6. AGENTS THAT REASON LOGICALLY (LOGICAL AGENTS)

6. AGENTS THAT REASON LOGICALLY (LOGICAL AGENTS)

6.1 INTRODUCTION

The *knowledge based agent* (constructed using agent program) has to perform the following task using logic representation. The tasks are:

(i) to know the current state of the world

(ii) how to infer the unseen properties of the world

(iii) new changes in the environment

(iv) goal of the agent

(v) how to perform actions depends on circumstances.

Logic : A formal language to represent the knowledge in which reasoning is carried out to achieve the goal state

6.2 A KNOWLEDGE–BASED AGENT

The component of a knowledge based agent is its *Knowledge Base or KB*. Each individual representation is called a *sentence*. The sentences are expressed in a language called a *knowledge representation language*. How to add new sentences to the existing knowledge base and how to derive the answer for the given percept (inference)? To perform these operations two standard tasks are used (i.e) TELL and ASK.

(i) TELL - to add new sentences to the knowledge base

(ii) ASK - to query what is known for the given percept.

```
function KB–AGENT(percept) returns an action
    static: 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
```

Figure 6.1 A generic knowledge-based agent

MAKE–PERCEPT–SENTENCE**:** takes the percept and returns a sentence representing the fact about the world.

MAKE–ACTION-QUERY**:** takes the time as input, and returns a sentence.

Three levels of knowledge based agent

(i) *The knowledge level or epistemological level* : An external level which describes the agent by saying what it knows as a formal languages representation. If TELL and ASK work correctly then most of the time we can work at the knowledge level and not worry about lower levels.

Example: The Bridge links Adyar and Thiruvanmiyur.

(ii) *The logical level* – is the level at which the knowledge is encoded into logical sentences.

Example : Links (Bridge, Adyar, Thiruvanmiyur)

(iii) *The implementation level*-is the level that runs on the agent architecture. The logical sentences are represented as a physical representation in the KB (i.e) the string “Links (Bridge, Adyar, Thiruvanmiyur)”

The three dimensioned table indexed by Bridge name and location pairs.

6.3 THE WUMPUS WORLD ENVIRONMENT

Problem statement : Wumpus was an early game, based on an agent who explores a cave consisting of rooms connected by passage ways. Lurking somewhere in the cave is the wumpus, a beast that eats anyone who enters its room. To make matters worse, some rooms contain bottomless pits that will trap anyone who wanders into these rooms (except for the wumpus, who is too big to fall in). The only mitigating feature of living in this environment is the occasional heap of gold.

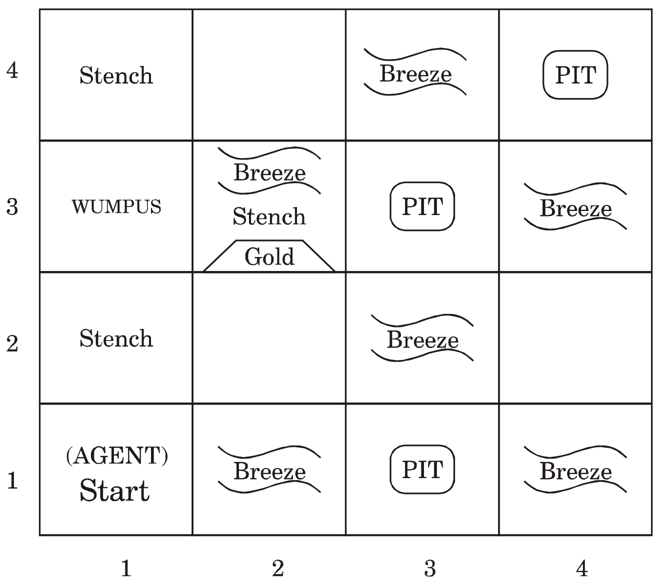

Figure 6.2 Wumpus world environment

Given

(i) Wumpus world is a grid of squares surrounded by walls.

(ii) Each square can contain agent and objects.

(iii) In the square containing the *wumpus* and in the adjacent squares the agent will perceive a *stench*.

(iv) In the squares adjacent to a *pit*, the agent will perceive a *breeze*.

(v) In the square where the *gold* is, the agent will perceive a *glitter*.

(vi) When an agent walks into a *wall*, it will perceive a *bump*.

(vii) When the wumpus is *killed*, it will give a *scream*, perceived anywhere in the cave.

(viii) To perceive the location by the agent the percept is given in the following order of [*Stench, Breeze, Glitter, Bump, Scream*]

Operators

(i) Move *forward*

(ii) Turn *left* by 90^0

(iii) Turn *right* by 90^0

(iv) *Grab* – pick up an object

(v) *Shoot* – fire an arrow to kill the wumpus, to be used only once.

(vi) *Climb* – to leave the cave, from the start square.

Initial state

Agent in the location of square (1,1).

Goal state

To find the gold and bring it out of the cave as quickly as possible, without getting killed.

Solution

(i)

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 $\boxed{A}$ *OK*	*2,1* *OK*	*3,1*	*4,1*

$\boxed{A}$ - *Agent*

B - *Breeze*

G - *Glitter, Gold*

OK - *Safe square*

P - *Pit*

S - *Stench*

V - *Visited*

W - *Wumpus*

Figure 6.3 The first step taken by the agent in the wumpus world. The initial situation, after percept [*None, None, None, None, None*]

(ii)

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 $\boxed{A}$ B OK	3,1 P?	4,1

Figure 6.4 After one move with percept [*None, Breeze, None, None, None*].

The agent perceives, Breeze in square (2,1). Therefore the pit may be exist in any one of the adjacent squares (3,1) (or) (2,2) (as per the given rule). Now the agent backtracks to the previous square (1,1) and selects a move towards the other direction (1,2).

(iii)

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 $\boxed{A}$ S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

Figure 6.5 After the third move with percept [*Stench, None, None, None, None*]

The agent perceives a stench in the square (1,2). Therefore the wumpus may exist in any one of the adjacent squares (1,3) (or) (2,2). Now the agent identifies the next safe square (2,2) with the following two conditions.

(i) If the wumpus is exist in (2,2) then stench should exist in (2,1) also. Therefore the Wumpus is exist only in the square (1,3).

(ii) If the pit is exist in (2,2) then the breeze should be perceived by the agent in (1,2) also. Therefore the pit is exist only in the square (3,1)

(iv)

1,4	2,4	3,4	4,4
1,3 W!	2,3 OK	3,3	4,3
1,2 S V OK	2,2 $\boxed{A}$ OK	3,2 OK	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

Figure 6.6 After the fourth move with percept [*None, None, None, None, None*]

(v)

1,4	2,4	3,4	4,4
1,3 W!	2,3 $\boxed{A}$ S G B	3,3	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

Figure 6.7 After the fifth move with percept [*Stench, Breeze, Glitter, None, None*]

The square (2,3) has the percept – Glitter, indicates the gold is exist in the square. Now the agent uses the following operators in sequence to come out of the cave with gold.

(i) Grab - pick up an object (Gold)

(ii) Climb - to leave the cave from the start square.

6.4 REPRESENTATION, REASONING AND LOGIC

The main aim of knowledge representation is to express knowledge in computer tractable form, to help agents to improve their performance. A knowledge representation language is defined by two aspects :

(i) The *syntax* of a language describes how the sentences are represented inside the computer. For now, think of this as being a physical pattern of electrons in the computer's memory.

(ii) The *semantics* determines the facts in the world (i.e) when a particular configuration exists within an agent, the agent believes the world.

Example : x=y, a valid statement – *syntax*.

If the value of x and y matches then true else false – *semantics*.

Logic: A knowledge representation language in which syntax and semantics are defined correctly is known as *logic*.

Entailment: To generate new sentences that are necessarily true, given that the old sentences are true. This relation between sentences is called entailment and the relation of entailment between a knowledge base KB and a sentence $\propto$ is written as KB $\models \alpha$ (i.e) "KB entails α".

The connection between sentences (Representation) and facts (World) is provided by the semantics of the language. The property of one fact following from some other fact is mirrored by the property of one sentence being entailed by some other sentence is shown below in the Figure 6.8.

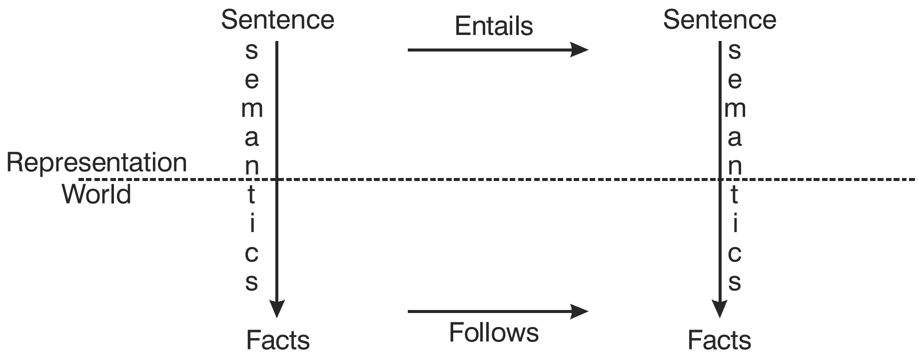

Figure 6.8 The connection between sentences and fact

An inference procedure can do one of two things :

(i) From a knowledge base KB, the new sentence can be entailed (i.e $\propto$)

(ii) Given a knowledge base KB and another sentence $\propto$, inference procedure can report whether or not α entailed by KB.

An inference procedure that generates only entailed sentences is called *sound* or *truth preserving*.

An inference procedure (i) which derives $\propto$ from KB is represented as: $KB \vdash_i \alpha$ (Alpha is derived from KB by i (or) i derives alpha from KB)

The record of operation of a sound inference procedure is called a *proof.*

An inference procedure is complete if it can find a proof for any sentence that is entailed.

Semantics : In logic, semantics represents the meaning of a sentence is what it states about the world.

Inference (or) Reasoning: From the set of given facts a conclusion is derived using logical *inference* or *deduction*.

Validity : A sentence is valid or necessarily true if it is true under all possible interpretations in all possible worlds (i.e) universally described.

Example : There is a stench at [1,1] or there is not a stench at [1, 1]-wumpus world problem.

The above example is always true, because, any one of the constraint is necessarily true in the wumpus world environment.

Satisfiability: A sentence is satisfiable if and only if there is some interpretation in some world for which it is true.

Example : There is a wumpus at [1,2] – wumpus world problem.

The above sentence is satisfiable in the wumpus world environment.

Example : There is a wall in front of me and there is no wall in front of me-wumpus world problem.

The above sentence is unsatisfiable, since it is *self contradictory.*

6.5 LOGICS - AN INTRODUCTION

Logics consist of the following two representations in sequence.

(i) A formal system - to describe the state of the world

- * Syntax - which describes how to make sentences.
- * Semantics - which describes the meaning of the sentence.

(ii) The proof theory - a set of rules for deducing the entailments of a set of sentences.

We will represent the sentences using two different logics. They are:

1. Propositional logic (or) Boolean logic.
2. Predicate logic (or) First order logic.

Propositional logic	Predicate logic
(i) Each fact is represented by one symbol.	(i) Representation of world in terms of objects and predicates on objects (i.e properties of objects or relations between objects).
(ii) Proposition symbols can be connected with boolean connectives, to give more complex meaning. Connectives: $\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$	(ii) Connectives and quantifiers are used to represent the meaning of the sentence. Connectives: $\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$ Quantifiers : $\forall, \exists$
(iii) Simple statements are implemented	(iii) Complex statements are implemented.
(iv) Example : Plato is a man *PLATOMAN*	(iv) Example : Plato is a man *MAN (PLATO)*

Figure 6.9 Propositional Vs predicate logic.

Note:

$\wedge$ - Conjunction (AND)
$\vee$ - Disjunction (OR)
$\neg$ (or) ~ - Negation (NOT)
$\Rightarrow$ - Implication
$\Leftrightarrow$ - Equivalence

How the logics can be represented using ontological commitment and epistemological commitment?

Ontological Commitment : What exists in the world
Epistemological Commitment : What an agent believes about facts.

Language	Ontological Commitment	Epistemological Commitment
Propositional logic	facts	true | false | unknown
First order logic	facts, objects, relations	true | false | unknown
Temporal logic	facts, objects, relations, times	true | false | unknown
Probability theory	facts	degree of belief 0.....1
Fuzzy logic	degree of truth	degree of belief 0.....1

Figure 6.10 Formal languages and their ontological and epistemological commitment

6.6 PROPOSITIONAL LOGIC

Symbols : The symbols of propositional logic are the logical constants (True and False).

Example : P, Q

Connectives

(i) $\wedge$(***and***) - Conjunction

Example : $P \wedge Q$

(ii) $\vee$(***or***) - Disjunction

Example : $P \vee Q$

(iii) $\rightarrow$ (***implies***) - Implication or Conditional

Example : $(P \wedge Q) \Rightarrow R$

$(P \wedge Q)$ - premise or antecedent

R - conclusion or consequent

(iv) $\Leftrightarrow$ (***equivalent***) - Equivalence or Biconditional

Example : $(P \wedge Q) \Leftrightarrow (Q \wedge P)$

(v) $\neg$ (***not***) - Negation is the only connective that operates on a single sentence.

Example : $\neg P$

A BNF grammer of sentence in propositional logic

Sentence $\rightarrow$ Atomic sentence | Complex sentence

Atomic Sentence $\rightarrow$ True | False | P | Q | R |

Complex sentence $\rightarrow$ (Sentence)

| Sentence Connective Sentence

| $\neg$ Sentence

Connective $\rightarrow$ $\wedge$ | $\vee$ | $\Leftrightarrow$ | $\Rightarrow$

Order of precedence (from highest to lowest) : $\neg$, $\wedge$, $\vee$, $\Rightarrow$ and $\Leftrightarrow$

Example : $\neg P \vee Q \wedge R \Rightarrow$ S is equivalent to $((\neg P) \vee (Q \wedge R)) \Rightarrow S$.

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
F	F	T	F	F	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
T	T	F	T	T	T	T

Figure 6.11 Truth table for the five logical connectives.

Validity

Truth tables can be used to define the validity of a sentence. If the sentence is true in every row (i.e for different types of logical constants) then the sentence is a valid sentence.

Example : $((P \vee H) \wedge \neg H) \Rightarrow P$ check whether the given sentence is a valid sentence or not.

P	H	$P \vee H$	$(P \vee H) \wedge \neg H$	$((P \vee H) \wedge \neg H) \Rightarrow P$
F	F	F	F	T
F	T	T	F	T
T	F	T	T	T
T	T	T	F	T

Figure 6.12 Truth table for $((P \vee H) \wedge \neg H) \Rightarrow P$

The given sentence is a valid sentence, because the sentence is true in every row for different types of logical constants.

Inference rules for propositional logic

α ■ β (or) $\frac{\alpha}{\beta}$ - This notation represents that β can be derived from α by inference. The propositional logic has seven inference rules. They are :

(i) Modus Ponens (or) Implication - Elimination

From an implication and the premise of the implication, conclusion is inferred.

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

(ii) And – Elimination

From a conjunction, any of the conjunct is inferred.

$$\frac{\alpha_1 \wedge \alpha_2 \wedge .... \wedge \alpha_n}{\alpha_1}$$

(iii) And – Introduction

From a list of sentences their conjunction is inferred.

$$\frac{\alpha_1, \alpha_2 ...... \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge ...... \wedge \alpha_n}$$

(iv) Or – Introduction

From a sentence, its disjunction with anything is inferred.

$$\frac{\alpha_i}{\alpha_1 \vee \alpha_2 \vee ...... \vee \alpha_n}$$

(v) Double – Negation Elimination

From a double negation elimination positive sentence is inferred

$$\frac{\neg\neg\alpha}{\alpha}$$

(vi) Unit Resolution

From a disjunction, if one of the disjuncts is false, then the true one is inferred.

$$\frac{\alpha \vee \beta, \neg\beta}{\alpha}$$

(vii) Resolution

Implication is transitive.

$$\frac{\alpha \vee \beta, \neg\beta \vee \gamma}{\alpha \vee \gamma} \, or \, \frac{\neg\alpha \Rightarrow \beta, \beta \Rightarrow \gamma}{\neg\alpha \Rightarrow \gamma}$$

6.7 AN AGENT FOR THE WUMPUS WORLD-PROPOSITIONAL LOGIC

In this topic we will discuss the knowledge base representation and a method to find the wumpus using propositional logic representation.

Given :

Assume that the agent has reached the square (1,2), which is shown in the following Figure 6.13.

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 $\boxed{A}$ S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

Figure 6.13 Wumpus world environment

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

The Knowledge Base : The agent percepts are converted into sentences and entered into the knowledge base, with some valid sentences that are entailed by the percept sentences.

From the above Figure 6.13, we can perceive the following percept sentences and it is added to the knowledge base.

$S_{1,2}$ - There is a Stench in (1,2)

$\neg B_{1,2}$ - There is no Breeze in (1,2)

$\neg S_{2,1}$ - There is no Stench in (2,1)

$B_{2,1}$ - There is a Breeze in (2,1)

$\neg S_{1,1}$ - There is no Stench in (1,1)

$\neg B_{1,1}$ - There is no Breeze in (1,1)

$\neg S_{1,1}$, $\neg B_{1,1}$ - for the square (1,1)

$\neg S_{2,1}$, $B_{2,1}$ - for the square (2,1)

$S_{1,2}$, $\neg B_{1,2}$ - for the square (1,2)

In addition, the agent should have some knowledge about the environment. For example, if there is no smell in a square, then neither the square nor any of its adjacent square can house a wumpus. Now we will frame the rule of three squares, are:

$$R_1 : \neg S_{1,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$$

$$R_2 : \neg S_{2,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{2,2} \wedge \neg W_{2,1} \wedge \neg W_{3,1}$$

$$R_3 : \neg S_{1,2} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,2} \wedge \neg W_{1,3}$$

If there is a stench in (1,2) then there must be a wumpus in (1,2) or in one or more of the neighboring squares.

$$R_4 : S_{1,2} \Rightarrow W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}$$

Finding the Wumpus :

1. Applying *Modus Ponens* to $\neg S_{1,1}$ and the sentence R_1, we obtain

$$\frac{\neg S_{1,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}, \neg S_{1,1}}{\neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}} \text{ (i.e)} \frac{\alpha \Rightarrow \beta, \alpha}{\beta}$$

(i.e) $\neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$ is equivalent to β.

2. Applying *And – Elimination* to this, we obtain the three separate sentences

$$\neg W_{1,1} \quad \neg W_{1,2} \quad \neg W_{2,1}$$

3. Applying *Modus Ponens* to $\neg S_{2,1}$ and the sentence R_2, we obtain

$$\neg W_{1,1} \wedge \neg W_{2,2} \wedge \neg W_{2,1} \wedge \neg W_{3,1}$$

4. Applying *And – Elimination* to this, we obtain four separate sentences

$$\neg W_{1,1} \quad \neg W_{2,2} \quad \neg W_{2,1} \quad \neg W_{3,1}$$

5. Applying *Modus Ponens* to $S_{1,2}$ and the sentence R_4, we obtain

$$W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}$$

6. Apply the *Unit Resolution* rule, where α is $W_{1,3} \vee W_{1,2} \vee W_{2,2}$ and β is $W_{1,1}$ (derived in step 5 is splitted into α and β) and the sentence $\neg W_{1,1}$ (from step 2), yields:

$$W_{1,3} \vee W_{1,2} \vee W_{2,2}$$

7. Apply the *Unit Resolution* rule, where α is $W_{1,3} \vee W_{1,2}$ and β is $W_{1,2}$ and β is $W_{2,2}$ (from step 6) and the sentence $\neg W_{2,2}$ (from step 3), yields:

$$W_{1,3} \vee W_{1,2}$$

8. Apply the *Unit Resolution* rule, where α is $W_{1,3}$ and β is $W_{1,2}$ (from step 7) and the sentence $\neg W_{1,2}$ (from step 2), yields :

$W_{1,3}$ = A square where the wumpus is exists.