INTRODUCTION TO ARTIFICIAL INTELLIGENCE AND ITS APPLICATIONS

Code:19EAI232

Dr. Praveen Gupta,
Asst. Professor
Department of Computer Science and Engineering,
GITAM Institute of Technology,
GITAM (Deemed to be University), Visakhapatnam(AP), India
Email Id: pgupta@gitam.edu , M: 9783750898

MODULE III: ROADMAP

- Constraint Satisfaction Problems:
  - ✓ Defining Constraint Satisfaction Problems,
  - ✓ Constraint Propagation:
    - Inference in CSPs,
    - Backtracking Search for CSPs,
    - Local Search for CSPs,
    - The Structure of Problems.

- Local Agents:
  - ✓ Knowledge-based Agents,
  - ✓ The Wumpus World,
  - ✓ Logic,
    - Propositional Logic:
    - A Very Simple Logic,
    - Propositional Theorem Proving.

LEARNING OUTCOMES:

After completion of this module, the student will be able to:

- define constraint satisfaction problems (L1)
- illustrate inference in constraint satisfaction problems (L2)
- contrast backtracking search and local search for constraint satisfaction problems (L2)
- explain knowledge - based agent (L2)
- apply propositional logic for real time problems (L3)

MODULE III: ROADMAP

- Constraint Satisfaction Problems:
  - ✓ Defining Constraint Satisfaction Problems,
  - ✓ Constraint Propagation:
    - Inference in CSPs,
    - Backtracking Search for CSPs,
    - Local Search for CSPs,
    - The Structure of Problems.

- Local Agents:
  - ✓ Knowledge-based Agents,
  - ✓ The Wumpus World,
  - ✓ Logic,
    - Propositional Logic:
    - A Very Simple Logic,
    - Propositional Theorem Proving.

LOCAL AGENTS: KNOWLEDGE-BASED AGENTS

- An intelligent agent needs knowledge about the real world for taking decisions and reasoning to act efficiently.
- Knowledge-based agents are those agents who have the capability of maintaining an internal state of knowledge, reason over that knowledge, update their knowledge after observations and take actions.
- These agents can represent the world with some formal representation and act intelligently.
- Knowledge-based agents are composed of two main parts:

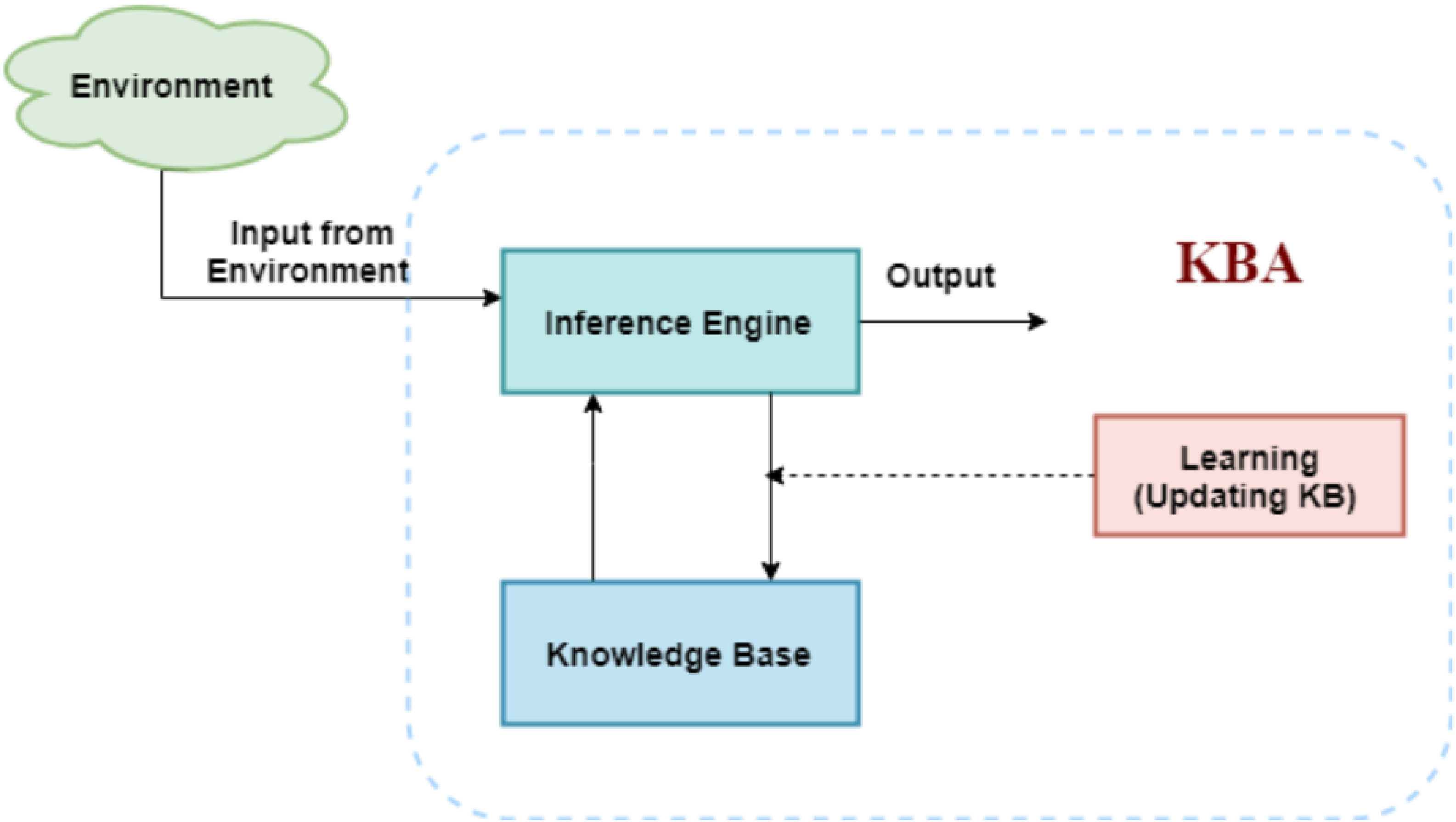
  i. Knowledge-base and

  ii. Inference system.

LOCAL AGENTS: KNOWLEDGE-BASED AGENTS

- A knowledge-based agent must able to do the following:
  - An agent should be able to represent states, actions, etc.
  - An agent Should be able to incorporate new percepts
  - An agent can update the internal representation of the world
  - An agent can deduce the internal representation of the world
  - An agent can deduce appropriate actions.

THE ARCHITECTURE OF KNOWLEDGE-BASED AGENT



THE ARCHITECTURE OF KNOWLEDGE-BASED AGENT

- The above diagram is representing a generalized architecture for a knowledge-based agent. The knowledge-based agent (KBA) take input from the environment by perceiving the environment.
- The input is taken by the inference engine of the agent and which also communicate with KB to decide as per the knowledge store in KB. The learning element of KBA regularly updates the KB by learning new knowledge.
- Knowledge base: Knowledge-base is a central component of a knowledge-based agent, it is also known as KB. It is a collection of sentences (here 'sentence' is a technical term and it is not identical to sentence in English). These sentences are expressed in a language which is called a knowledge representation language. The Knowledge-base of KBA stores fact about the world.

OPERATIONS PERFORMED BY KNOWLEDGE-BASED AGENTS

GITAM
(DEEMED TO BE UNIVERSITY)
VISAKHAPATNAM • HYDERABAD • BENGALURU

- Following are three operations which are performed by KBA in order to show the intelligent behavior:
  - TELL: This operation tells the knowledge base what it perceives from the environment.
  - ASK: This operation asks the knowledge base what action it should perform.
  - Perform: It performs the selected action.

```
function KB-AGENT(percept):
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

- The knowledge-based agent takes percept as input and returns an action as output. The agent maintains the knowledge base, KB, and it initially has some background knowledge of the real world. It also has a counter to indicate the time for the whole process, and this counter is initialized with zero.
- Each time when the function is called, it performs its three operations:
- Firstly it TELLs the KB what it perceives.
- Secondly, it asks KB what action it should take
- Third agent program TELLS the KB that which action was chosen.
- The MAKE-PERCEPT-SENTENCE generates a sentence as setting that the agent perceived the given percept at the given time.
- The MAKE-ACTION-QUERY generates a sentence to ask which action should be done at the current time.
- MAKE-ACTION-SENTENCE generates a sentence which asserts that the chosen action was executed.

VARIOUS LEVELS OF KNOWLEDGE-BASED AGENT

A knowledge-based agent can be viewed at different levels which are given below:

1. Knowledge level

- Knowledge level is the first level of knowledge-based agent, and in this level, we need to specify what the agent knows, and what the agent goals are. With these specifications, we can fix its behavior. For example, suppose an automated taxi agent needs to go from a station A to station B, and he knows the way from A to B, so this comes at the knowledge level.

2. Logical level:

- At this level, we understand that how the knowledge representation of knowledge is stored. At this level, sentences are encoded into different logics. At the logical level, an encoding of knowledge into logical sentences occurs. At the logical level we can expect to the automated taxi agent to reach to the destination B.

3. Implementation level:

- This is the physical representation of logic and knowledge. At the implementation level agent perform actions as per logical and knowledge level. At this level, an automated taxi agent actually implement his knowledge and logic so that he can reach to the destination.

APPROACHES TO DESIGNING A KNOWLEDGE-BASED AGENT

There are mainly two approaches to build a knowledge-based agent:

1. Declarative approach: We can create a knowledge-based agent by initializing with an empty knowledge base and telling the agent all the sentences with which we want to start with. This approach is called Declarative approach.

2. Procedural approach: In the procedural approach, we directly encode desired behavior as a program code. Which means we just need to write a program that already encodes the desired behavior or agent.

However, in the real world, a successful agent can be built by combining both declarative and procedural approaches, and declarative knowledge can often be compiled into more efficient procedural code.

MODULE III: ROADMAP

- Constraint Satisfaction Problems:
  - ✓ Defining Constraint Satisfaction Problems,
  - ✓ Constraint Propagation:
    - Inference in CSPs,
    - Backtracking Search for CSPs,
    - Local Search for CSPs,
    - The Structure of Problems.

- Local Agents:
  - ✓ Knowledge-based Agents,
  - ✓ The Wumpus World,
  - ✓ Logic,
    - Propositional Logic:
    - A Very Simple Logic,
    - Propositional Theorem Proving.

THE WUMPUS WORLD

GITAM (DEEMED TO BE UNIVERSITY) VISAKHAPATNAM • HYDERABAD • BENGALURU

- The Wumpus world is a simple world example to illustrate the worth of a knowledge-based agent and to represent knowledge representation.
- The Wumpus world is a cave which has 4/4 rooms connected with passageways. So there are total 16 rooms which are connected with each other. We have a knowledge-based agent who will go forward in this world.
- The cave has a room with a beast which is called Wumpus, who eats anyone who enters the room. The Wumpus can be shot by the agent, but the agent has a single arrow. In the Wumpus world, there are some Pits rooms which are bottomless, and if agent falls in Pits, then he will be stuck there forever.
- The exciting thing with this cave is that in one room there is a possibility of finding a heap of gold. So the agent goal is to find the gold and climb out the cave without fallen into Pits or eaten by Wumpus. The agent will get a reward if he comes out with gold, and he will get a penalty if eaten by Wumpus or falls in the pit.
- https://thiagodnf.github.io/wumpus-world-simulator/

THE WUMPUS WORLD: EXAMPLE

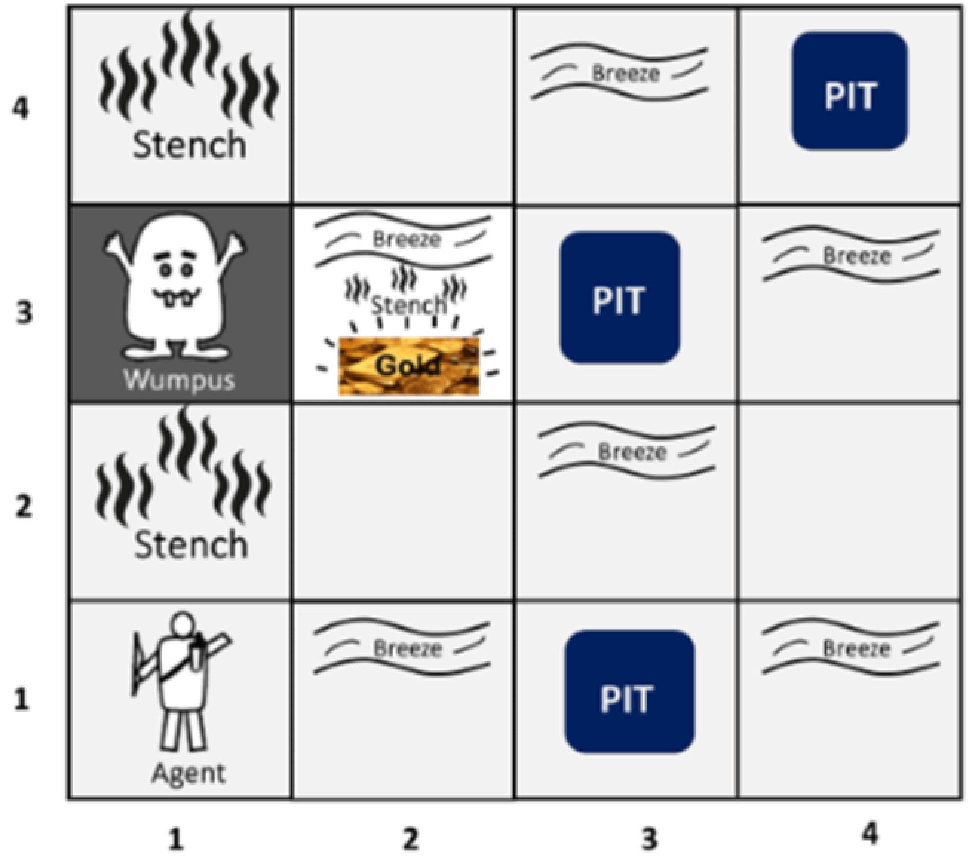
- Following is a sample diagram for representing the Wumpus world. It is showing some rooms with Pits, one room with Wumpus and one agent at (1, 1) square location of the world.
- There are also some components which can help the agent to navigate the cave. These components are given as follows:

➢The rooms adjacent to the Wumpus room are smelly, so that it would have some stench.

➢The room adjacent to PITs has a breeze, so if the agent reaches near to PIT, then he will perceive the breeze.

➢There will be glitter in the room if and only if the room has gold.

➢The Wumpus can be killed by the agent if the agent is facing to it, and Wumpus will emit a horrible scream which can be heard anywhere in the cave. https://thiagodnf.github.io/wumpus-world-simulator/



PEAS DESCRIPTION OF WUMPUS WORLD

- Performance measure:
  - +1000 reward points if the agent comes out of the cave with the gold.
  - -1000 points penalty for being eaten by the Wumpus or falling into the pit.
  - -1 for each action, and -10 for using an arrow.
  - The game ends if either agent dies or came out of the cave.
- Environment:
  - A 4*4 grid of rooms.
  - The agent initially in room square [1, 1], facing toward the right.
  - Location of Wumpus and gold are chosen randomly except the first square [1,1].
  - Each square of the cave can be a pit with probability 0.2 except the first square.

PEAS DESCRIPTION OF WUMPUS WORLD

- Actuators:

  - Left turn, Right turn, Move forward, Grab

- Sensors:

  - The agent will perceive the stench if he is in the room adjacent to the Wumpus. (Not diagonally). The agent will perceive breeze if he is in the room directly adjacent to the Pit.

  - The agent will perceive the glitter in the room where the gold is present.

  - The agent will perceive the bump if he walks into a wall.

  - When the Wumpus is shot, it emits a horrible scream which can be perceived anywhere in the cave.

  - These percepts can be represented as five element list, in which we will have different indicators for each sensor.

  - Example if agent perceives stench, breeze, but no glitter, no bump, and no scream then it can be represented as: [Stench, Breeze, None, None, None].

THE WUMPUS WORLD PROPERTIES

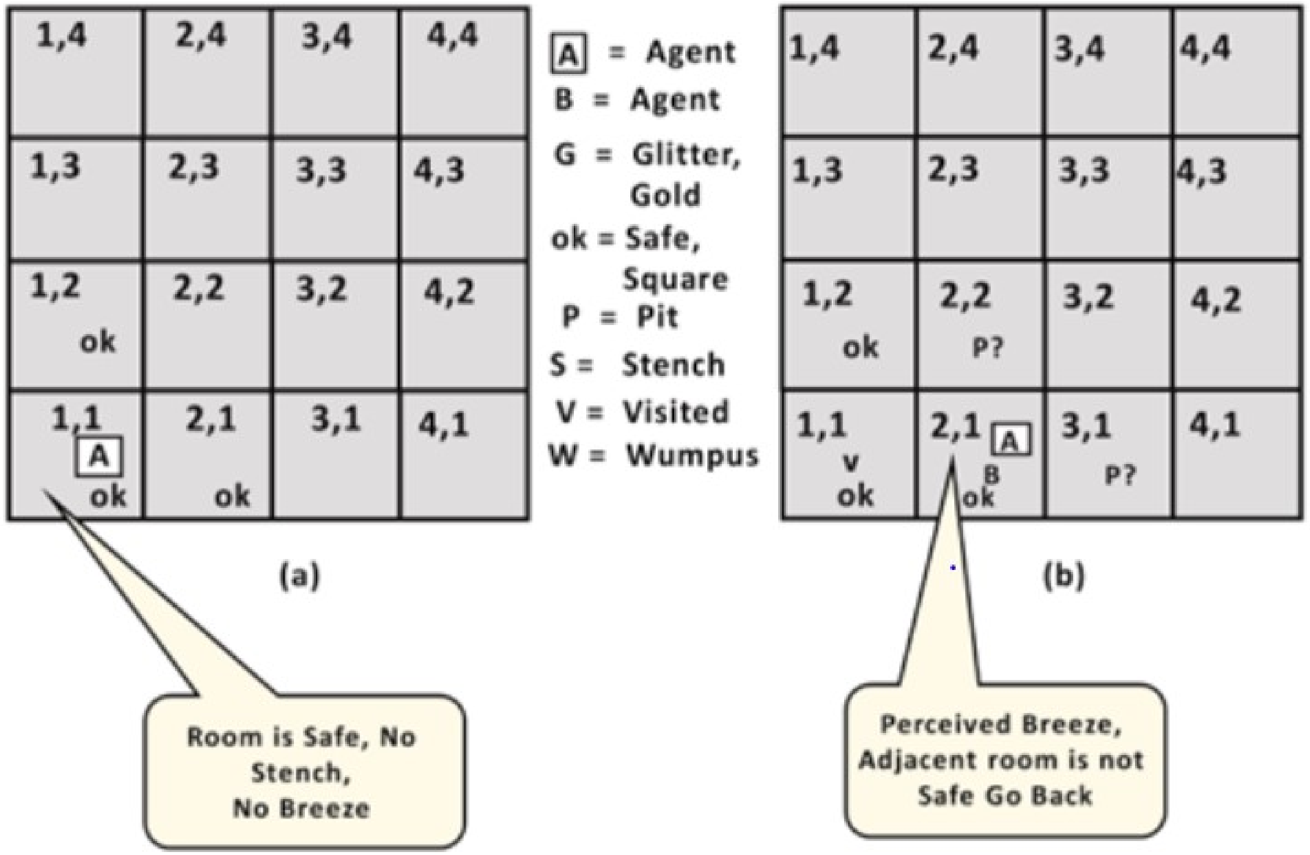
- Partially observable: The Wumpus world is partially observable because the agent can only perceive the close environment such as an adjacent room.
- Deterministic: It is deterministic, as the result and outcome of the world are already known.
- Sequential: The order is important, so it is sequential.
- Static: It is static as Wumpus and Pits are not moving.
- Discrete: The environment is discrete.
- One agent: The environment is a single agent as we have one agent only and Wumpus is not considered as an agent.

EXPLORING THE WUMPUS WORLD

- Now we will see how the agent will find its goal by applying logical reasoning.

1. Agent's First step:

- Initially, the agent is in the first room or on the square [1,1], and we already know that this room is safe for the agent, so to represent on the below diagram (a) that room is safe we will add symbol OK.
- Symbol A is used to represent agent, symbol B for the breeze, G for Glitter or gold, V for the visited room, P for pits, W for Wumpus.
- At Room [1,1] agent does not feel any breeze or any Stench which means the adjacent squares are also OK.



Contd..

2. Agent's second Step:

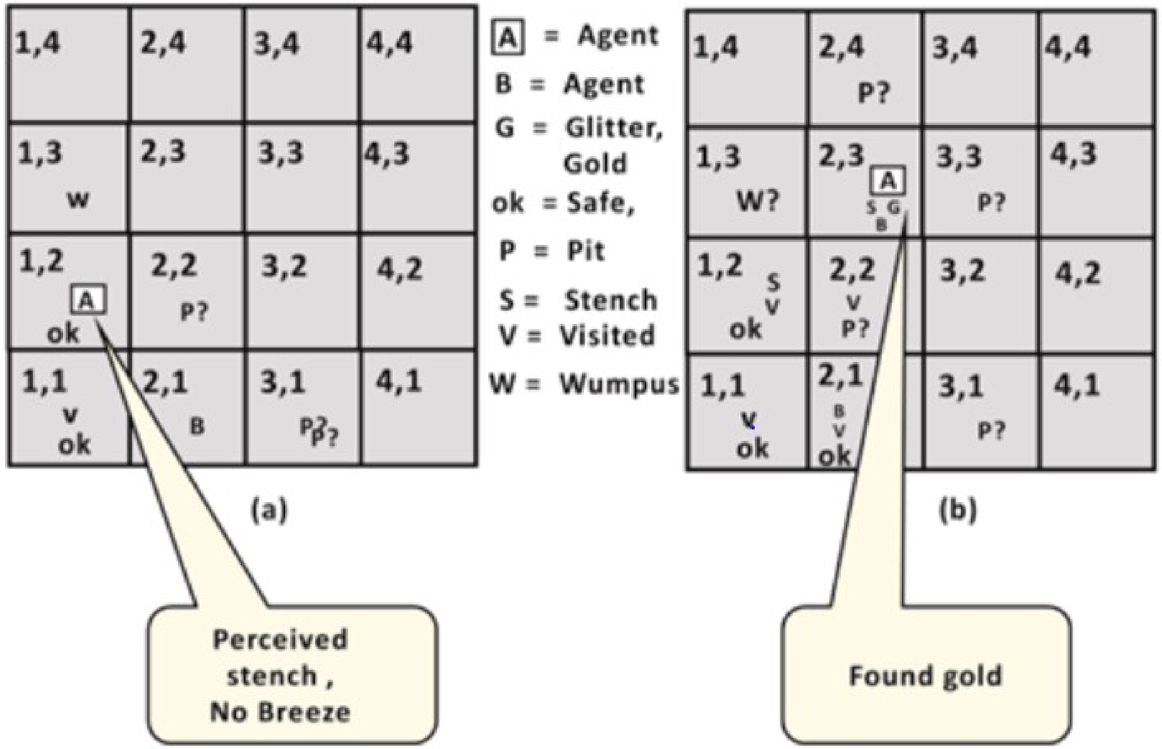
- Now agent needs to move forward, so it will either move to [1, 2], or [2,1]. Let's suppose agent moves to the room [2, 1], at this room agent perceives some breeze which means Pit is around this room. The pit can be in [3, 1], or [2,2], so we will add symbol P? to say that, is this Pit room?
- Now agent will stop and think and will not make any harmful move. The agent will go back to the [1, 1] room. The room [1,1], and [2,1] are visited by the agent, so we will use symbol V to represent the visited squares.

Contd..

3. Agent's third step:

- At the third step, now agent will move to the room [1,2] which is OK. In the room [1,2] agent perceives a stench which means there must be a Wumpus nearby. But Wumpus cannot be in the room [1,1] as by rules of the game, and also not in [2,2] (Agent had not detected any stench when he was at [2,1]).

- Therefore agent infers that Wumpus is in the room [1,3], and in current state, there is no breeze which means in [2,2] there is no Pit and no Wumpus. So it is safe, and we will mark it OK, and the agent moves further in [2,2].



4. **Agent's fourth step:**

- At room [2,2], here no stench and no breezes present so let's suppose agent decides to move to [2,3]. At room [2,3] agent perceives glitter, so it should grab the gold and climb out of the cave.

MODULE III: ROADMAP

- Constraint Satisfaction Problems:
  - ✓ Defining Constraint Satisfaction Problems,
  - ✓ Constraint Propagation:
    - Inference in CSPs,
    - Backtracking Search for CSPs,
    - Local Search for CSPs,
    - The Structure of Problems.

- Local Agents:
  - ✓ Knowledge-based Agents,
  - ✓ The Wumpus World,
  - ✓ Logic:
    - Propositional Logic: A Very Simple Logic, Propositional Theorem Proving.

LOGIC: PROPOSITIONAL LOGIC

- Propositional logic (PL) is the simplest form of logic where all the statements are made by propositions.
- A proposition is a declarative statement which is either true or false.
- It is a technique of knowledge representation in logical and mathematical form.

Example:

a) It is Sunday.

b) The Sun rises from West (False proposition)

c) 3+3= 7(False proposition)

d) 5 is a prime number.

SOME BASIC FACTS ABOUT PROPOSITIONAL LOGIC

- Propositional logic is also called Boolean logic as it works on 0 and 1.
- In propositional logic, we use symbolic variables to represent the logic, and we can use any symbol for a representing a proposition, such A, B, C, P, Q, R, etc.
- Propositions can be either true or false, but it cannot be both.
- Propositional logic consists of an object, relations or function, and logical connectives.
- These connectives are also called logical operators.
- The propositions and connectives are the basic elements of the propositional logic.
- Connectives can be said as a logical operator which connects two sentences.
- A proposition formula which is always true is called tautology, and it is also called a valid sentence.
- A proposition formula which is always false is called Contradiction.
- A proposition formula which has both true and false values is called
- Statements which are questions, commands, or opinions are not propositions such as "Where is Rohini", "How are you", "What is your name", are not propositions.

SYNTAX OF PROPOSITIONAL LOGIC

GITAM (DEEMED TO BE UNIVERSITY)
VISAKHAPATNAM • HYDERABAD • BENGALURU

- The syntax of propositional logic defines the allowable sentences for the knowledge representation. There are two types of Propositions:
  - Atomic Proposition
  - Compound propositions
- Atomic Proposition: Atomic propositions are the simple propositions. It consists of a single proposition symbol. These are the sentences which must be either true or false.

Example:

> a) 2+2 is 4, it is an atomic proposition as it is a **true** fact.
>
> b) "The Sun is cold" is also a proposition as it is a **false** fact.

- Compound proposition: Compound propositions are constructed by combining simpler or atomic propositions, using parenthesis and logical connectives.

Example:

> a) "It is raining today, and street is wet."
>
> b) "Ankit is a doctor, and his clinic is in Mumbai."

PROPOSITIONAL LOGIC CONNECTIVES

Connective symbols	Word	Technical term	Example
$\wedge$	AND	Conjunction	$A \wedge B$
$\vee$	OR	Disjunction	$A \vee B$
$\rightarrow$	Implies	Implication	$A \rightarrow B$
$\Leftrightarrow$	If and only if	Biconditional	$A \Leftrightarrow B$
$\neg$ or $\sim$	Not	Negation	$\neg A$ or $\neg B$

Contd..

1. **Negation:** A sentence such as ¬ P is called negation of P. A literal can be either Positive literal or negative literal.

2. **Conjunction:** A sentence which has ∧ connective such as, **P ∧ Q** is called a conjunction.
   Example: Rohan is intelligent and hardworking. It can be written as,
   P= Rohan is intelligent,
   Q= Rohan is hardworking. → P∧ Q.

3. **Disjunction:** A sentence which has ∨ connective, such as **P ∨ Q**. is called disjunction, where P and Q are the propositions.
   Example: "Ritika is a doctor or Engineer",
   Here P= Ritika is Doctor. Q= Ritika is Doctor, so we can write it as **P ∨ Q**.

4. **Implication:** A sentence such as P → Q, is called an implication. Implications are also known as if-then rules. It can be represented as
   If it is raining, then the street is wet.
   Let P= It is raining, and Q= Street is wet, so it is represented as P → Q

5. **Biconditional:** A sentence such as **P⇔ Q is a Biconditional sentence, example If I am breathing, then I am alive**
   P= I am breathing, Q= I am alive, it can be represented as P ⇔ Q.

PROPERTIES OF OPERATORS

- **Commutativity:**
  - $P \wedge Q = Q \wedge P$, or
  - $P \vee Q = Q \vee P$.
- **Associativity:**
  - $(P \wedge Q) \wedge R = P \wedge (Q \wedge R)$,
  - $(P \vee Q) \vee R = P \vee (Q \vee R)$
- **Identity element:**
  - $P \wedge True = P$,
  - $P \vee True = True$.
- **Distributive:**
  - $P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$.
  - $P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$.
- **DE Morgan's Law:**
  - $\neg (P \wedge Q) = (\neg P) \vee (\neg Q)$
  - $\neg (P \vee Q) = (\neg P) \wedge (\neg Q)$.
- **Double-negation elimination:**
  - $\neg (\neg P) = P$.

LIMITATIONS OF PROPOSITIONAL LOGIC

- We cannot represent relations like ALL, some, or none with propositional logic. Example:
  - All the girls are intelligent
  - Some apples are sweet.
- Propositional logic has limited expressive power.
- In propositional logic, we cannot describe statements in terms of their properties or logical relationships.

References:

Text Book(s)

1. Stuart Russell, Peter Norvig, Artificial Intelligence: A Modern Approach, 3rd Edition, Pearson Publications, 2020

2. Dr. Nilakshi Jain, Artificial Intelligence : Making a System Intelligent, Wiley Publications,1st Edition,2019

Reference Book(s)

1. Elaine Rich, Kevin Knight and Shivashankar B. Nair, Artificial Intelligence, TMH Education Pvt. Ltd., 2008.

2. Dan W. Patterson, Introduction to Artificial Intelligence and Expert Systems, Pearson.

Internet:

https://www.javatpoint.com/artificial-intelligence-tutorial

https://www.geeksforgeeks.org/types-of-environments-in-ai/

https://www.tutorialspoint.com/artificial_intelligence/

etc