CS249
Artificial Intelligence

Week 9

2201AI43 - Venkata Haswanthi
2201AI44 - Vinay Sadupati
2201AI45 - Yash Kamdar
2201AI46 - Yashwant Saroj
2201AI47 - Ankit Singh

Knowledge Representation And Reasoning

It is part of AI that is concerned with AI agents' thinking and how thinking contributes to intelligent behaviour of agents.

Some important criteria of intelligence : -

- Understanding reasoning and interpreting knowledge
- Perform real world actions based on knowledge

How machines do all these things ??

These require knowledge representation and reasoning

What is knowledge ??

Representation of real world information -

- That can be understand
- That can be utilized to solve the real world complex problems

What is knowledge representation ??

- Knowledge representation is not just storing the data into some database.
- It includes enabling that machine to learn from that knowledge.
- Making machines behave intelligently like humans

How to represent the knowledge ??

Following are needed represent knowledge in AI systems:

- **Object:** All the facts about objects in our world domain. E.g., Guitars contains strings, trumpets are brass instruments.
- **Events:** Events are the actions which occur in our world.
- **Performance:** It describe behavior which involves knowledge about how to do things.
- **Meta-knowledge:** It is knowledge about what we know.
- **Facts:** Facts are the truths about the real world and what we represent.
- **Knowledge-Base:** The central component of the knowledge-based agents is the knowledge base. It is represented as KB. The Knowledgebase is a group of the Sentences (Here, sentences are used as a technical term and not identical with the English language).

Reasoning In AI :

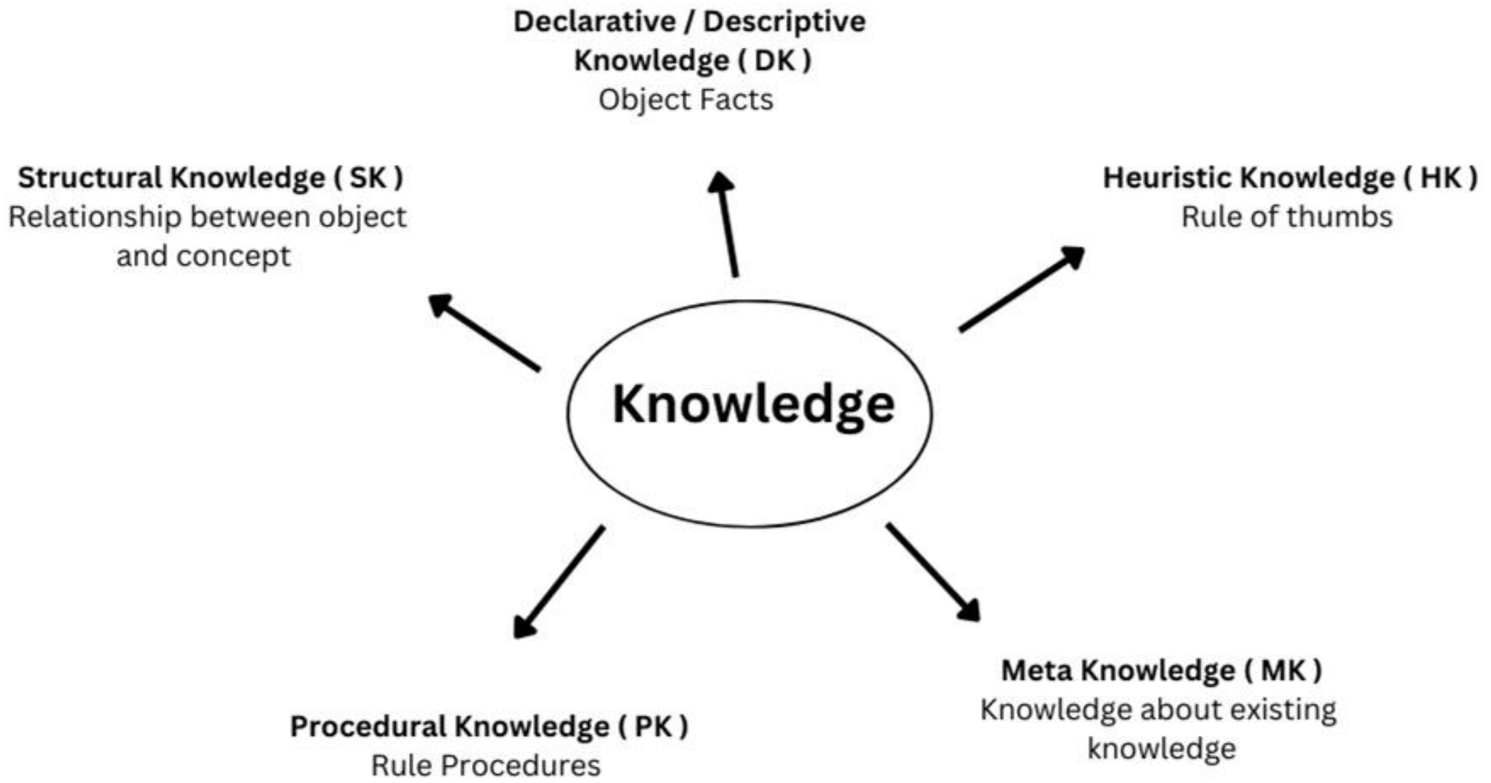
The reasoning is the mental process of deriving logical conclusion and making predictions from available knowledge, facts, and beliefs.

"**Reasoning is a way to infer facts from existing data**." It is a general process of thinking rationally, to find valid conclusions.

Reasoning in Artificial Intelligence refers to the process by which AI systems analyze information, make inferences, and draw conclusions to solve problems or make decisions. It is a fundamental cognitive function that enables machines to mimic human thought processes and exhibit intelligent behavior.

In artificial intelligence, reasoning can be divided into the following categories:

- Deductive reasoning
- Inductive reasoning
- Abductive reasoning
- Common Sense Reasoning
- Monotonic Reasoning
- Non-monotonic Reasoning


Declarative / Descriptive
Knowledge (DK)
Object Facts
Structural Knowledge (SK)
Relationship between object
and concept
Heuristic Knowledge (HK)
Rule of thumbs
Knowledge
Procedural Knowledge (PK)
Rule Procedures
Meta Knowledge (MK)
Knowledge about existing
knowledge

1. Declarative Knowledge:

- Declarative knowledge is to know about something.
- It includes concepts, facts, and objects.
- It is also called descriptive knowledge and expressed in declarative sentences.
- It is simpler than procedural language.

2. Procedural Knowledge

- It is also known as imperative knowledge.
- Procedural knowledge is a type of knowledge which is responsible for knowing how to do something.
- It can be directly applied to any task.
- It includes rules, strategies, procedures, agendas, etc.
- Procedural knowledge depends on the task on which it can be applied.

3. Meta-knowledge:

- Knowledge about the other types of knowledge is called Meta-knowledge.

4. Heuristic knowledge:

- Heuristic knowledge is representing knowledge of some experts in a file or subject.
- Heuristic knowledge is rules of thumb based on previous experiences, awareness of approaches, and which are good to work but not guaranteed.

5. Structural knowledge:

- Structural knowledge is basic knowledge to problem-solving.
- It describes relationships between various concepts such as kind of, part of, and grouping of something.
- It describes the relationship that exists between concepts or objects.

Logical Knowledge Representation

Logical Representation is a fundamental method of communicating knowledge to machines through a well-defined syntax with precise rules. This syntax should be unambiguous and able to handle prepositions, making it an ideal way to represent facts.

Logical representation consists of precisely defined syntax and semantics supporting the sound inference. Some concrete rules deal with propositions and draws a conclusion based on various conditions.

There are two types of logical representation:

- Propositional Logic
- First-order Logic

Syntax -

- These are the rules which decide how can we construct legal sentences in logic.
- It determines which symbols can be used in knowledge representation.
- Syntax determines how to write these symbols.
- The syntax of propositional logic defines the allowable sentences. The atomic sentences consist of a single proposition symbol.
- Each such symbol stands for a proposition that can PROPOSITION SYMBOL be true or false. We use symbols that start with an uppercase letter and may contain other letters or subscripts, for example: P, Q, R, W1,3 and North.
- The names are arbitrary but are often chosen to have some mnemonic value—we use W1,3 to stand for the proposition that the wumpus is in [1,3]. (Remember that symbols such as W1,3 are atomic, i.e., W, 1, and 3 are not meaningful parts of the symbol.)
- There are two proposition symbols with fixed meanings: True is the always-true proposition and False is the always-false proposition

Semantics -

- Having specified the syntax of propositional logic, we now specify its semantics.
- Semantics are the rules by which we can interpret the sentence in the logic.
- Semantic also involves assigning a meaning to each sentence
- The semantics defines the rules for determining the truth of a sentence with respect to a particular TRUTH VALUE model.
- In propositional logic, a model simply fixes the truth value—true or false—for every proposition symbol.

Advantages of logical representation:

- Logical representation enables us to do logical reasoning.
- Logical representation is the basis for the programming languages.

Disadvantages of logical Representation:

- Logical representations have some restrictions and are challenging to work with.
- Logical representation technique may not be very natural, and inference may not be so efficient.

Propositional Logic (PL)

Proposition is a declarative statement which is either TRUE (T) or FALSE (F).

- Propositional logic is a technique of knowledge representation in logical and mathematical form.
- It is the simplest form of logic
- All the statements are made by propositions.

Propositional Logic:-

1. Is also called Boolean logic as it works on 0 and 1.
2. Can either be True or False, But can’t be both.
3. Consists of an Object, Relations/FUnctions and Logical Connectives.
    a. Connectives are also called Logical Operators.
4. The propositions and connectives are the basic elements of the propositional logic

Logical Connectives

Used to connect 2 simple propositions

Let A and B be two propositions:-

SYMBOL	NAME	TECHNICAL TERM	SYNTAX
$\neg$	NOT	Negation	$\neg A$
$\vee$	OR	Disjunction	$A \vee B$
$\wedge$	AND	Conjunction	$A \wedge B$
$\rightarrow$	IMPLIES	Implication	$A \rightarrow B$
$\leftrightarrow$	IF AND ONLY IF	Biconditional	$A \leftrightarrow B$

Precedence of Logic

1. ( )
2. $\neg$
3. $\vee$
4. $\wedge$
5. $\rightarrow$
6. $\leftrightarrow$

Logical Equivalence

$$A \rightarrow B \equiv \neg A \vee B$$

$$A \leftrightarrow B \equiv (\neg A \vee B) \wedge (\neg B \vee A)$$

Truth Table

A	B	$\neg$A	A $\vee$ B	A $\wedge$ B	A $\rightarrow$ B	A $\leftrightarrow$ B
T	T	F	T	T	T	T
T	F	F	T	F	F	F
F	T	T	T	F	T	F
F	F	T	F	F	T	T

Types of Propositions:-

- Atomic: Consists of single propositional symbol.
- Compound: Combining atomic propositions using () and logical connectives

Propositional theorems:-

$$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha) \quad \text{commutativity of } \wedge$$
$$(\alpha \vee \beta) \equiv (\beta \vee \alpha) \quad \text{commutativity of } \vee$$
$$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma)) \quad \text{associativity of } \wedge$$
$$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma)) \quad \text{associativity of } \vee$$
$$\neg(\neg\alpha) \equiv \alpha \quad \text{double-negation elimination}$$
$$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha) \quad \text{contraposition}$$
$$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta) \quad \text{implication elimination}$$
$$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination}$$
$$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta) \quad \text{De Morgan}$$
$$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta) \quad \text{De Morgan}$$
$$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) \quad \text{distributivity of } \wedge \text{ over } \vee$$
$$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) \quad \text{distributivity of } \vee \text{ over } \wedge$$

Properties of Boolean Formula

- Valid/Tautology: A propositional formula which is always true
- Contradiction/Unsatisfiable: A propositional formula which is always false
- Satisfiable: A propositional formula which is true for at least one assignment of its variables

Inference

- Generating the conclusions from evidence and facts.
- The template for generating valid arguments
- Applied to derive proofs in AI
  - The proof is a sequence of conclusion that leads to the desired goal.

Deduction using proper logic:

Some terminologies related to inference rules:

- Implications: $A \rightarrow B$
- Converse: $B \rightarrow A$
- Inverse: $\neg A \rightarrow \neg B$
- Contrapositive: $\neg B \rightarrow \neg A$

NOTE: Implications and its contrapositives are equivalent

$(A \rightarrow B) \leftrightarrow (\neg B \rightarrow \neg A)$ is Valid/Tautology.(Always True)

Rules Of Inference

Modus Ponens:- If A and A → B are true we can infer that B is True.

$$\frac{\alpha \Rightarrow \beta, \qquad \alpha}{\beta}$$

Modus Tollens:- If B is false and A → B are true we can infer that A is also false

$$\frac{p \Rightarrow q, \neg q}{\therefore \neg p}$$

Addition:- If P is true then (P or Q) wil always be true

$$\frac{P}{\therefore P \vee Q}$$

Resolution:- If (p or q) is true and (NOT p or r) is also true then: q or r will always be true

$$\begin{array}{r} p \vee q \\ \neg p \vee r \\ \hline \therefore q \vee r \end{array}$$

Hypothetical Syllogism

ilemma

$$\frac{P \to Q, Q \to R}{\therefore P \to R}$$

$$\frac{(P \to Q), (R \to S), P \lor R}{\therefore Q \lor S}$$

Disju ogism

$$\frac{P \lor Q, \neg P}{\therefore Q}$$

Destructiv

$$\frac{P \to Q, R \to S, \neg Q \lor \neg S}{\therefore \neg P \lor \neg R}$$

$$\frac{P \land Q}{\therefore P}$$

cation

Conjunction

$$\frac{P, Q}{\therefore P \land Q}$$

Conjunctive Normal Form

- The resolution rule applies only to clauses (that is, disjunctions of literals), so it would seem to be relevant only to knowledge bases and queries consisting of clauses.
- How, then, can it lead to a complete inference procedure for all of propositional logic? The answer is that every sentence of propositional logic is logically equivalent to a conjunction of clauses.
- A sentence expressed as a conjunction of clauses is said to be in conjunctive normal form or CNF.We now describe a procedure for converting to CNF.
- We illustrate the procedure by converting the sentence $B1,1 \Leftrightarrow (P1,2 \vee P2,1)$ into CNF. The steps are as follows:

1. Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.

$$(B1,1 \Rightarrow (P1,2 \vee P2,1)) \wedge ((P1,2 \vee P2,1) \Rightarrow B1,1) .$$

2. Eliminate ⇒, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \lor \beta$:

$$(\neg B1,1 \lor P1,2 \lor P2,1) \land (\neg(P1,2 \lor P2,1) \lor B1,1) .$$

3. CNF requires ¬ to appear only in literals, so we "move ¬ inwards" by repeated application of the following equivalences :

$$\neg(\neg\alpha) \equiv \alpha \text{ (double-negation elimination)}$$

$$\neg(\alpha \land \beta) \equiv (\neg\alpha \lor \neg\beta) \text{ (De Morgan)}$$

$$\neg(\alpha \lor \beta) \equiv (\neg\alpha \land \neg\beta) \text{ (De Morgan)}$$

In the example, we require just one application of the last rule:

$$(\neg B1,1 \lor P1,2 \lor P2,1) \land ((\neg P1,2 \land \neg P2,1) \lor B1,1) .$$

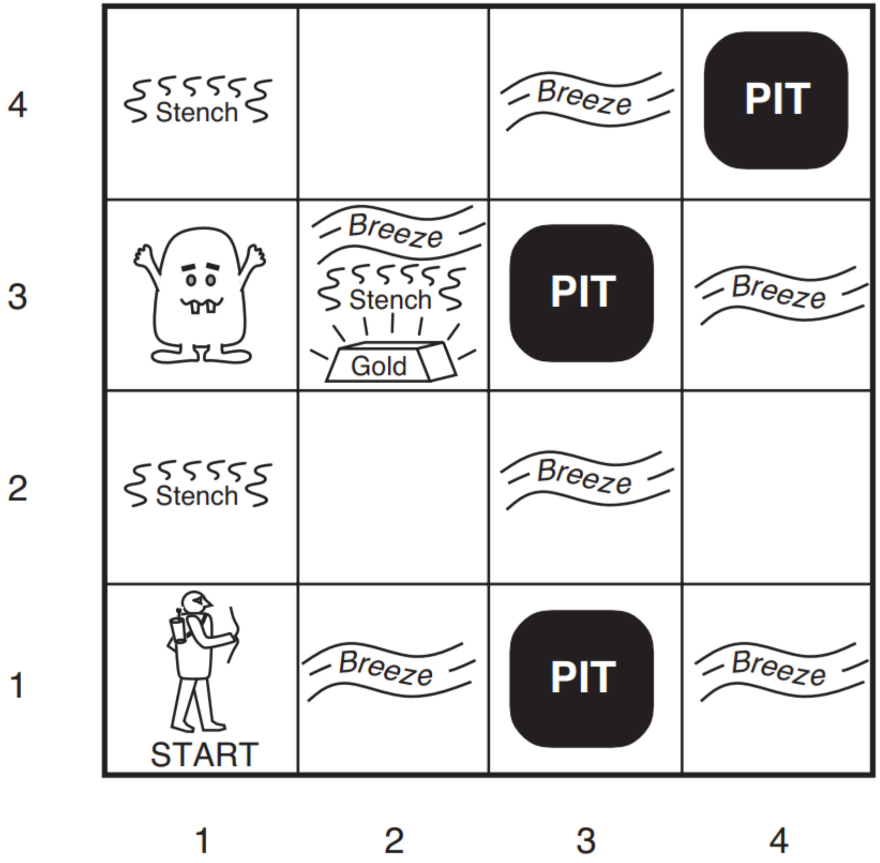
4. Now we have a sentence containing nested ∧ and ∨ operators applied to literals. We apply the distributive law ,distributing ∨ over ∧ wherever possible.

$$(\neg B1,1 \lor P1,2 \lor P2,1) \land (\neg P1,2 \lor B1,1) \land (\neg P2,1 \lor B1,1) .$$

The original sentence is now in CNF, as a conjunction of three clauses. It is much harder to read, but it can be used as input to a resolution procedure.

Wumpus World

"Wumpus World" is a classic text-based adventure game that involves exploring a cave system filled with hazards, most notably the wumpus, a dangerous creature lurking within.



Basic rules of the game:

Objective:

- The objective of the game is to explore the cave system and collect all the treasures while avoiding hazards like bottomless pits and the wumpus.

Cave Layout:

- The cave system consists of interconnected rooms, each represented by a grid or a set of connected nodes. The layout is initially hidden from the player, and they must navigate through it to uncover its features.

Hazards:

- Wumpus: The wumpus is a dangerous creature that resides in one of the rooms. If the player enters the room with the wumpus, they are eaten and the game ends.
- Pits: Some rooms contain bottomless pits. If the player falls into a pit, they are trapped and the game ends.

- Arrows: The player has a limited number of arrows that they can use to kill the wumpus. However, the arrows are finite, so the player must use them wisely.

Percept:

Before making a move, the player receives sensory information about the surrounding rooms. This information includes:

- Stench: A foul smell indicates the presence of the wumpus in an adjacent room.
- Breeze: A gentle breeze indicates the presence of a bottomless pit in an adjacent room.
- Glitter: A sparkling indicates the presence of treasure in the current room.

Actions:

- Move: The player can move to an adjacent room.
- Shoot: The player can shoot an arrow into an adjacent room to attempt to kill the wumpus.
- Grab: The player can pick up treasure if it's present in the current room.

Consequences:

- If the player moves into a room with the wumpus, they are eaten, and the game ends.
- If the player falls into a room with a bottomless pit, they are trapped, and the game ends.
- If the player runs out of arrows without killing the wumpus, they may still move and explore, but they can no longer shoot.
- If the player collects all the treasures without encountering hazards, they win the game.

Feedback:

- After each action, the game provides feedback to the player based on the outcome of their actions, such as whether they killed the wumpus, found treasure, or fell into a pit.

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.

Actuators:

- The agent can move Forward, TurnLeft by 90◦, or TurnRight by 90◦.
- The agent dies a miserable death if it enters a square containing a pit or a live wumpus. (It is safe, albeit smelly, to enter a square with a dead wumpus.) If an agent tries to move forward and bumps into a wall, then the agent does not move.
- The action Grab can be used to pick up the gold if it is in the same square as the agent.
- The action Shoot can be used to fire an arrow in a straight line in the direction the agent is facing. The arrow continues until it either hits (and hence kills) the wumpus or hits a wall. The agent has only one arrow, so only the first Shoot action has any effect.
- Finally, the action Climb can be used to climb out of the cave, but only from square [1,1].

Sensors:-

The agent has five sensors, each of which gives a single bit of information: –

- In the square containing the wumpus and in the directly (not diagonally) adjacent squares, the agent will perceive a Stench.
- In the squares directly adjacent to a pit, the agent will perceive a Breeze.
- In the square where the gold is, the agent will perceive a Glitter.
- When an agent walks into a wall, it will perceive a Bump.
- When the wumpus is killed, it emits a woeful Scream that can be perceived anywhere in the cave.

The percepts will be given to the agent program in the form of a list of five symbols; for example, if there is a stench and a breeze, but no glitter, bump, or scream, the agent program will get [Stench, Breeze, None, None, None].

Model the problem

Atomic proposition variable for Wumpus world:

- Let $\mathbf{P}_{i,j}$ be true if there is a Pit in the room [i, j].
- Let $\mathbf{B}_{i,j}$ be true if agent perceives breeze in [i, j], (dead or alive).
- Let $\mathbf{W}_{i,j}$ be true if there is wumpus in the square[i, j].
- Let $\mathbf{S}_{i,j}$ be true if agent perceives stench in the square [i, j].
- Let $\mathbf{V}_{i,j}$ be true if that square[i, j] is visited.
- Let $\mathbf{G}_{i,j}$ be true if there is gold (and glitter) in the square [i, j].

Let us watch a knowledge-based wumpus agent exploring the environment.

The agent's initial knowledge base contains the rules of the environment, as described previously; in particular, it knows that it is in [1,1] and that [1,1] is a safe square; we denote that with an "A" and "OK," respectively, in square [1,1].

The first percept is [None, None, None, None, None], from which the agent can conclude that its neighboring squares, [1,2] and [2,1], are free of dangers—they are OK.

A cautious agent will move only into a square that it knows to be OK.

Let us suppose the agent decides to move forward to [2,1]. The agent perceives a breeze (denoted by "B") in [2,1], so there must be a pit in a neighboring square. The pit cannot be in [1,1], by the rules of the game, so there must be a pit in [2,2] or [3,1] or both.

At this point, there is only one known square that is OK and that has not yet been visited. So the prudent agent will turn around, go back to [1,1], and then proceed to [1,2].

The agent perceives a stench in [1,2], resulting in the state of knowledge shown in Figure 7.4(a). The stench in [1,2] means that there must be a wumpus nearby. But the

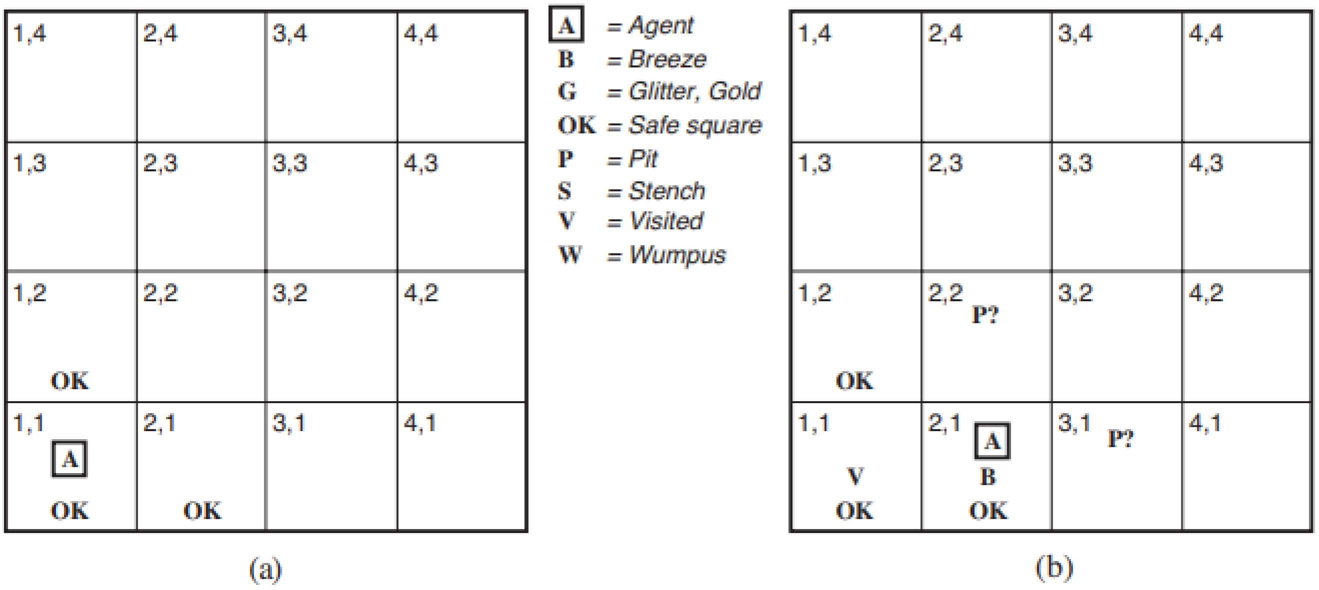


Figure 7.3 The first step taken by the agent in the wumpus world. (a) The initial situation, after percept $[None, None, None, None, None]$. (b) After one move, with percept $[None, Breeze, None, None, None]$.

But the wumpus cannot be in [1,1], by the rules of the game, and it cannot be in [2,2] (or the agent would have detected a stench when it was in [2,1]).

Therefore, the agent can infer that the wumpus is in [1,3]. The notation W! indicates this inference.

Moreover, the lack of a breeze in [1,2] implies that there is no pit in [2,2]. Yet the agent has already inferred that there must be a pit in either [2,2] or [3,1], so this means it must be in [3,1].

This is a fairly difficult inference, because it combines knowledge gained at different times in different places and relies on the lack of a percept to make one crucial step.

The agent has now proved to itself that there is neither a pit nor a wumpus in [2,2], so it is OK to move there. We do not show the agent's state of knowledge at [2,2]; we just assume that the agent turns and moves to [2,3]

In [2,3], the agent detects a glitter, so it should grab the gold and then return home.

Note that in each case for which the agent draws a conclusion from the available information, that conclusion is guaranteed to be correct if the available information is correct.

This is a fundamental property of logical reasoning.

In the rest of this chapter, we describe how to build logical agents that can represent information and draw conclusions such as those described in the preceding paragraphs.

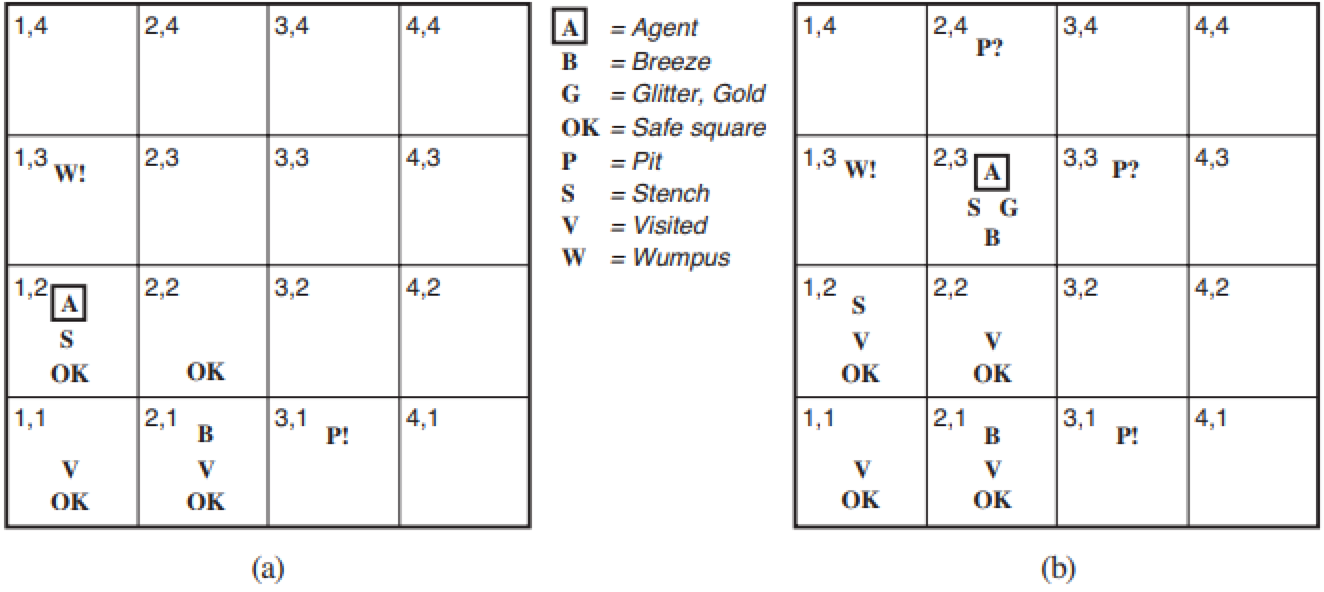


Figure 7.4 Two later stages in the progress of the agent. (a) After the third move, with percept [*Stench*, *None*, *None*, *None*, *None*]. (b) After the fifth move, with percept [*Stench*, *Breeze*, *Glitter*, *None*, *None*].

CITATIONS

- https://people.eecs.berkeley.edu/~russell/slides/
- https://www.cs.utexas.edu/~mooney/cs343/slide-handouts
- https://www.geeksforgeeks.org/proofs-and-inferences-in-proving-propositional-theorem/
- Artificial Intelligence : A Modern Approach by Stuart Russell and Peter Norvig

THANK YOU

2201AI43

2201AI44

2201AI45

2201AI46

2201AI47