THE JAINA LOGIC WAY OF WUMPUS WORLD

(1) R.Malathi (2) T.Venugopal
(1) Assistant Professor, (2) Professor of Mathematics, SCSVMV University,
malathilathar@gmail.com, drmathtvg@gmail.com,

Abstract: **In this paper, we focus our concentration upon the Jaina doctrine of syat. It is popularly known as "Syadvada", if the agent uses the Jaina logic it can give a best result.**

Key Words: **Syadvada, Truth table, Primitives, Wumpus world and Knowledge-Based Agent.**

I. INTRODUCTION

The true substance is jungle wood and another substance with reference to which the negative proposition is made is rose wood. This point is explained by Jaina thinkers in a technical way.

There can be no judgment absolutely true and excluding every other judgment about the same topic. Hence we have recourse to qualified assertions as the only available ones under the circumstances. These qualified or conditional assertions are primarily two affirmation and negation.

(1) Perhaps X is.
(2) Perhaps X is not.

These two aspects are inherent in the same thing; hence we can say

(3) Perhaps X is and is not; here we are contemplating the whole thing in its two aspects which are kept apart and attended to severally. But these two aspects are inherent in and expressive of one single identity. Hence, they may be considered together jointly as expressing the single identity. In that case, there is no chance of asserting two conjointly by a single predicate, for the simple reason that there can be no such predicate. Therefore we have to confess inability to describe and proclaim the bankruptcy of vocabulary for having such an assertion. This fact becomes the fourth mode of predication.
(4) Perhaps X is indescribable.

Remembering this helpless nature of our tongue, we may still qualify this by each of the first three predicates. Thus we have the last three modes of predications, which are:

(5) Perhaps X is though indescribable.
(6) Perhaps X is not though indescribable.
(7) Perhaps X is and is not though indescribable.

In their traditional form these are:

II. SYADVADA:

1. True
2. False
3. True and False
4. Indeterminant
5. Indeterminant and True
6. Indeterminant and False
7. Indeterminant, True and False

According to **Buddhism** everything is momentary and unreal. Both these views are rejected by Jainas as extremes. The former is true according to the principle of **Dravyarthika** point of view; the latter is true according to **Paryayarthika** point of view. Hence, each is true in its own way and is not true absolutely. Again reality is indescribable according to the **Vedantins** who emphasize the **Nirvachaniya** aspect of reality. Even this is only partially true, for otherwise, even this predication "That Reality is indescribable" will be impossible [1].

The same seven modes of predication may be obtained in the case of following pairs of attributes; eternal and changing, one and many, universal and particular, etc. these pairs of opposites can very well be predicated of reality and these may yield the other derivative modes of predication. Thus **practically, every attribute by being affirmed and denied according to different**

aspects may bring about seven fundamental propositions true of real subject.

1.1 JAINA LOGIC AGENT

This chapter introduces knowledge – based agents. The concepts that we discuss – the representation of knowledge and the reasoning processes that bring knowledge to life – are central to the entire field of artificial intelligence.

Humans, it seems, know things and do reasoning. Knowledge and reasoning are also important for artificial agents because they enable successful behaviors that would be very hard to achieve otherwise. (The knowledge of problem – solving agents is, however, very specific and inflexible. A chess program can calculate the legal moves of its king, but does not know in any useful sense that no piece can be on two different squares at the same time. Knowledge – based agents can benefit from knowledge expressed in very general forms, combining and recombining information to suit myriad purposes.)

Knowledge and reasoning also play a crucial role in dealing with partially observable environments. Knowledge – based agent can combine general knowledge with current percepts to infer hidden aspects of the current state prior to selecting actions. For example, a physician diagnoses a patient – that is, infers a disease state that is not directly observable – prior to choosing a treatment. Some of the knowledge that the physician uses in the form of rules learned from textbooks and teachers, and some is in the form of patterns of association that the physician may not be able to consciously describe. If its inside the physician's head, it counts as knowledge.

Understanding natural language also requires inferring hidden state, namely, the intention of the speaker. When we hear, "John saw the diamond through the window and coveted it," refers to the diamond and not the window – we reason, perhaps unconsciously, with our knowledge of relative value. Similarly, when we hear, "John threw the brick through the window and broke it," we know "it" refers to the window. Reasoning allows us to cope with the virtually infinite variety of utterances using a finite store of commonsense knowledge. Problem – solving agents have difficulty with this kind of ambiguity because their representation of contingency problem is inherently exponential.

Our final reason for studying knowledge – based agents is their flexibility. They are able to accept new tasks in the form of explicitly described goals, they can achieve competence quickly by being told or learning new knowledge about the environment, and they can adapt to changes in the environment by updating the relevant knowledge.

We are being in section 2.1 with the overall agent design. Section 2.2 introduces a simple new environment, the wumpus world, and illustrates the operation of a knowledge – based agent without going into any technical detail. The knowledge of logical agents is always definite – each proposition is true, false or indeterminant in the world, although the agent may be agnostic about some propositions.

Section 2.3 of this chapter defines a logic called "Jaina Logic". Jaina logic serves to illustrate all the basic concepts of logic.

2.1. KNOWLEDGE – BASED AGENTS

The central component of a knowledge – based agent is its knowledge base, or KB. Informally, a knowledge base is a set of sentences. (Here "sentence" is used as a technical term. It is related but is not identical to the sentences of English and other natural languages.) Each sentence is expressed in a language called a knowledge representation language and represents some assertion about the world.

There must be a way to add new sentences to the knowledge base and a way to query what is known the standard names for these tasks are TELL and ASK respectively. Both tasks may involve inference – that is, deriving new sentences from old. In logical agents, which are the main subject of study in this chapter, inference must obey the

```
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
```

Table : 2.1 A generic Knowledge – based agent.

Table: 2.1 show the outline of a knowledge – based agent program. Like all over agents, it takes a percept as input and returns an action. The agent maintains a knowledge base, KB, which may initially contain some background knowledge. Each time the agent program is called, it does two things. First, it TELLs the knowledge base what it perceives. Second, it ASKs the knowledge base what action it should perform. In the process of answering this query, extensive reasoning may be done about the outcomes of possible action sequences, and so on. Once the action is chosen, the agent records its choice with TELL and executes the action. The second TELL is necessary to let the knowledge base know that the hypothetical action has actually been executed.

One can build a knowledge – based agent simply by TELLing it what it needs to know. The agent's initial program, before it starts to receive percepts, is built by adding one by one the sentences that represent the designer's knowledge of the environment. Designing the representation language to make it easy to express this knowledge fundamental requirement that when one asks a question of the knowledge base, the answer should follow from what has been told (or rather, TELLed) to the knowledge base previously.
in the form of sentences simplifies the construction problem enormously. This is called the declarative approach to system building. In contrast, the procedural approach encodes desired behaviors directly as program code; minimizing the role of explicit representation and reasoning can result in a much more efficient system.

In addition to TELLing it what it needs to know, we can provide a knowledge – based agent with mechanisms that allow it to learn for itself. Create general knowledge about the environment out of a series of percepts. This knowledge can be incorporated into the agent's knowledge base and used for decision making. In this way, the agent can be fully autonomous.

All these capabilities – representation, reasoning, and learning – rest on the centuries – long development of the theory and technology of logic. Before explaining that theory and technology, however, we will create a simple world with which to illustrate them.

2.2. THE WUMPUS WORLD

The wumpus world is a cave consisting of rooms connected by passageways. Lurking some – where in the cave is the wumpus, a beast that eats anyone who enters its room. The wumpus can be shot by an agent, but the agent has only one arrow. Some rooms contain bottomless pits that will trap anyone who wanders into these rooms (except for the wumpus, which is too big to fall in). The only mitigating feature of living in this environment is the possibility of finding a rose. Although the wumpus world is rather tame by modern computer game standards, it makes an excellent testbed environment for intelligent agents. Michael Genesereth was the first to suggest this[6].

A sample wumpus world is shown in fig 2.1. We have to specify the performance measure, the environment, and the agent's actuators and sensors. We will group all these together under the heading of the **task environment**. For the acronymically minded, we call this the **PEAS** (**P**erformance, **E**nvironment, **A**ctuators, **S**ensors) description. In designing an agent, the first step must always be to specify the task environment as fully as possible.

By the PEAS description:

- **Performance measure:** +1000 for picking up the rose, -1000 for falling into a pit or being eaten by the wumpus, -1 for each action and -10 for using up the arrow.
- **Environment:** A 4 X 4 grid of rooms. The agent always starts in the square labeled [1,1], facing to the right. The locations of the gold and the wumpus are chosen randomly.
- **Actuators:** The agent can move forward, turn left by 90^o. the agent dies a miserable death if it enters a square containing a pit or a live wumpus. Moving forward has no effect if there is a wall in front of the agent. The action *Grab* can be used to pick up an object that 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 only has one arrow, so only the first *shoot* action has an effect.
- **Sensors:** The agent has 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 squares directly adjacent to a rose, the agent will perceive a fragrant and in the square where the rose is, the agent will perceive a fragrant.
  - → 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 in the form of a list of five symbols; for example, if there is a stench and a breeze, but no fragrant, bump, or scream, the agent will receive the percept [Stench, Breeze, None, None, None].

Exercise 2.1

In most instances of the wumpus world, it is possible for the agent to retrieve the rose safely. Occasionally, the agent must choose between going home empty – handed and risking death to find the rose, because the rose is in a pit or surrounded by pits.

Let us watch a knowledge – based wumpus agent exploring the environment shown in fig 2.1. the agent's initial knowledge base contains the rules of the environment, as listed previously; in particular, it knows that it is in [1,1] and that [1,1] is a safe square. We will see how its knowledge evolves as new percepts arrive and actions are taken.

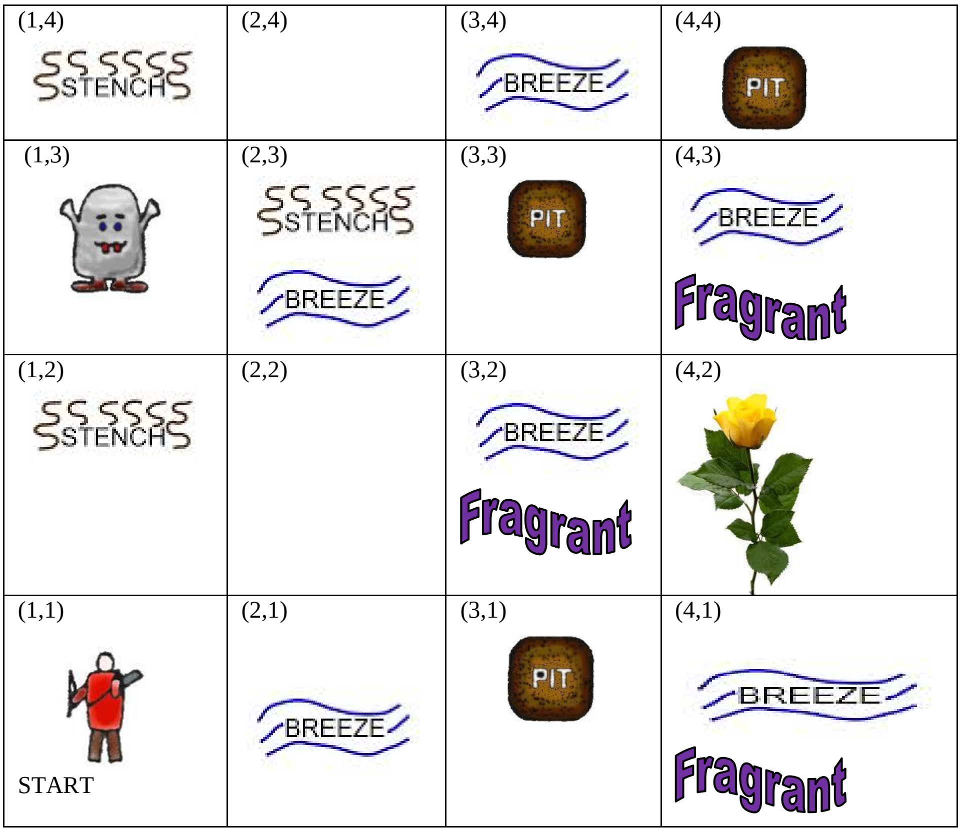

Fig :2.1 A typical wumpus world. The agent is in the bottom left corner.

The first percept is [None, None, None, None, None], from which the agent can conclude that its neighboring squares are safe. Fig 2.2 shows the agent's state of knowledge at this point.

(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) **OK**	(2,1) **OK**	(3,1)	(4,1)

Fig: 2.2 The initial situation,with percept [None, None, None, None, None]

From the fact that there was no stench or breeze in [1, 1], the agent can infer that [1, 2] and [2, 1] are free of dangers. They are marked with an ok to indicate this. A cautious agent will move only into a square that it knows is ok. Let us suppose the agent decides to move forward to [2, 1], giving the scene in Fig 2.3.

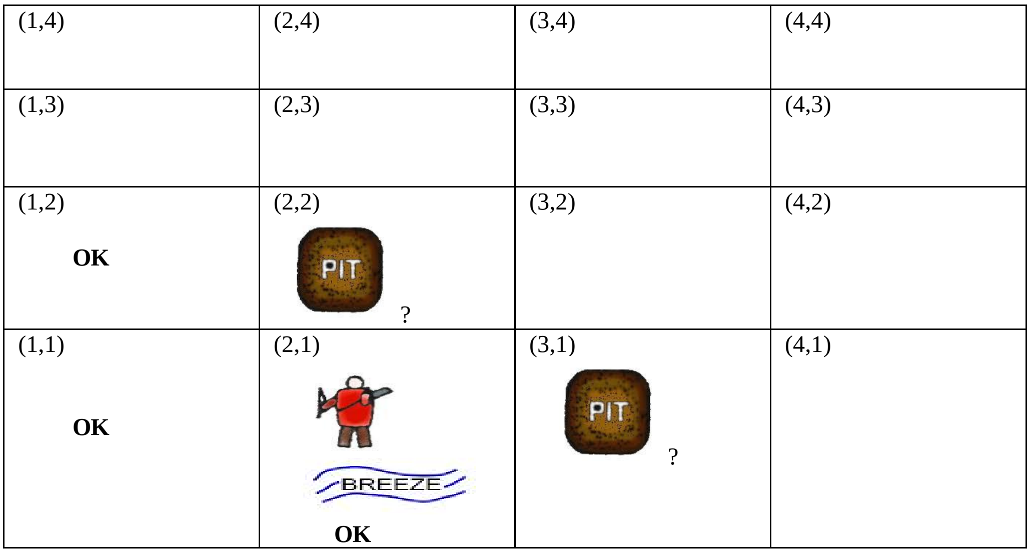

Fig 2.3 After one move, with percept [None, Breeze, None, None,None]

The agent detects a breeze 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. The notation 

? in Fig 2.3 indicates a possible pit in those squares. At this point, there is only one known square that is OK and has not been visited yet. So the prudent agent will turn around, go back to [1, 1], and then proceed to [1, 2].

The new percept in [1, 2] is [Stench, None, None, None, None], resulting in the state of knowledge shown in Figure 2.4. The stench in [1, 2] means that there must be a wumpus nearby. 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 wumpus! Indicates this. Moreover, the lack of a Breeze in [1, 2] implies that there is no pit in [2, 2]. Yet we 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 inference is beyond the abilities of most animals, but it is typical of the kind of reasoning that a logical agent does

.

(1,4)	(2,4)	(3,4)	(4,4)
(1,3)  !	(2,3)	(3,3)	(4,3)
(1,2)   OK	(2,2) OK	(3,2)	(4,2)
(1,1) Visited OK	(2,1) Visited  OK	(3,1)  !	(4,1)

Fig: 2.4 After the third move,with percept [Stench, None, None, None, None]

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.

The new percept in [2, 2] is [None, None, None, None, None], the agent can infer that [2, 3] and [3, 2] are free of dangers. They are marked with an OK to indicate this. Let us suppose the agent decides to move forward to [2, 3], giving the scene in figure 2.5

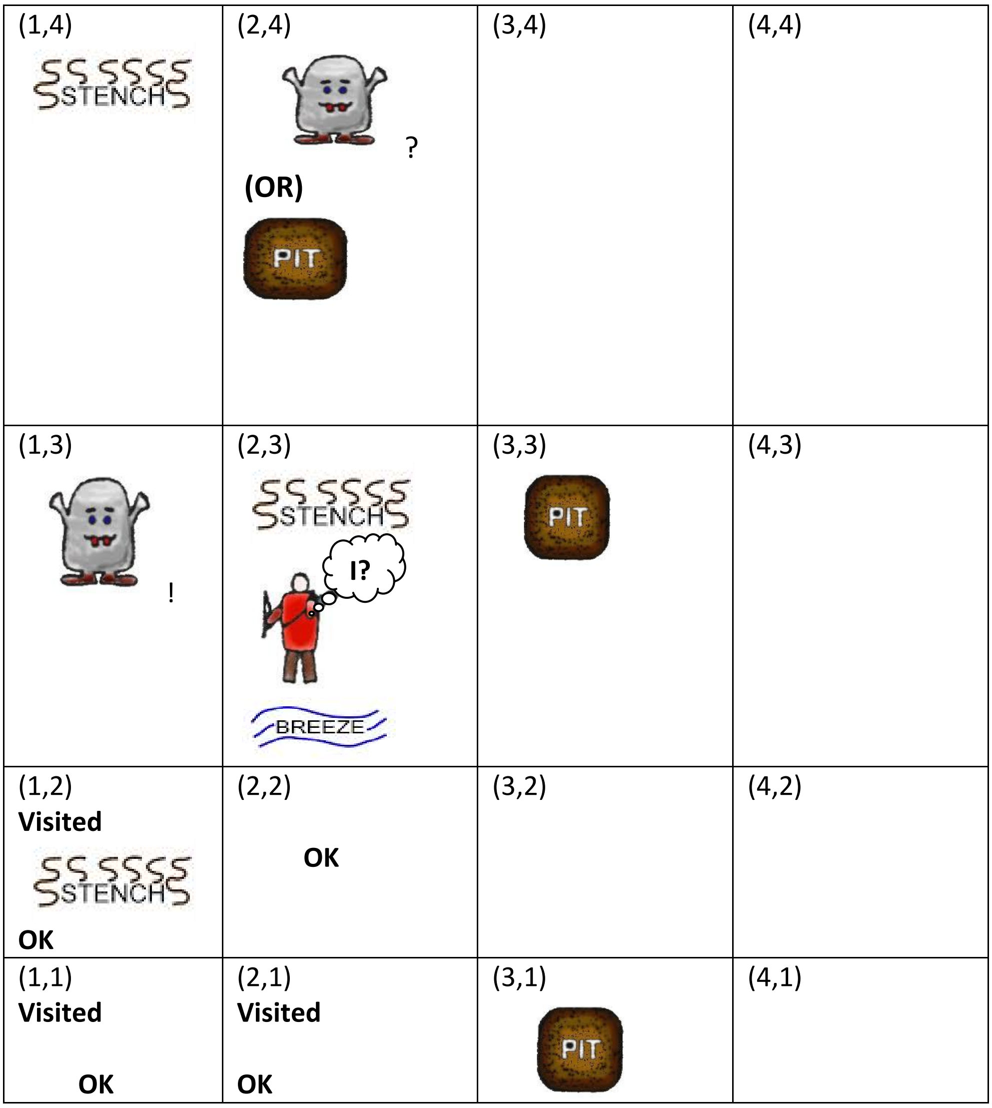

Fig: 2.5 After the fifth move, with percept [Stench,Breeze, None, None, None]

The agent detects a breeze and a stench in [2, 3], so there must be a pit and a wumpus in a neighboring square. At this point, there is only one known square that is OK and has not visited yet. So the prudent agent will turn around, go back to [2, 2], and then proceed to [3, 2].

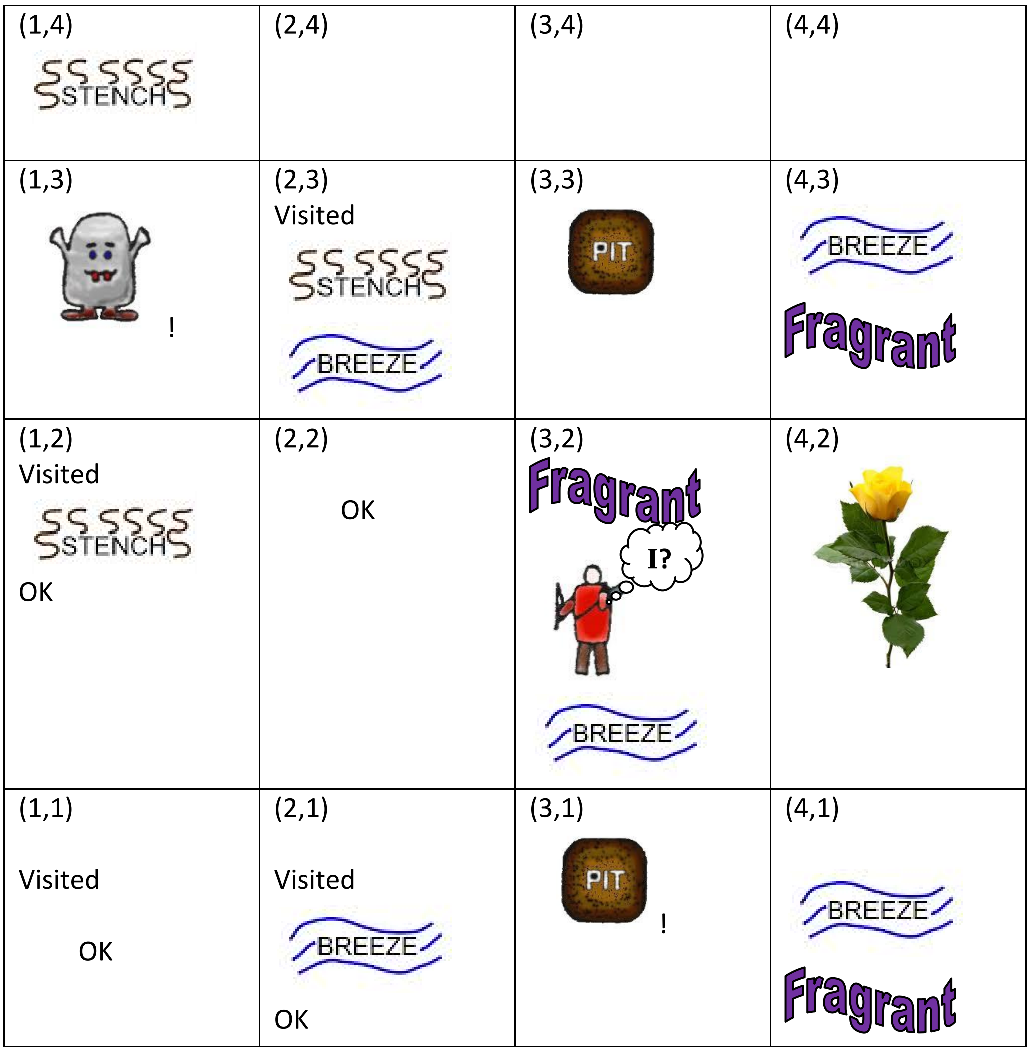

Fig :2.6 After the seventh move, with percept [Fragrant, Breeze, None, None, None]

The agent detects a breeze and a fragrant in [3, 2], so there must be a pit and a rose in a neighboring square. Since the agent detects a breeze in [2, 3] and [3, 2], therefore the pit can be in [3, 3]. In [4, 2], the agent detects a fragrant. So it should grab the rose and thereby end the game.

In each case where 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.

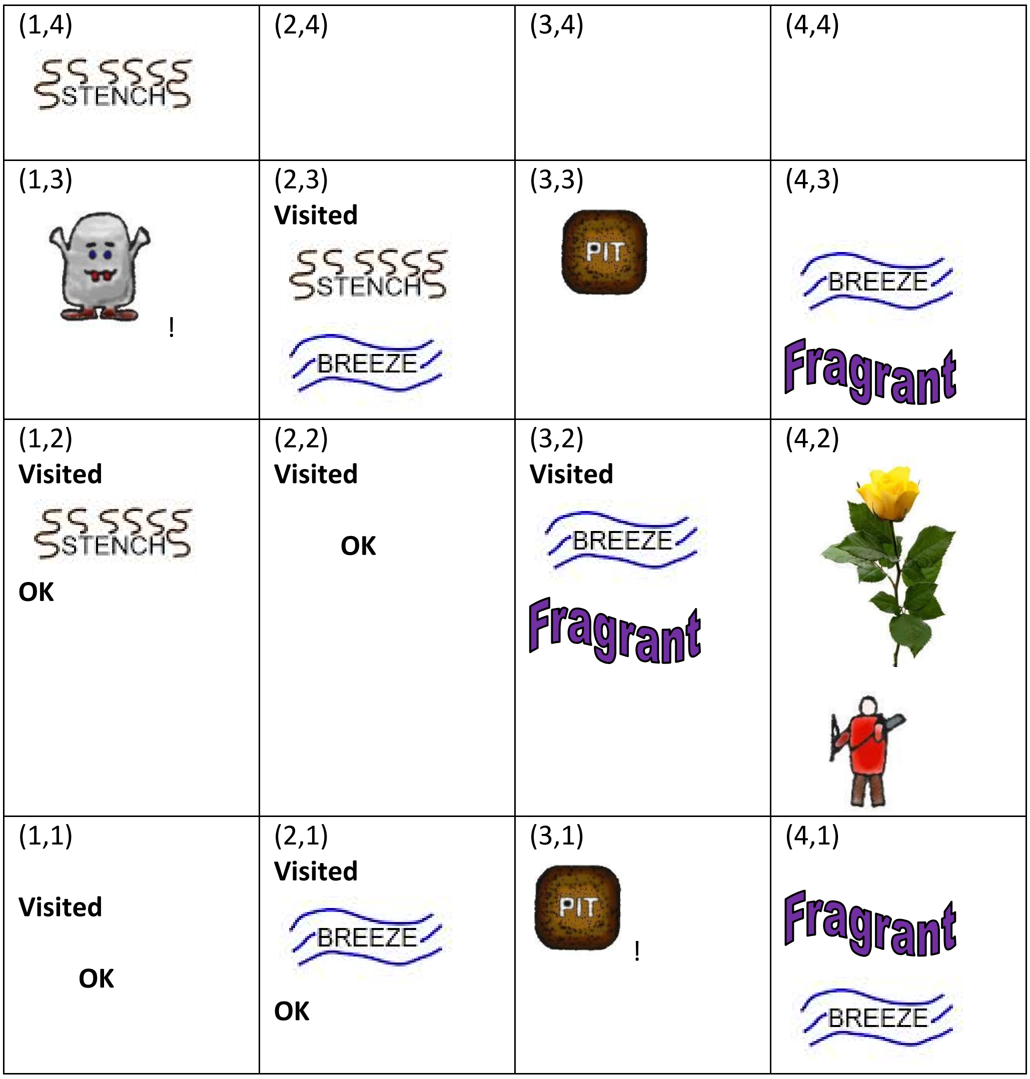

Fig :2.7 At the eighth move, the agent will grab the rose.

2.3. JAINA LOGIC:

We now present a logic called **jaina logic**. We cover the syntax of jaina logic and its semantics – the way in which the truth of sentences is determined. Then we look at **entailment** – the relation between a sentence and another sentence that follows from it – and see how this leads to a simple algorithm for logical inference. Everything takes place, of course, in the wumpus world.

SYNTAX

The **syntax** of propositional logic defines the allowable sentences. The **atomic sentences** – the indivisible syntactic elements – consist of a single **jaina symbol**. Each such symbol stands for a true, a false, a true and a false, a indeterminant, a indeterminant and a true, a indeterminant and a false and a indeterminant, a true and a false. We will use lowercase names for symbols: p, q, r and so on. The names are arbitrary but are often chosen to have some mnemonic value to the reader. For example, we might use $W_{1,3}$ to stand for the wumpus is in [1, 3]. There are seven proposition symbols with fixed meanings: True is the always – true (T) proposition, False is the always – false (F) proposition, True and False is the always – true false (TF) proposition, Indeterminant is the always – indeterminant (I) proposition, Indeterminant and True is the always – indeterminant true (IT) proposition, Indeterminant and False is the always – indeterminant false (IF) proposition, Indeterminant, True and False is the always – indeterminant, true, false (ITF) proposition.

Complex sentences bare constructed from simpler sentences using **logical connectives.** There are five connectives in common use:

$\neg$ (not): A sentence such as $\neg$ W1, 3 is called the negation of W1, 3.

$\wedge$ (and): A sentence whose main connective is $\wedge$, such as W1, 3 $\wedge$ P3, 1, is called a conjunction; its parts are the conjuncts.

$\vee$ (or): A sentence using $\vee$, such as (W1, 3 $\wedge$ P3, 1) $\vee$ W2, 2, is a disjunction of the disjuncts (W1, 3 $\wedge$ P3, 1) and W2, 2.

$\Rightarrow$ (implies): A sentence such as (W1, 3 $\wedge$ P3, 1) $\Rightarrow$ $\neg$W2, 2 is called an implication (or conditional). Its premise or antecedent is (W1, 3 $\wedge$ P3, 1), and its conclusion or consequent is $\neg$ W2, 2. Implications are also known as rules or if - then statements.

The implication symbol is sometimes written in other books as $\supset$ or $\rightarrow$.

$\Leftrightarrow$(if and only if): The sentence W1, 3 $\Leftrightarrow$ $\neg$ W2, 2 is a biconditional.

Table 2.2 gives a formal grammar of jaina logic:

Sentence $\rightarrow$ Atomic sentence / complex sentence	
Atomic Sentence $\rightarrow$	**True / False / True and False / Indeterminant / Indeterminant and True / Indeterminant and False / Indeterminant ,True and False /** Symbol.
Symbol $\rightarrow$ p / q / r / . . .	
Complex Sentence $\rightarrow$ $\neg$ Sentence	
	| (Sentence $\wedge$ Sentence)
	| (Sentence $\vee$ Sentence)
	| (Sentence $\Rightarrow$ Sentence)
	| (Sentence $\Leftrightarrow$ Sentence)

Notice that the grammar is very strict about parentheses: every sentence constructed with binary connectives must be enclosed in parentheses. This ensures that the syntax is completely unambiguous. It also means that we have to write $((A \wedge B) \Rightarrow C)$ instead of $A \wedge B \Rightarrow C$, for example. To improve readability, we will often omit parentheses, relying instead on an order of precedence for the connectives. This is similar to the procedure used in arithmetic – for example, ab+c is read as ((ab) +c) rather than a (b+c) because multiplication has higher precedence than addition. The order of precedence in jaina logic is: $\neg$, $\wedge$, $\vee$, $\Rightarrow$ and $\Leftrightarrow$. Hence, the sentence

$$\neg p \vee q \wedge r \Rightarrow s$$

Is equivalent to the sentence

$$((\neg p) \vee (q \wedge r)) \Rightarrow s.$$

Precedence does not resolve ambiguity in sentences such as A $\wedge$ B $\wedge$ C, which could be read as ((A $\wedge$ B) $\wedge$ C) or (A $\wedge$(B $\wedge$ C)). Because these two readings mean the samething according to the semantics, sentences such as A $\wedge$ B $\wedge$ C are allowed. We also allow $A \vee B \vee$ C and A $\Leftrightarrow$ B $\Leftrightarrow$ C. Sentences such as A $\Rightarrow$ B $\Rightarrow$ Care not allowed because the two readings have different meanings; we insist on parentheses in this case. Finally, we will sometimes use square brackets instead of parentheses when it makes the sentence clearer.

SEMANTICS

Having specified the syntax of jaina logic, we now specify its semantics. The semantics defines the rules for determining the truth of a sentence with respect to a particular model. In jaina logic, a model simply fixes the truth value – true or false or true false or indeterminant or indeterminant true or indeterminant false or indeterminant true false – for every proposition symbol. For example, if the sentences in the knowledge base make use of the proposition symbols $P_{1,2}$, $P_{2,2}$ and $P_{3,1}$, then one possible model is

$$m_1 = \{P_{1,2} = \text{false}, P_{2,2} = \text{false}, P_{3,1} = \text{true}\}.$$

with three proposition symbols, there are $2^3 = 8$ possible models .Notice, however, that because we have pinned down the syntax, the models become purely mathematical objects with no necessary connection to wumpus world.$P_{1,2}$ is just a symbol;it might mean"there is a pit in[1, 2]"

The semantics for propositional logic must specify how to compute the truth value of any sentence, given a model. This is done recursively. All sentences are constructed from atomic sentences and the five connectives; therefore, we need to specify hoe to compute the truth of sentences formed with each of the five connectives. Atomic Sentences are easy:

- True is true in every model
  False is false in every model
  True and False is true and false in every model
  Indeterminant is Indeterminant in every model
  Indeterminant and True is Indeterminant and True in every model
  Indeterminant and False is Indeterminant and False in every model
  Indeterminant, True and False is indeterminant, true and false in every model.
- The truth value of every other proposition symbol must be specified directly in the model. For example, in the model m, given earlier, $P_{1,2}$ is false.

Such rules reduce the truth of a complex sentence to the truth of simpler sentences. The rules for each connective can be summarized in a **truth table** that specifies the truth value of a complex sentence for each possible assignment of truth values to its components. Truth tables for the five logical connectives are given in Table 2.3. Using these tables, the truth value of any sentence s can be computed with respect to any model m by a simple process of recursive evaluation. For example, the sentence $\neg P_{1,2} \wedge (P_{2,2} \vee P_{3,1})$, evaluated in m_1, gives true $\wedge$ (false $\vee$ true) = true $\wedge$ true = true.

Previously we said that a knowledge base consists of a set of sentences. We can now see that a logical knowledge base is a conjunction of those sentences. That is, if we start with an empty KB and do TELL (KB.S,). . . .TELL (KB.Sn) then we have KB = $S_1 \wedge S_2 \wedge \ldots \wedge S_n$. This means that we can treat knowledge bases and sentences interchangeably.

We know that

$$\bar{p} = 1 - p$$
$$p \vee q = \max(p, q)$$
$$p \wedge q = \min(p, q)$$
$$p \Rightarrow q = \min(1, 1 + q - p)$$
$$p \Leftrightarrow q = 1 - |p - q|$$ [3]

p	q	~p	p∧q	p v q	p⇒q	p⇔q
T	T	F	T	T	T	T
T	F	F	F	T	F	F
T	TF	F	TF	T	TF	TF
T	I	F	I	T	I	I
T	IT	F	IT	T	IT	IT
T	IF	F	IF	T	IF	IF
T	ITF	F	ITF	T	ITF	ITF

p	q	~p	p∧q	pvq	p⇒q	p⇔q
F	T	T	F	T	T	F
F	F	T	F	F	T	T
F	TF	T	F	TF	T	ITF
F	I	T	F	I	T	ITF
F	IT	T	F	IT	T	IF
F	IF	T	F	IF	T	ITF
F	ITF	T	F	ITF	T	TF

p	q	~p	p∧q	pvq	p⇒q	p⇔q
TF	T	ITF	TF	T	T	TF
TF	F	ITF	F	TF	ITF	ITF
TF	TF	ITF	TF	TF	T	TF
TF	I	ITF	TF	I	T	ITF
TF	IT	ITF	TF	IT	T	ITF
TF	IF	ITF	TF	IF	ITF	ITF
TF	ITF	ITF	TF	ITF	T	ITF

p	q	~p	p∧q	pvq	p⇒q	p⇔q
I	T	I	I	T	T	I
I	F	I	F	I	I	I
I	TF	I	TF	I	IT	IT
I	I	I	I	I	T	T
I	IT	I	I	IT	T	ITF
I	IF	I	IF	I	ITF	ITF
I	ITF	I	I	ITF	T	IT

p	q	~p	p∧q	pvq	p⇒q	p⇔q
IT	T	IF	IT	T	T	IT
IT	F	IF	F	IT	IF	IF
IT	TF	IF	TF	IT	I	I
IT	I	IF	I	IT	ITF	ITF
IT	IT	IF	IT	IT	T	T
IT	IF	IF	IF	IT	TF	TF
IT	ITF	IF	ITF	IT	IT	IT

p	q	~p	p∧q	pvq	p⇒q	p⇔q
IF	T	IT	IF	T	T	IF
IF	F	IT	F	IF	IT	IT
IF	TF	IT	IF	TF	T	IT
IF	I	IT	IF	I	T	ITF
IF	IT	IT	IF	IT	T	TF
IF	IF	IT	IF	IF	T	TF
IF	ITF	IT	IF	ITF	T	I

p	q	~p	p∧q	pvq	p⇒q	p⇔q
ITF	T	TF	ITF	T	T	ITF
ITF	F	TF	F	ITF	TF	TF
ITF	TF	TF	TF	ITF	ITF	ITF
ITF	I	TF	I	ITF	IT	IT
ITF	IT	TF	ITF	IT	T	IT
ITF	IF	TF	TF	ITF	I	I
ITF	ITF	TF	ITF	ITF	T	T

Table:2.3 Truth tables for the five logical connectives.

The truth table for a biconditional, p ⇔ q,shows that it is true whenever both p ⇒ q and q ⇒ p are true. In English, this is often written as " p if and only if q" or " p iff q".The rules of the wumpus world are best written using ⇔.For example, a square is breezy if a neighboring square has a pit, and a square is breezy only if a neighboring square has a pit .So we need biconditionals such as

$$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1}) ,$$

Where $B_{1,1}$ means that there is a breeze in[1,1].Notice that the one-way implication

$$B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})$$

is true in the wumpus world, but incomplete. It does not rule out models in which $B_{1,1}$ is false and $P_{1,2}$ is true, which would violate the rules of the wumpus world. Another way of putting it is that the implication requires the presence of pits if there is a breeze, whereas the biconditional also requires the absence of pits if there is no breeze.

A SIMPLE KNOWLEDGE BASE

Now that we have defined the semantics for jaina logic, we can construct a knowledge base for the wumpus world. For simplicity, we will deal only with pits;

First, we need to choose our vocabulary of proposition symbols. For each i, j:

- Let $P_{i,j}$ be true if there is a pit in [i, j].
- Let $B_{i,j}$ be true if there is a breeze in [i, j].

The knowledge base includes the following sentences, each one labeled for convenience:

- There is no pit in [1, 1]:
  $R_1 : \neg P_{1,1}$
- A square is breezy if and only if there is a pit in a neighboring square. This has to be stated for each square; for now, we include just the relevant squares:
  $R_2 : B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
  $R_3 : B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
- The preceding sentences are true in all wumpus world.Now we include

the breeze percepts for the first two squares visited in the specific world the agent is in, leading up to the situation in Figure 2.4.

$R_4 : \neg B_{1,1}$

$R_5 : B_{2,1}$

The knowledge base, then consists of sentences R_1 through R_5. It can considered as a single sentence – the conjunction $R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$ – because it asserts that all the individual sentences are true.

III. CONCLUSION:

According to Jaina logic, **every attribute by being affirmed and denied according to different aspects may bring about seven fundamental propositions true of real subject,** there are no possibilities other than this. So, it is very useful to take a correct decision for the agent.

REFERENCES:

[1] *The Religion of Ahimsa* by A.Chakravarthi.

[2] *Lord Mahavir and Jain Religion*, Complied by Pravin K. Shah Jain Study Center of North Carolina

[3] **Fuzzy Sets and Fuzzy Logic: Theory and Applications** 1st Edition by George J.Klir, BoYuan

[4] *Context-Sensitivity in Jain Philosophy.A Dialogical Study of Siddhars.Igan*. I's Commentary on the handbook of logic.Nicolas Clerbout, Marie- Hélène Gorisse, Shahid Rahman

[5] *Bhagavan Mahavira : Life And Philosophy* (2001) – Acharya Tulsi

[6] Artificial Intelligence *A Modern Approach (*2003) - Stuart Russell,Peter Norvig

[7] *Many valued logics* –Nicholas J.J.Smith.

[8] *Interactions Between Philosophy and Artificial Intelligence* – Aaron Sloman,England

[9] *The Logical Way to Be Artificially Intelligent* – Robert Kowalski

[10] *Logic And Artificial Intelligence* – Nils J.Nilsson

[11] *The Jaina Concept Of Logic* (Indian Philosophical Quarterly, Vol.IX, No.4, July 1982)

[12] *Indian Logic And AI System Design*–Dr.Ananda Mohan Ghosh , Dr.Ashok Banerji