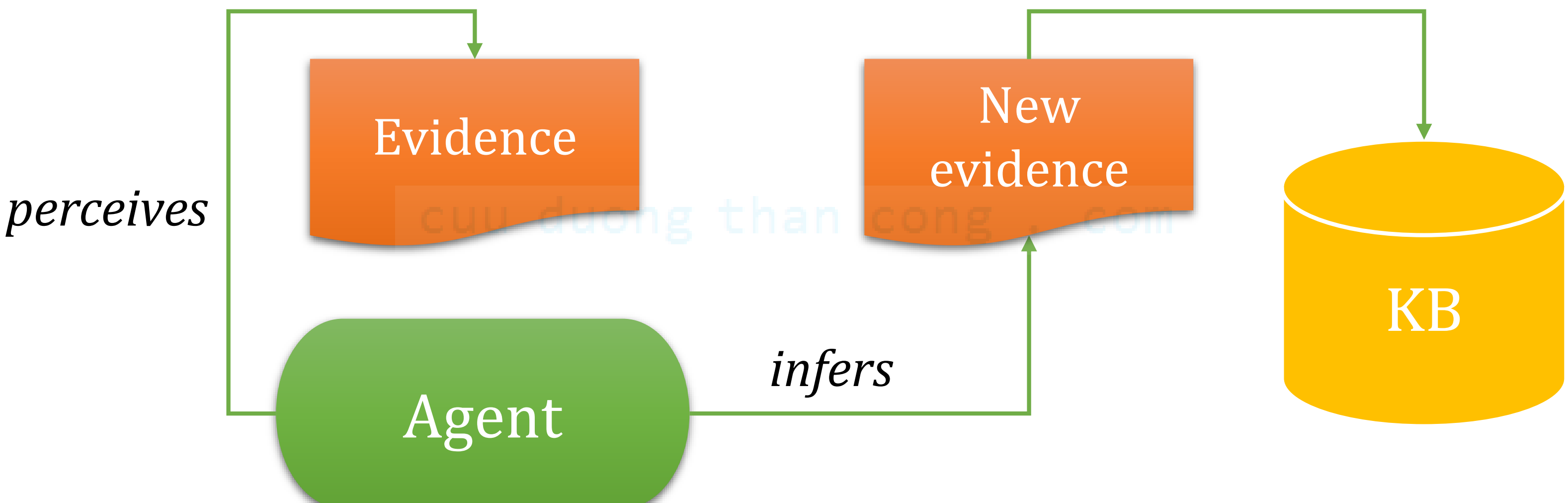
- Problem-solving agents in chapter 2:
  - State-space model:
    - Limited knowledge
    - Inflexible
    - Cannot make deduction
  - Constraint satisfaction problem solver:
    - Enabling some parts of the agent to work in a domain-independent way
    - More efficient algorithms
    - Can be developed to logical agents

Knowledge-Based Agents

- Knowledge-based agents:
  - Can combine and recombine information
  - Can learn new knowledge about the environment
  - Can adapt the changes in the environment by updating the relevant knowledge

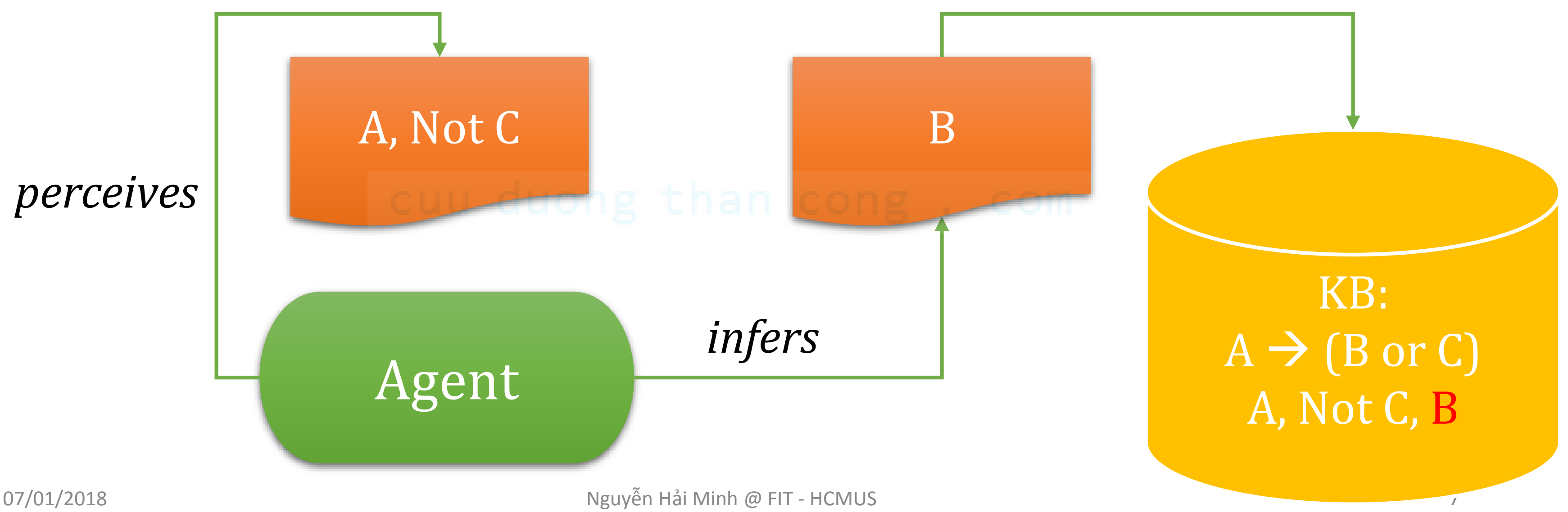
Knowledge-Based Agents

- ❑KB = Knowledge base
  - A set of sentences or facts (in a logic language)
- ❑Inference
  - Derive (infer) new sentences from old ones
- ❑A simple model for reasoning:



Knowledge-Based Agents

- A simple model for reasoning:
  - KB = {A → (B or C)}, then given A and Not C, we can infer that B is true
  - B is now added to the KB even though it is not explicitly asserted



A generic knowledge-based agent

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

A generic knowledge-based agent

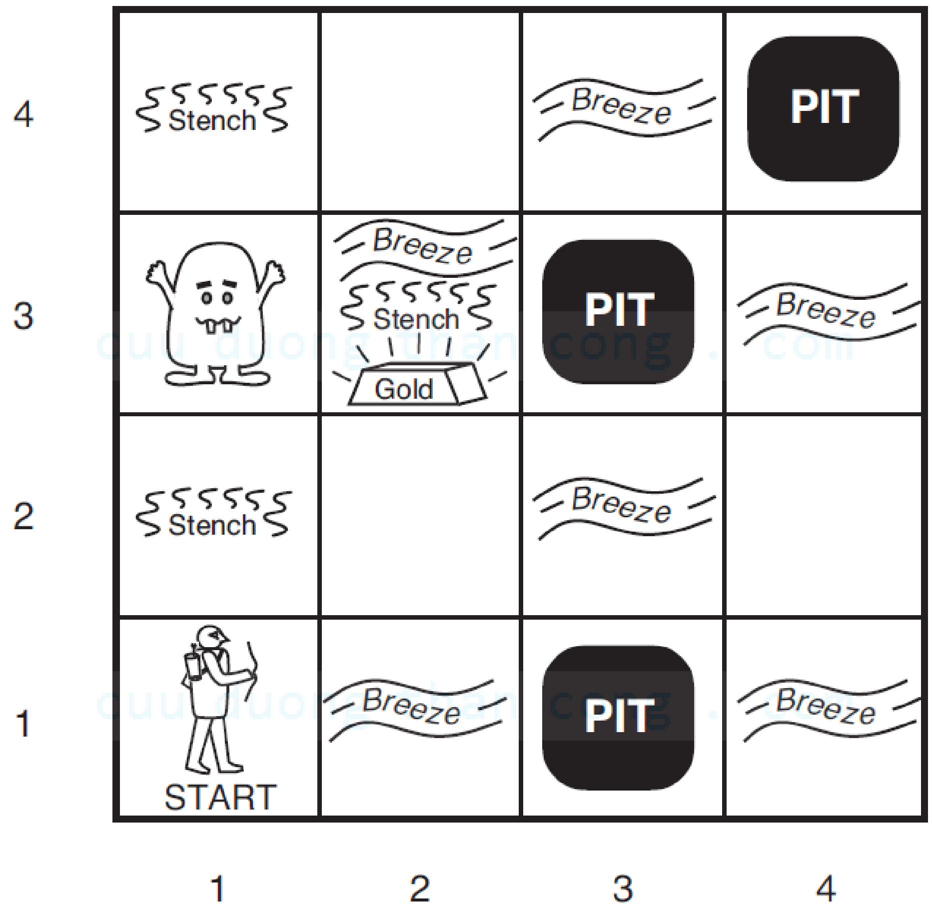
- ❑ Declarative approach
  - Empty KB → TELL the agent the facts (sentences) one by one until it knows how to operate in its environment.
- ❑ Procedural approach
  - Encode desired behaviors directly as program code.
- ❑ Combine approach

→ *Partially autonomous*

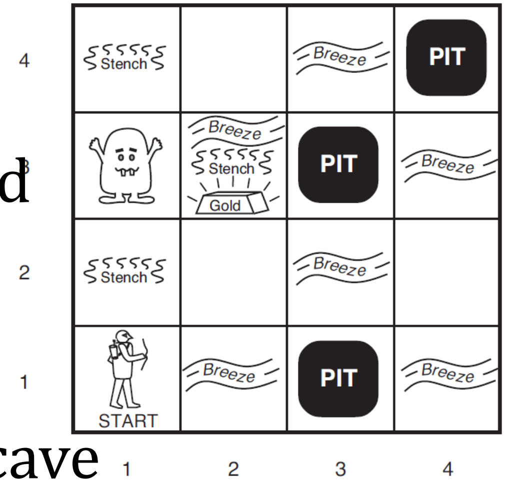
- ❑ Learning approach (chapter 4)
  - Provide a knowledge-based agent with mechanisms that allow it to learn for itself.

→ *Fully autonomous*

The Wumpus World



The Wumpus World - PEAS



❑ Performance measure

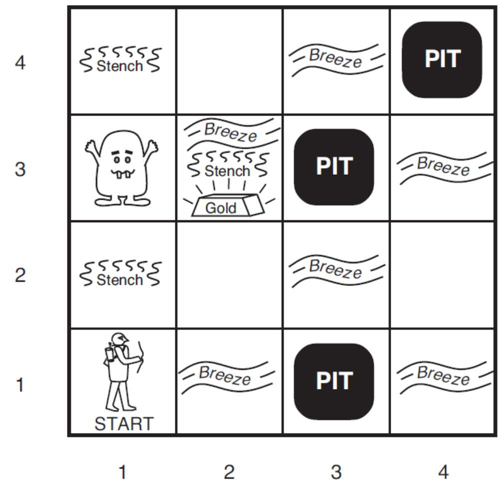
- +1000 for climbing out of the cave with gold
- -1000 for death
- -1 per step, -10 for using the arrow
- Ends when agent dies or climbs out of the cave

❑ Environment

- A 4×4 grid of rooms
- Agent starts in the square [1,1], facing to the right
- Gold and Wumpus locations are random
- Each square can be a pit, with probability 0.2

The Wumpus World - PEAS

- ❑Actuators:
  - Move forward, TurnLeft/Right 90°
  - Grab, Shoot, Climb
- ❑Sensors: 5 sensors to perceive:
  - Stench
  - Breeze
  - Glitter
  - Bump
  - Scream



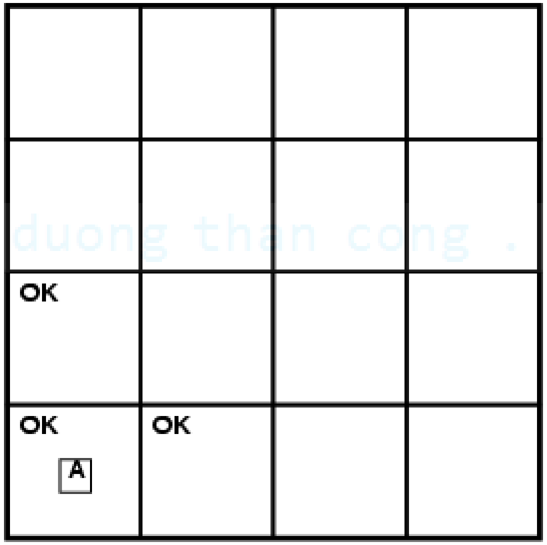
Percept: [Stench, Breeze, None, None, None]

The Wumpus World - Characterization

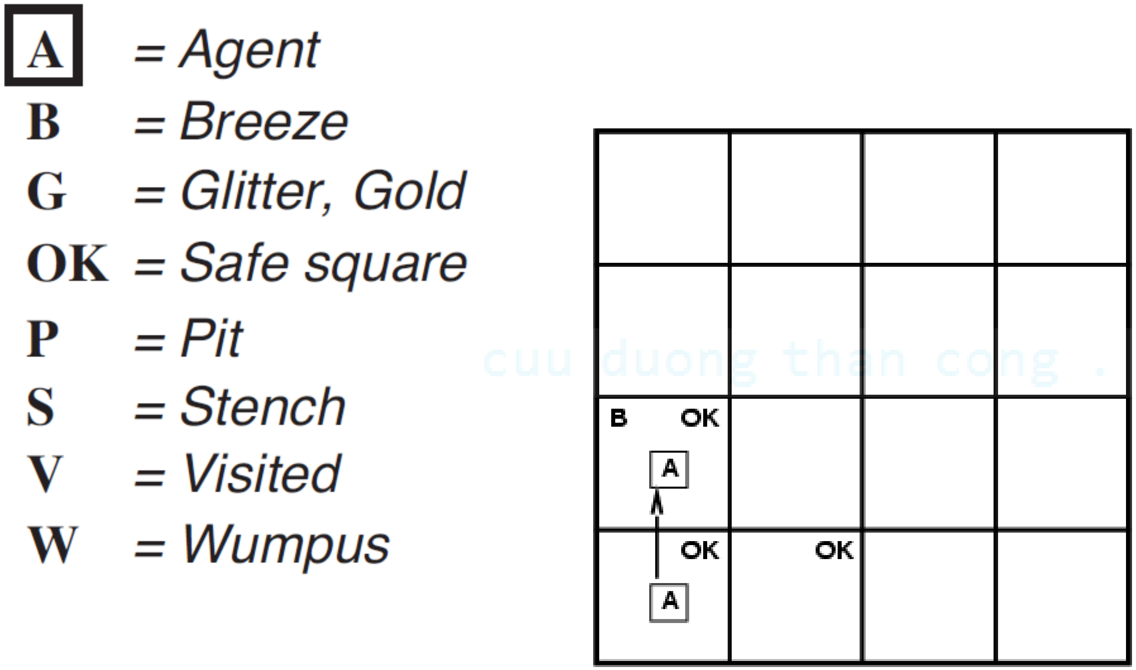
- Fully Observable: No – only local perception
- Deterministic: Yes – outcomes exactly specified
- Episodic: No – sequential at the level of actions
- Static: Yes – Wumpus and Pits do not move
- Discrete: Yes
- Single-agent: Yes – Wumpus is essentially a natural feature

Exploring a wumpus world

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

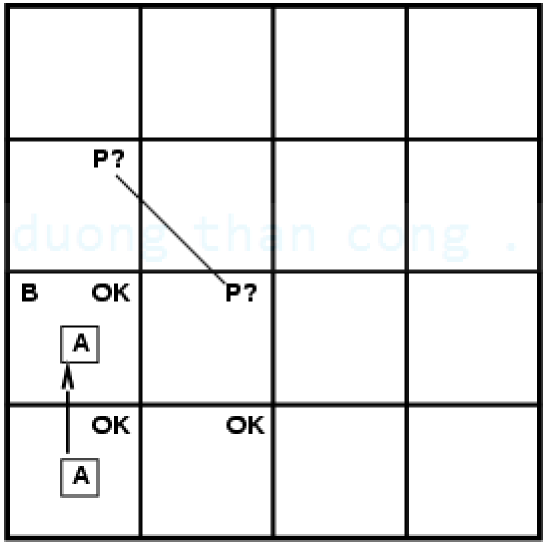


Exploring a wumpus world



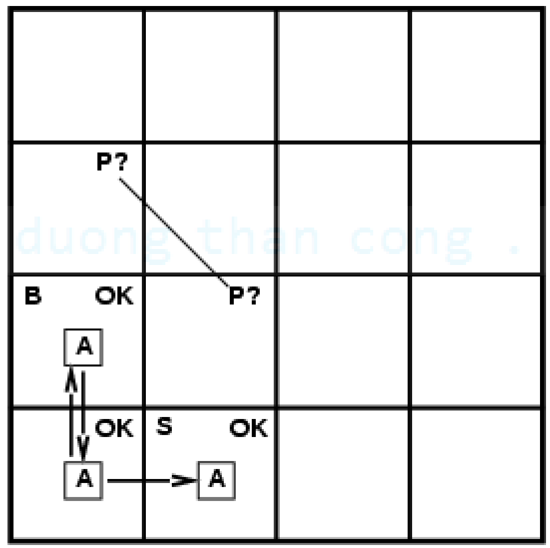
Exploring a wumpus world

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



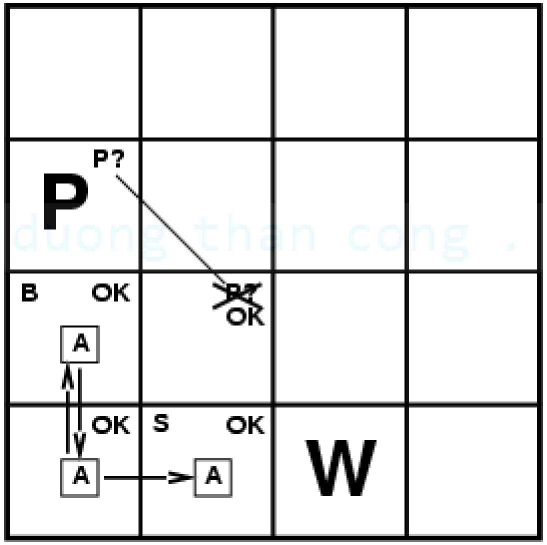
Exploring a wumpus world

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

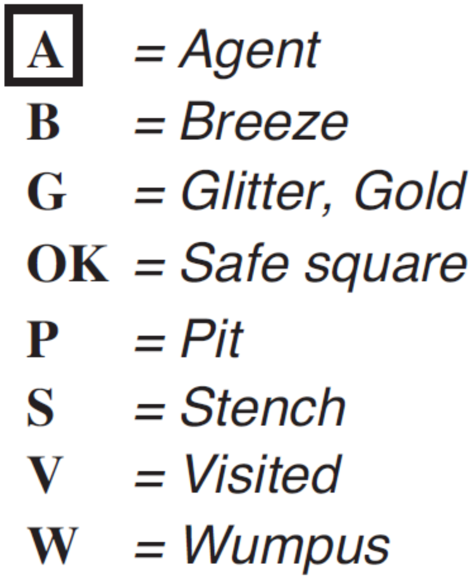


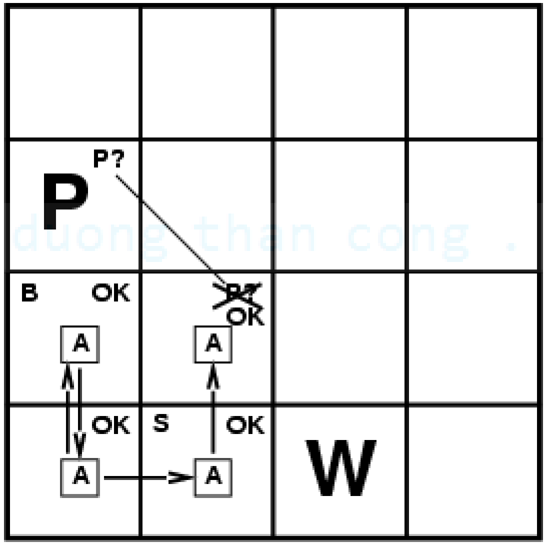
Exploring a wumpus world

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



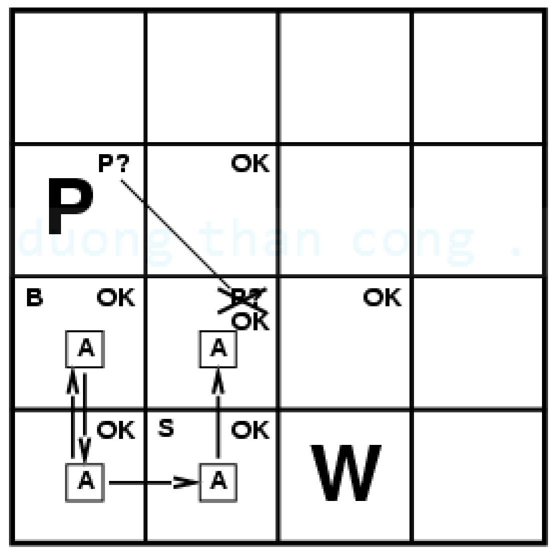
Exploring a wumpus world





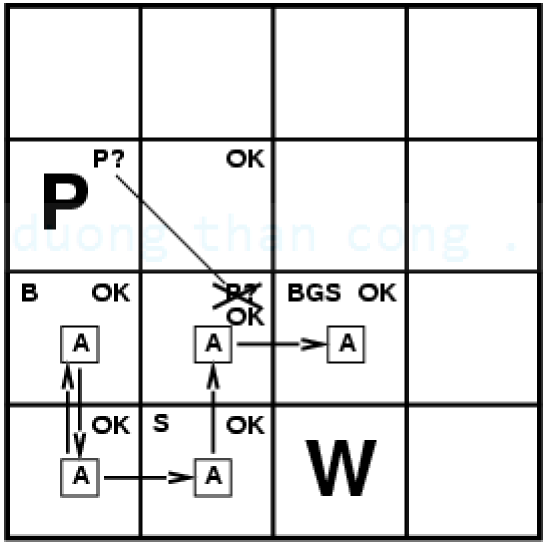
Exploring a wumpus world

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



Exploring a wumpus world

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



Logic in general

- Logics are formal languages for representing information such that conclusions can be drawn
- Syntax defines the sentences in the language
- Semantics define the "meaning" of sentences
  - i.e., define truth of a sentence in a world
- E.g., the language of arithmetic
  - $x+2 \geq y$ is a sentence; $x2+y > \{\}$ is not a sentence
  - $x+2 \geq y$ is true iff the number $x+2$ is no less than the number y
  - $x+2 \geq y$ is true in a world where $x = 7, y = 1$
  - $x+2 \geq y$ is false in a world where $x = 0, y = 6$

Entailment

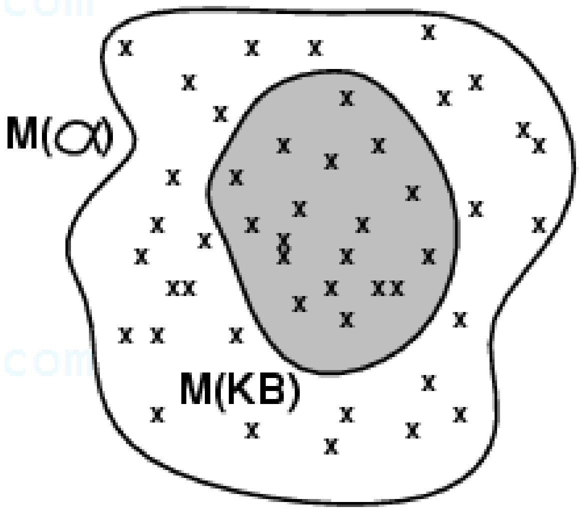
- Entailment means that one thing follows from another:

$$KB \models \alpha$$

- Knowledge base *KB* entails sentence α if and only if α is true in all worlds where *KB* is true
  - E.g., the KB containing "Apple is red" and "Tomato is red" entails "Either the apple or the tomato is red"
  - E.g., x+y = 4 entails 4 = x+y
  - Entailment is a relationship between sentences (i.e., syntax) that is based on semantics

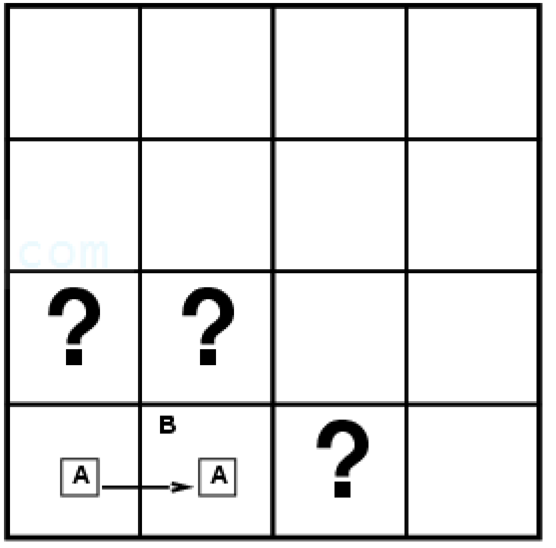
Models

- Logicians typically think in terms of models, which are formally structured worlds with respect to which truth can be evaluated
- We say *m* is a model of a sentence α if α is true in *m*
- *M(α)* is the set of all models of α
- Then KB $\models$ α iff $M(KB) \subseteq M(\alpha)$
  - E.g. *KB* = Apple and tomato are red
  - α = Apple is red



Entailment in the wumpus world

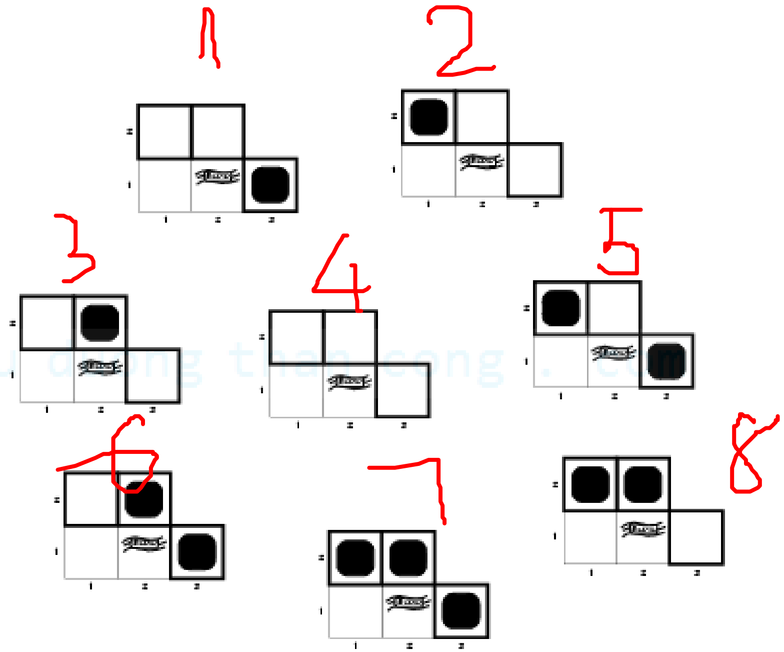
Situation after detecting nothing in [1,1], moving right, breeze in [2,1]



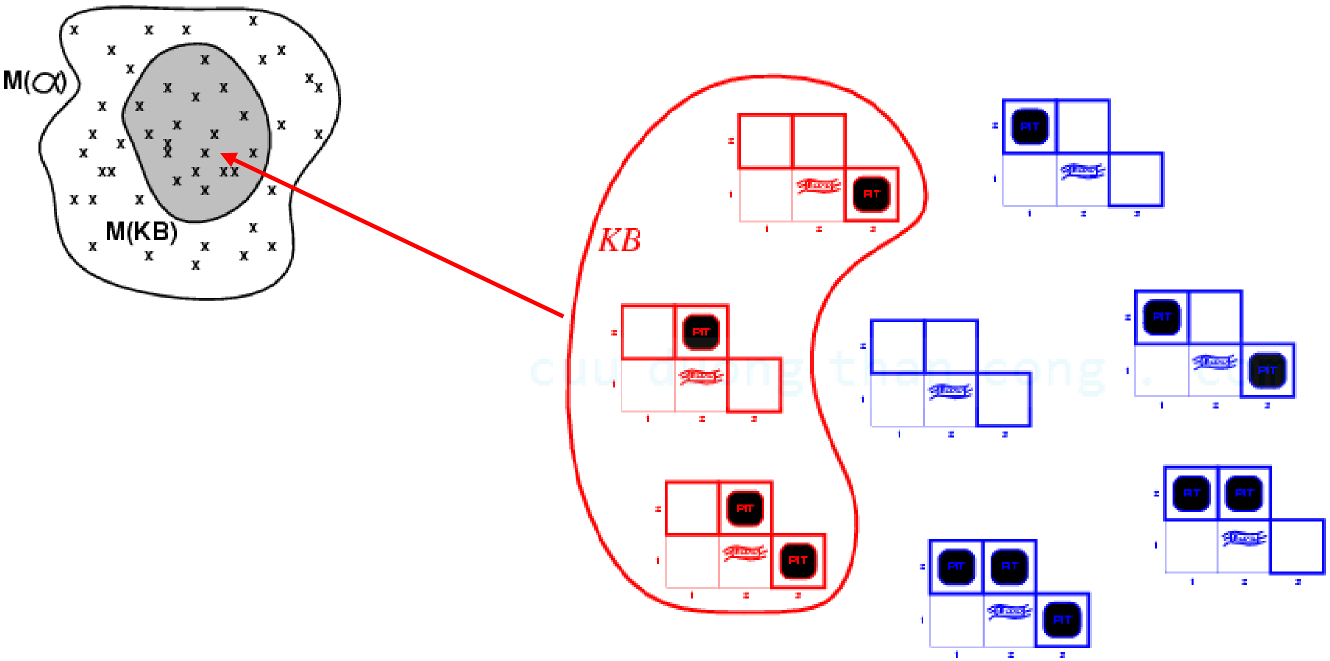
Consider possible models for *KB* assuming only pits

3 Boolean choices $\Rightarrow$ 8 possible models

Wumpus models

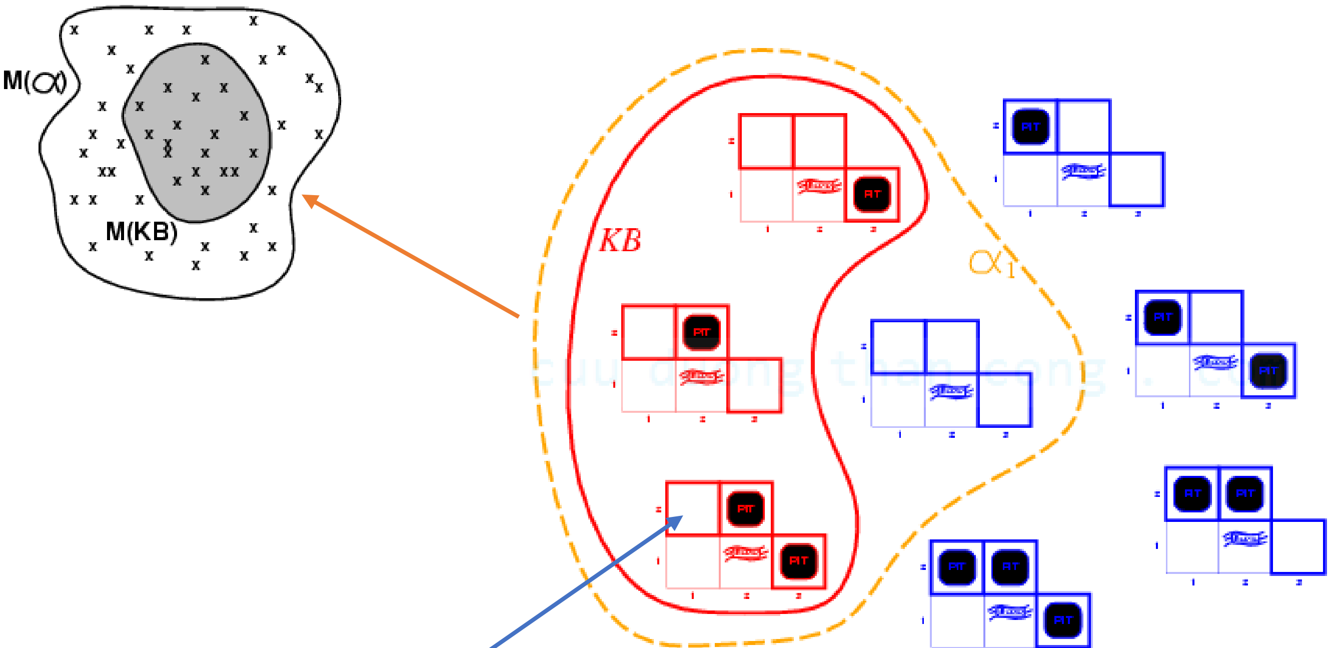


Wumpus models



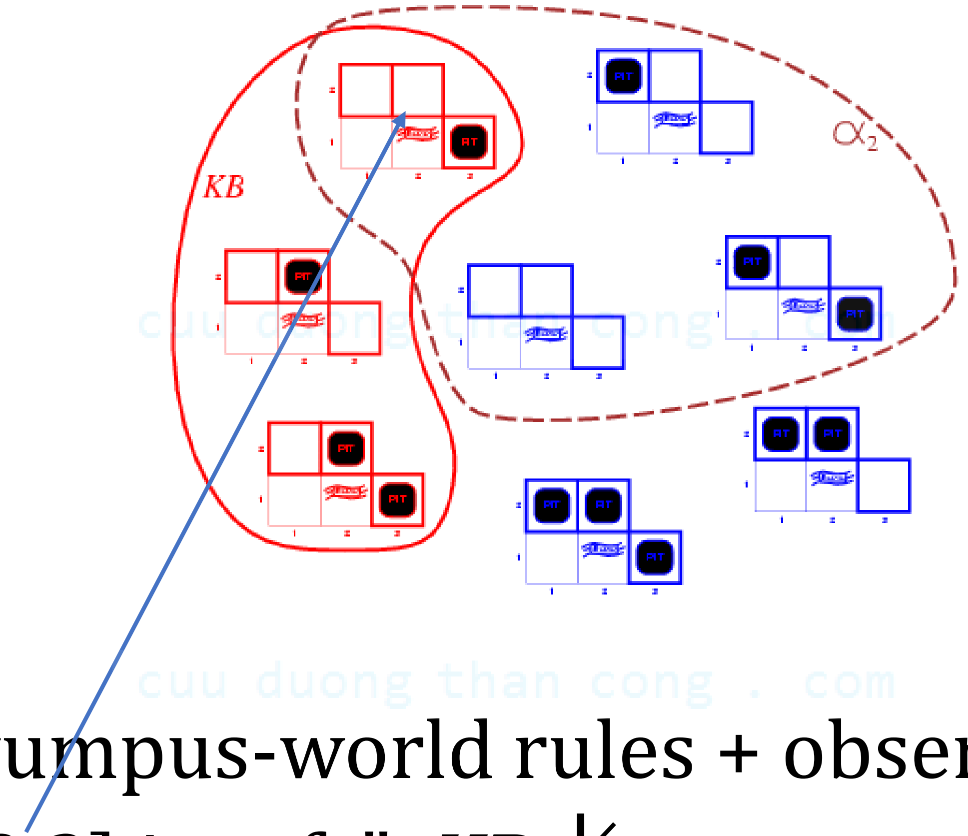
❑*KB* = wumpus-world rules + observations

Wumpus models



- *KB* = wumpus-world rules + observations
- α_1 = "[1,2] is safe", $KB \models \alpha_1$, proved by model checking

Wumpus models



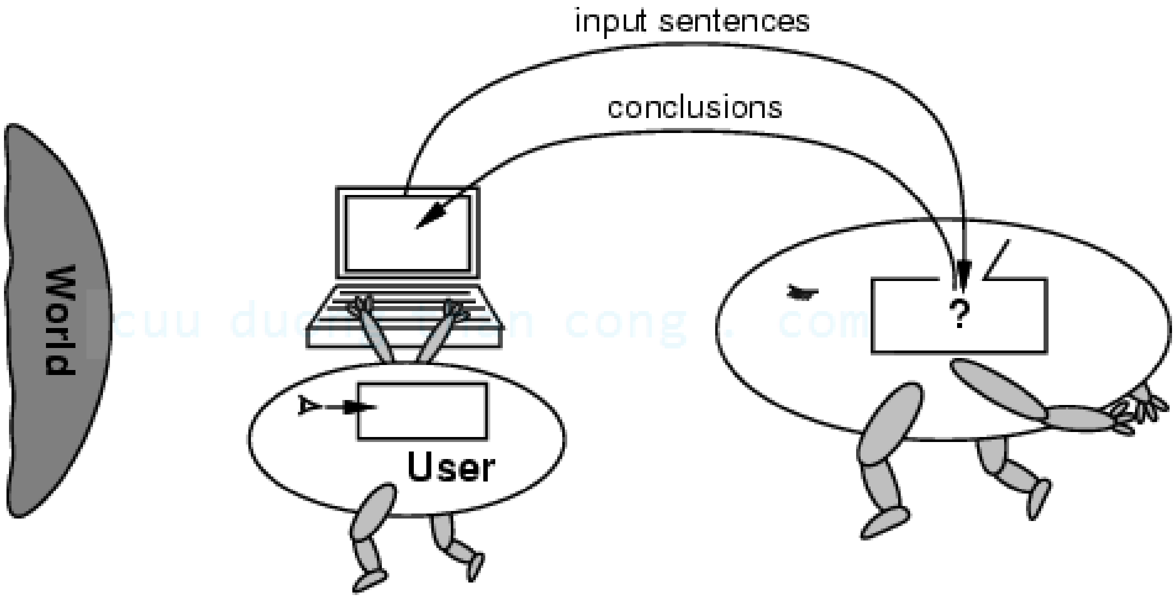
- KB = wumpus-world rules + observations
- α_2 = "[2,2] is safe", $KB \not\models \alpha_2$

Inference

- $KB \vdash_i \alpha$ = sentence α can be derived from *KB* by procedure *i*
- Soundness: *i* is sound if whenever $KB \vdash_i \alpha$, it is also true that $KB \models \alpha$
- Completeness: *i* is complete if whenever $KB \models \alpha$, it is also true that $KB \vdash_i \alpha$
- Preview: we will define a logic (first-order logic) which is expressive enough to say almost anything of interest, and for which there exists a sound and complete inference procedure.
- That is, the procedure will answer any question whose answer follows from what is known by the *KB*.

No independent access to the world

- The reasoning agent often gets its knowledge about the facts of the world as a sequence of logical sentences and must draw conclusions only from them without independent access to the world.
- Thus it is very important that the agent's reasoning is sound!



Summary

- Intelligent agents need knowledge about the world for making good decisions.
- The knowledge of an agent is stored in a knowledge base in the form of **sentences** in a knowledge representation language.
- A knowledge-based agent needs a **knowledge base** and an **inference mechanism**. It operates by storing sentences in its knowledge base, inferring new sentences with the inference mechanism, and using them to deduce which actions to take.
- A **representation language** is defined by its syntax and semantics, which specify the structure of sentences and how they relate to the facts of the world.
- The **interpretation** of a sentence is the fact to which it refers. If this fact is part of the actual world, then the sentence is true.