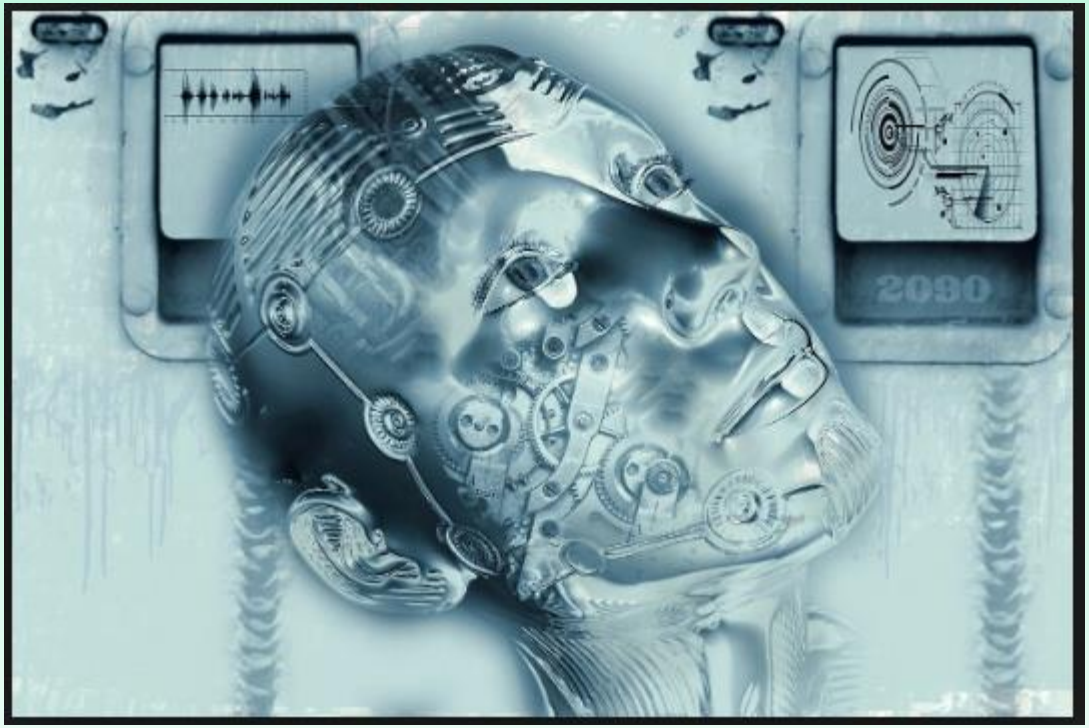
Knowledge-based Agents
Planning

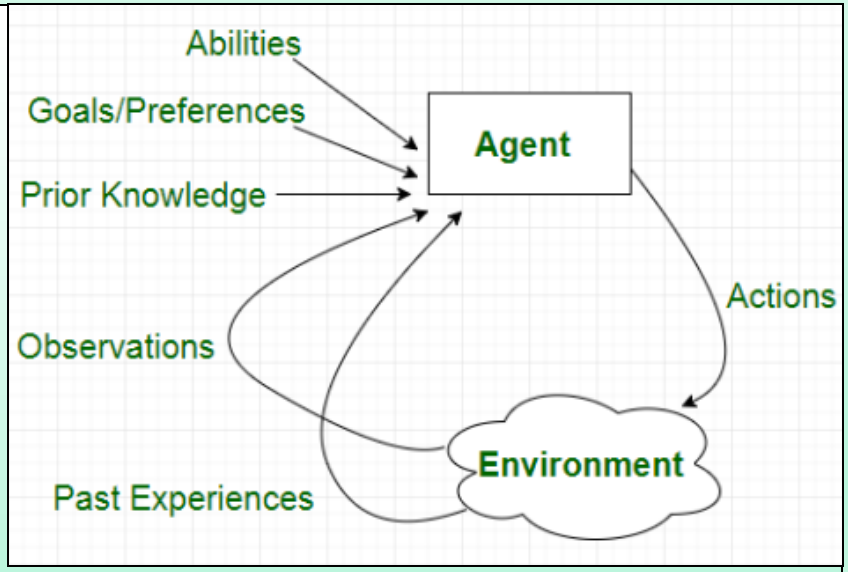


Knowledge-Based Agents - Topics

- Introduction
- Knowledge-Based Agents
- WUMPUS WORLD Environment
- Propositional Logic
- First Order Predicate Logic
- Forward and Backward Chaining

Introduction

- Human beings know things
- This helps them to do things intelligently based on reasoning



- Process of reasoning operates based on **internal representation (storage) of knowledge**
- Same approach is followed by **Knowledge-based Agents**
- **Logic** is a class of representation that supports Knowledge-based Agents
- They can adopt to changes in env. by updating knowledge

Knowledge-based Agents (Design)

- **Central component – knowledge base (KB)**
- **Knowledge Base** – Set of **sentences** expressed in **Knowledge Representation Language**
- **Operations**
  - **TELL** – **Add** new sentence to KB
  - **ASK** – **Query** what is known
- An **KB Agent program** takes a **percept** as **input** & **returns** an **action**
- The **KB** initially contains some **"background knowledge"**
- The **Agent program** does 3 things
  - **TELLs** the **KB** what it **perceives**
  - **ASKs** the **KB** what **action should be performed**
  - **TELLs** the **KB** what **action was chosen** & **executes the action**

KB Agents Program

Agent KB-Agent (Percept) Returns an action

Persistent: KB – a knowledge base // Maintain a KB
t (time) = 0 //time is initialized to 0

// Input percept sequence & time to KB
TELL (KB, Make-Percept-Sentence (percept, t))

// Find suitable action to be taken from KB
action = ASK (KB, Make-Action-Query (t))

// Update KB with action corresponding to the percept seq at time t
TELL (KB, Make-Action-Sentence (percept, t)

t = t + 1 // Increment time

return action // Return action

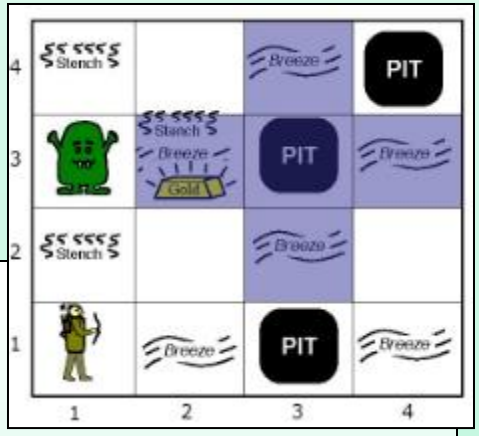
KB Agents Program

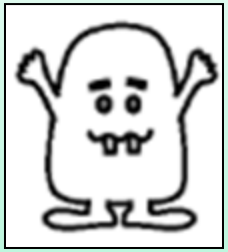
Two System building approaches employed by a **designer** to an empty KB

1. **Declarative approach**
   - **TELL** sentences one-by-one until the agent knows how to operate
2. **Procedural approach**
   - **Encodes** desired behavior directly into program code

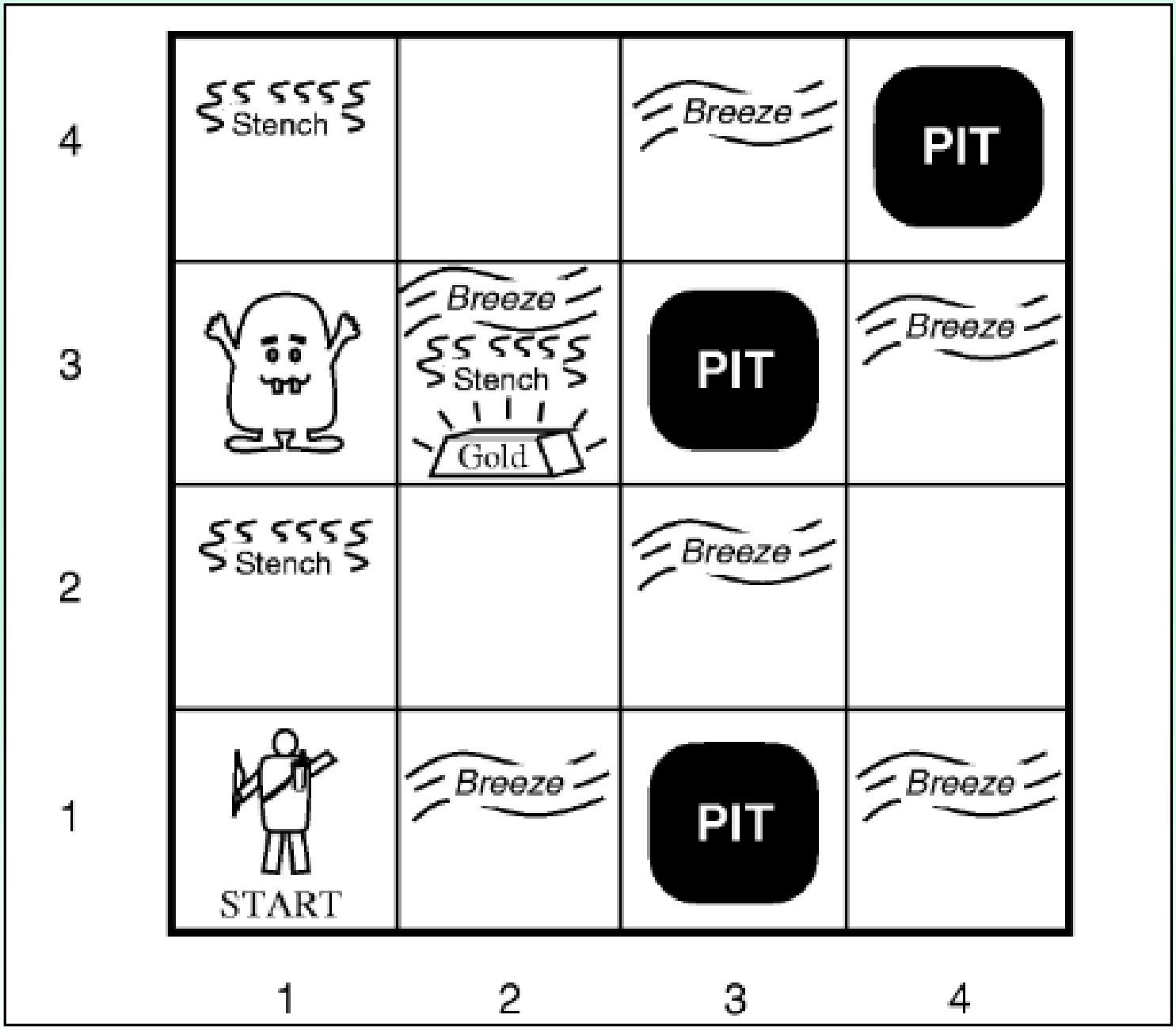
A successful agent must combine both approaches

The Wumpus World Environment



- **Wumpus World**
- A **cave** containing rooms connected by passageways
- The **Wumpus (beast)**  hidden in the cave – Eats anyone entering the room
- The **Agent**  has only one arrow to shoot
- Some rooms has bottom-less **pits** 

to trap anyone entering
- **Only Reward** - Possibility of finding a gold heap 


4
3
2
1
Stench
Breeze
PIT
Stench
Breeze
Gold
PIT
Breeze
Stench
Breeze
START
Breeze
PIT
Breeze
1
2
3
4

Task Environment Description - PEAS

- **Performance measure**
  - **+1000** – Coming out of cave with gold
  - **-1000** – Falling into Pit or Eaten by Wumpus
  - **-1** – For each action
  - **-10** – For using the arrow
  - **End of game** – Agent dies or climbs out of cave
- **Env**
  - A **4X4 grid** of **rooms**
  - **Agent** starts in **[1,1]**
  - Location of **Gold** & **Wumpus** chosen randomly (except starting one)
  - Each square (except starting one) can be a pit with probability 0.2

Task Environment Description - PEAS

- **Actuators**
- **Agent Moves** – ***Forward, TurnLeft, TurnRight***
- **Death** – Falling into Pit or Eaten by Wumpus (Safe to enter room with dead wumpus)
- **Forward move against wall**– Not allowed
- **Actions**– **Grab** (pickup gold), **Shoot** (one Arrow), **Climb** (out of cave from [1,1])
- **End of game** – Agent dies or climbs out of cave

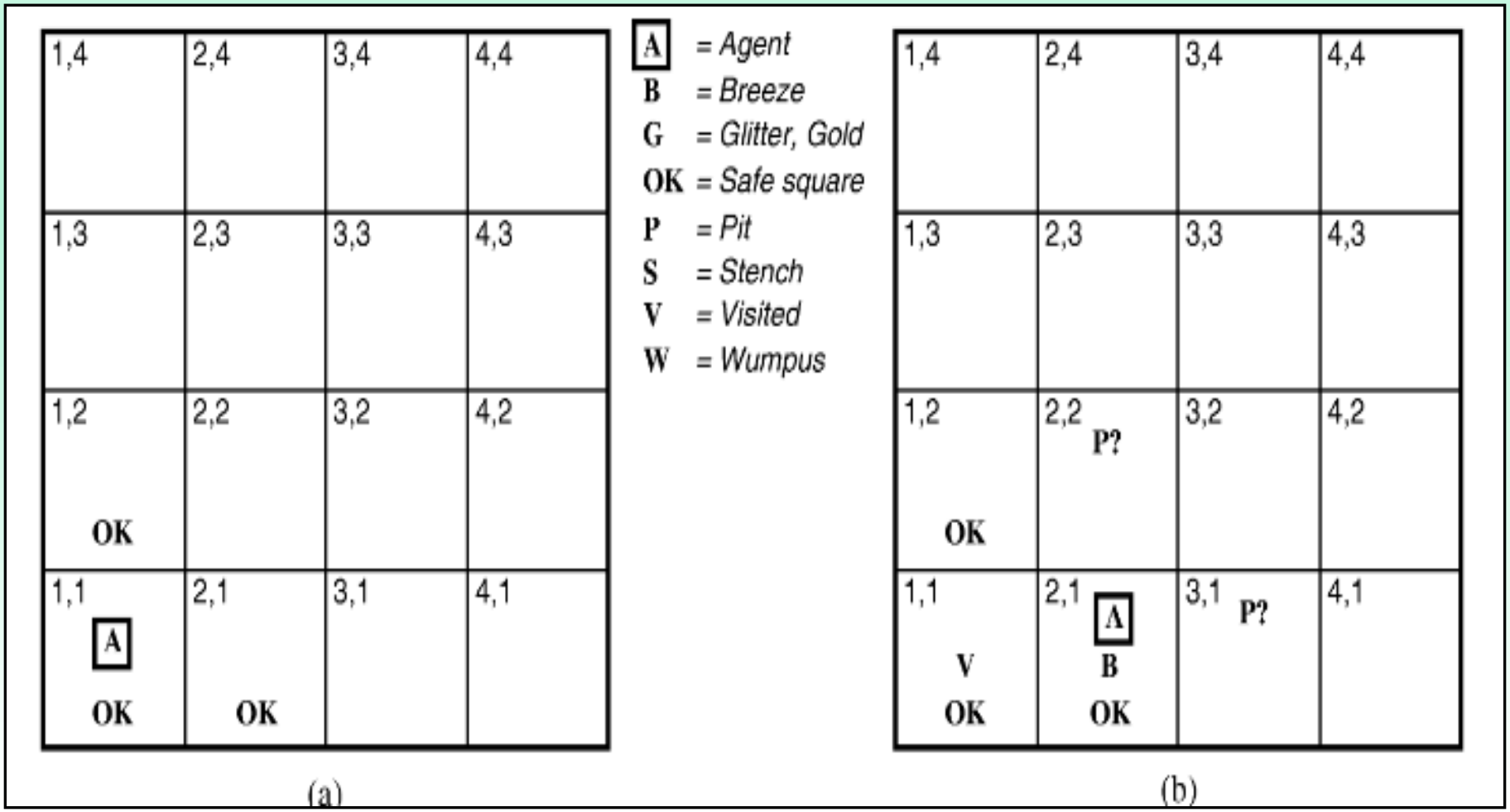
- **Sensors**
- **Stench**: Perceived in squares **containing** & **adjacent** to **wumpus**
- **Breeze**: Perceived in squares **adjacent** to a **pit**
- **Glitter**: Perceived in squares containing **Gold**
- **Bump**: Perceived when walking into a **Wall**
- **Kill Wumpus**: Perceived **Scream** anywhere in the cave

Wumpus World - Steps

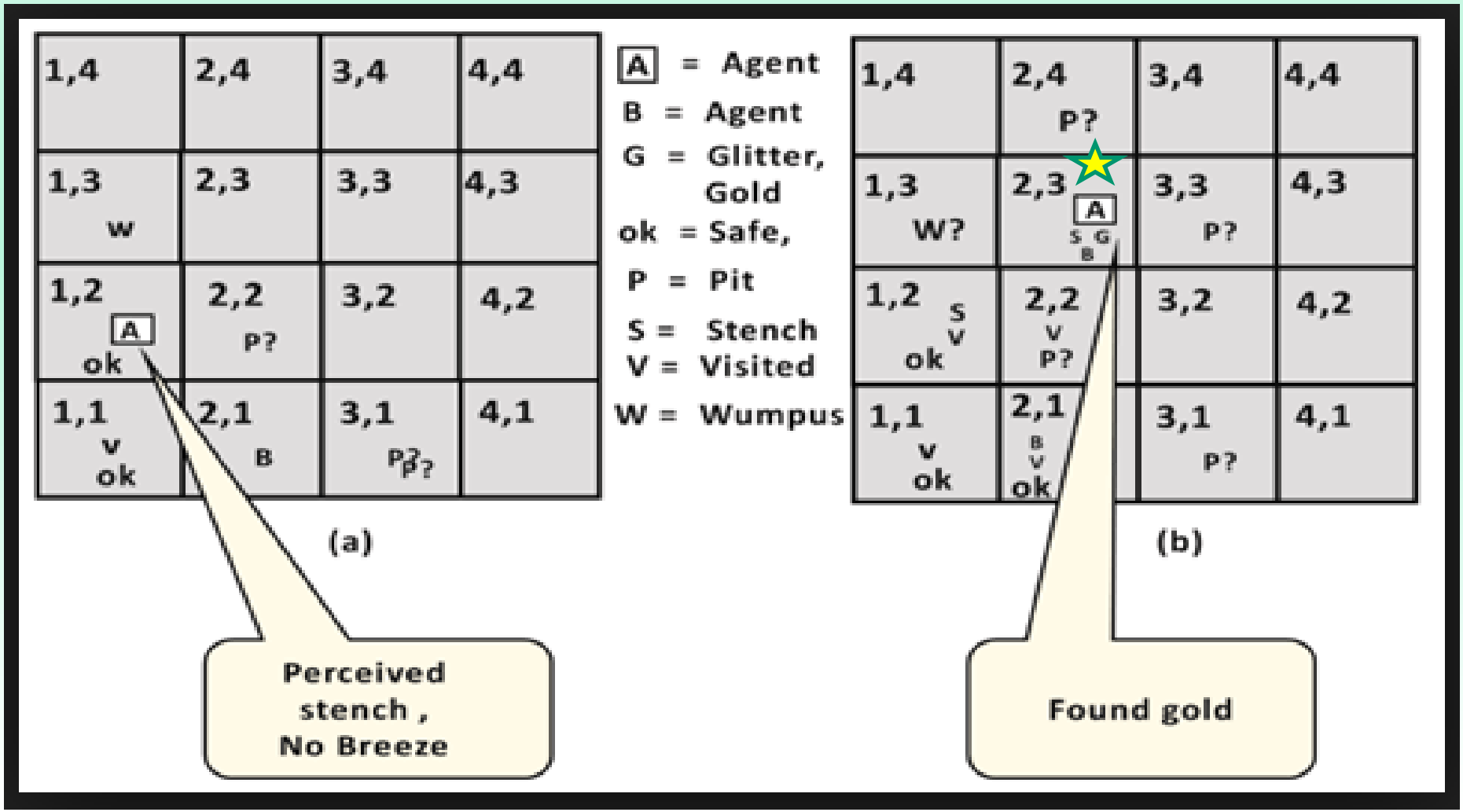
- Challenges for Agent - Initial ignorance of env configuration (require logical reasoning)
- *Good possibility of agent getting out **with gold***
- *Sometimes, agent will have to **choose** between **empty-hand return** or **death***
- ***21%** times **gold** is in a **pit** or **surrounded by pits***
- **Knowledge Representation Language (KRL) used –** writing **symbols** in the **grids [Stench, Breeze, Glitter, Bump, Kill Wumpus]**
- **Initial KB –** contains **rules** of the game

Start grid [1,1] & it is **safe** – denoted by **A** (agent) & **OK**

- 1st percept is **[None, None, None, None]** => neibouring grids **[1,2]** & **[2,1]** are safe (**OK**)
- If **Agent** moves to **[2,1]** => Perceives **breeze** (**B**)=>**Pit(s)** present in **[2,2]** or **[3,1]** or **both (P?)**
- Only safe square is **[1,2].** Hence agent should move back to **[1,1]** & then to **[1,2]**



- In [1,2] perceived **Stench** & **No breeze** – denoted by **S - [Stench, None, None, None, None]**
- After 5th move perceived **[Stench, Breeze, Glitter, None, None]** => **Found Gold**



THANK YOU !!!