20CS/IT505

Hall Ticket Number:

III/IV B.Tech (Regular) DEGREE EXAMINATION

February, 2023 **Common to CSE & IT Branches**

Fifth Semester **Artificial Intelligence**

Time: Three Hours **Maximum:** 70 Marks

Answer Question No.1 compulsorily. ***1X14=14M***

Answer ONE question from each unit. ***4X14=56M***

1. Answer all questions

a. Define artificial intelligence.

Artificial Intelligence is the study of how to make computers do things which, at the moment, people do better.

b. What are the Advantages of DFS?

i. It requires less memory since only the nodes of the current path are stored.
ii. By chance, it may find a solution without examining much of the search space at all.

c. What is Bidirectional Search?

In Bidirectional Search run two simultaneous searches → one **forward from the initial state** another **backward from the goal**, **stop when the two searches meet.**

d. Define agent in AI?

An **agent** is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators.**

e. What is Horn clause?

Horn clause → which is a disjunction of literals of which *at most one is positive*.

f. Define knowledge based agent?

Knowledge based agent includes knowledge base and inference system, knowledge based is set of representations of real world facts.

g. Define First Order Logic in AI?

First-order logic is another way of knowledge representation in artificial intelligence. It is an extension to propositional logic. FOL is sufficiently expressive to represent the natural language statements in a concise way. First-order logic is also known as Predicate logic or First-order predicate logic.

h. Define Semantic Nets?

In semantic nets information is represented as:

set of nodes connected to each other by a set of labeled arcs.
• Nodes represent: various objects / values of the attributes of object.
• Arcs represent: relationships among nodes.

i What is PTRANS in CD?

- **PTRANS** Transfer of the physical location of an object (e.g., go or walk)

j. What are the components of Planning system?

Choose the best rule based upon heuristics.
Apply this rule to create a new state.
Detect when a solution is found.
Detect dead ends so that they can be avoided.
Detect when a nearly solved state occurs and use special methods to make it a solved state.(More complex problem solvers often add a fifth task)

k. Differentiate Forward and Backward chaining systems in prepositional logic?

Forward chaining is known as data-driven technique because we reaches to the goal using the available data. Backward chaining is known as goal-driven technique because we start from the goal and reaches the initial state in order to extract the facts.

l. List out applications of expert systems?

Information management.
Hospitals and medical facilities.
Help desks management.
Employee performance evaluation.
Loan analysis.
Virus detection.
Useful for repair and maintenance projects.
Warehouse optimization.

m. Define rote learning?

- Rote Learning is basically *memorization.* It is a basic necessity for any intelligent program -- is it a separate learning process?
- Memorization can be a complex subject -- how best to store knowledge?

n. Define Expert System?

The expert systems are the computer applications developed to solve complex problems in a particular domain, at the level of extra-ordinary human intelligence and expertise.

UNIT-I

2a) **Explain in detail about nature of environment and its properties? 7M**

Description:3m

Properties:4m

An environment in artificial intelligence is the surrounding of the agent. The agent takes input from the environment through sensors and delivers the output to the environment through actuators.

- Some programs operate in the entirely **artificial environment** confined to keyboard input, database, computer file systems and character output on a screen.
- In contrast, some software agents (software robots or softbots) exist in rich, unlimited softbots domains. The simulator has a **very detailed, complex environment**.
- The software agent needs to choose from a long array of actions in real time. A softbot designed to scan the online preferences of the customer and show interesting items to the customer works in the real as well as an artificial environment.
- The most famous artificial environment is the Turing Test environment, in which one real and other artificial agents are tested on equal ground. This is a very challenging environment as it is highly difficult for a software agent to perform as well as a human.

Properties of Environment

- The environment has multifold properties −
- **Discrete / Continuous** − If there are a limited number of distinct, clearly defined, states of the environment, the environment is discrete (For example, chess); otherwise it is continuous (For example, driving).
- **Observable / Partially Observable** − If it is possible to determine the complete state of the environment at each time point from the percepts it is observable; otherwise it is only partially observable.
- **Static / Dynamic** − If the environment does not change while an agent is acting, then it is static; otherwise it is dynamic.
- **Single agent / Multiple agents** − The environment may contain other agents which may be of the same or different kind as that of the agent.
- **Accessible / Inaccessible** − If the agent's sensory apparatus can have access to the complete state of the environment, then the environment is accessible to that agent.
- **Deterministic / Non-deterministic** − If the next state of the environment is completely determined by the current state and the actions of the agent, then the environment is deterministic; otherwise it is non-deterministic.
- **Episodic / Non-episodic** − In an episodic environment, each episode consists of the agent perceiving and then acting. The quality of its action depends just on the episode itself. Subsequent episodes do not depend on the actions in the previous episodes. Episodic environments are much simpler because the agent does not need to think ahead

b). List the different types of Agents. Explain those agents with their structure? 7M

Description:3m

Diagrams 4m

Agent: An agent is just something that acts (agent comes from the Latin agere, to do). Of course, all computer programs do something, but computer agents are expected to do more: operate autonomously, perceive their environment, persist over a prolonged time period, adapt to change, and create and pursue goals.

Four basic types in order of increasing generality:

- simple reflex agents
- reflex agents with state
- goal-based agents
- utility-based agents

All these can be turned into learning agents

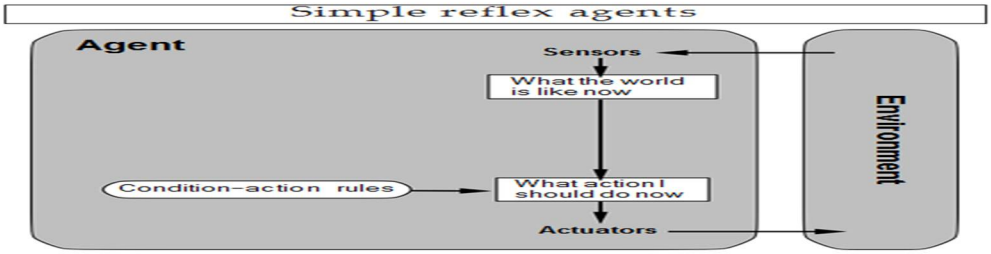
1. Simple reflex agents: The simplest kind of agent is the simple reflex agent. These agents select actions on the basis of the current percept, ignoring the rest of the percept history. It performs actions based on a current situation.

They choose actions only based on the current percept.

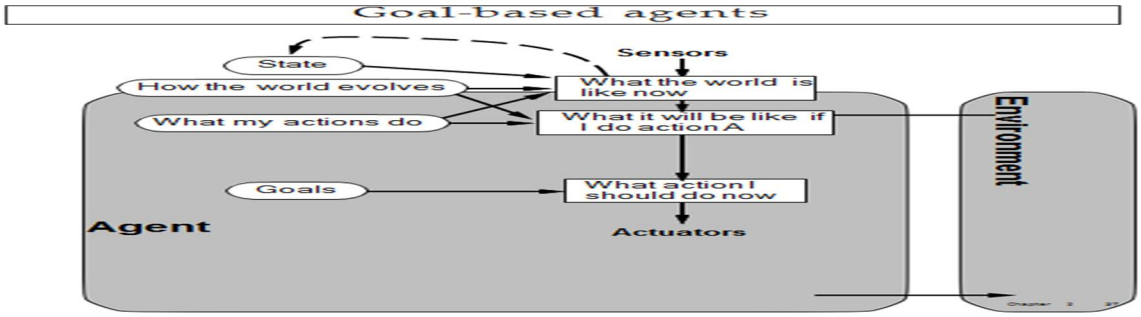
They are rational only if a correct decision is made only on the basis of current precept.

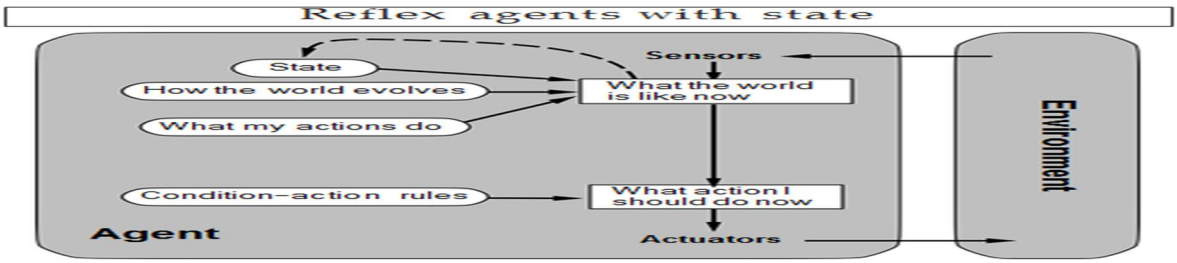
Their environment is completely observable.

Condition-Action Rule − It is a rule that maps a state (condition) to an action.



- They use a model of the world to choose their actions. They maintain an internal state.
- **Model** − knowledge about "how the things happen in the world".
- **Internal State** − It is a representation of unobserved aspects of current state depending on percept history.
- **Updating the state requires the information about** −
- How the world evolves.
- How the agent's actions affect the world.

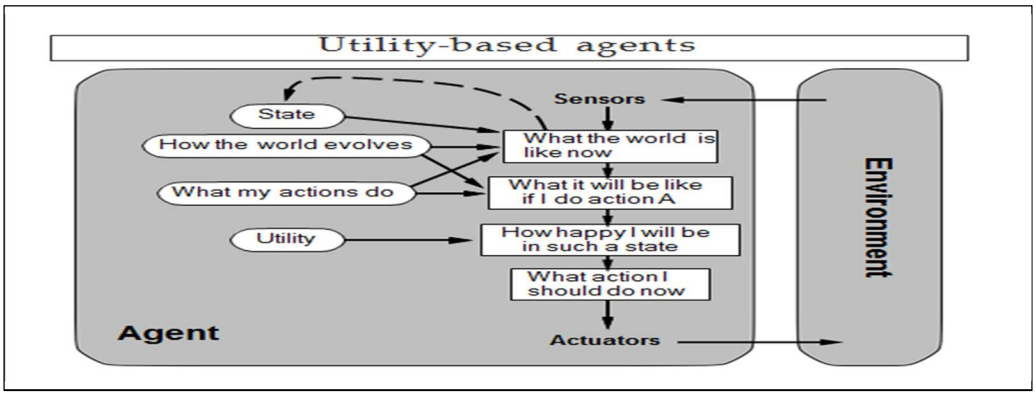




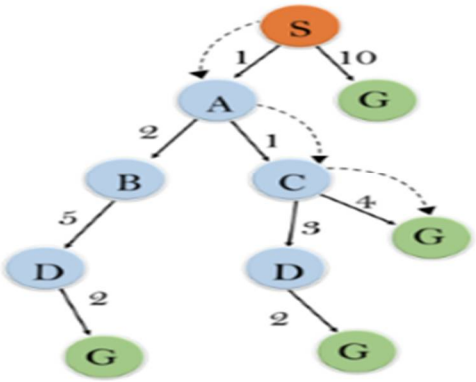
Ex: best guess for what the world is like now.

Utility Based Agents

- They choose actions based on a preference (utility) for each state.
- Goals are inadequate when −
- There are conflicting goals, out of which only few can be achieved.
- Goals have some uncertainty of being achieved and you need to weigh likelihood of success against the importance of a goal.



Solution:



Initialization: {(S, 5)}

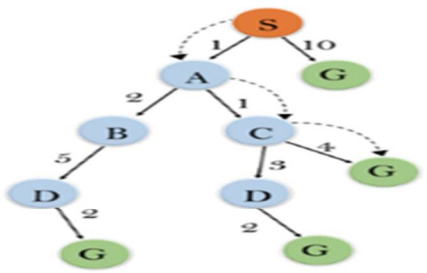
Iteration1: {(S--> A, 4), (S-->G, 10)}

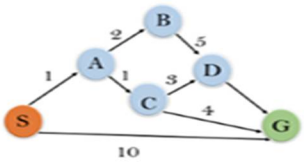
Iteration2: {(S--> A-->C, 4), (S--> A-->B, 7), (S-->G, 10)}

Iteration3: {(S--> A-->C--->G, 6), (S--> A-->C--->D, 11), (S--> A-->B, 7), (S-->G, 10)}

Iteration 4 will give the final result, as **S--->A--->C--->G** it provides the optimal path with cost 6.

Solution:





State	h(n)
S	5
A	3
B	4
C	2
D	6
G	0

Initialization: {(S, 5)}

Iteration1: {(S--> A, 4), (S-->G, 10)}

Iteration2: {(S--> A-->C, 4), (S--> A-->B, 7), (S-->G, 10)}

Iteration3: {(S--> A-->C--->G, 6), (S--> A-->C--->D, 11), (S--> A-->B, 7), (S-->G, 10)}

Iteration 4 will give the final result, as **S--->A--->C--->G** it provides the optimal path with cost 6.

b) Trace the constraint satisfaction Procedure to find the solution for the following 7M Cryptarithmetic Problem D O N A L D + G E R A L D = R O B E R T
Description:3m
Procedure:4m

DONALD+GERALD=ROBERT

2) DONALD + GERALD = ROBERT

Given 'D=5' (If not given assume D=5 at initial stage)

```
  6  5  4  3  2  1
  D  O  N  A  L  D
+ G  E  R  A  L  D
  c1 c2 c3 c4 c5
-----------------------------
  R  O  B  E  R  T
```

1. **'D=5'** is assumed, so **'D+D=T'** therefore **'T=0'** & **'c5=1'**.

2. In column 5 **'O+E=O'** as 'T=0' so E cannot be 0, therefore **'E=9'**. 'O+9=O' is possible if 'c2=1'. Therefore **'c2=1'**.
3. In column 3 **'A+A=9'**, but addition of any 2 same number is always **even**, given that addition is 9 which is possible when there is carry. Therefore **'c4=1'**, so **'A=4'**.
4. **Remaining** numbers to be assigned are **{1,2,3,6,7,8**} to **{O,N,R,B,L,G}.**

5. We have 'E=9' & 'c2=1' so from column 5 we get **'c1=1'**. Also from column2 we have **'L+L+c5=R'** where 'c5=1' therefore **R is odd** so **R can be 1or3or7**. As 'D+G' does not generate carry shown in column 6 so **R cannot be 1or3**. Therefore **'R=7'** & **'G=1'**.
6. We have 'R=7' so from column 2 we have **'L+L+1=17'**, therefore **'L=8'**.

7. From column 3 we get that **'A+A+c4=E'** and so there is no carry, therefore **'c3=0'**.
8. From column 4 we get **'N+R+c3=B'** we have R=7 & 'c3=0', so **'N+7=10+B'**, therefore **'N=B+3'**. **{2,3,6}are remaining** to be assigned so to satisfy the constraint 'N=B+3' we get **'B=3'** & **'N=6'**.
9. And remaining **'O=2'**.

SOLUTION:

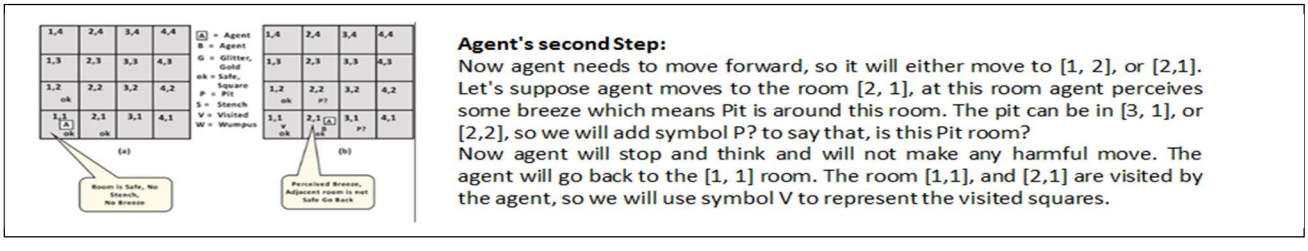
```
  5  2  6  4  8  5
+ 1  9  7  4  8  5
-----------------------
  7  2  3  9  7  0
```

VALUES:
D=5,O=2,N=6,A=4,L=8,G=1,E=9,R=7,B=3,T=0

Unit-II

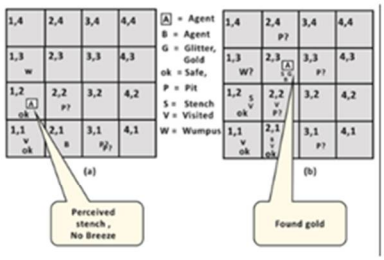
4a) Explain in detail about forward chaining and backward chaining in predicate logic? 7m
Desription:3m
Properties:4m
Forward Chaining:
The Forward-chaining algorithm starts from known facts, triggers all rules whose premises are satisfied, and add their conclusion to the known facts. This process repeats until the problem is solved.
Properties of Forward-Chaining:
It is a down-up approach, as it moves from bottom to top.
It is a process of making a conclusion based on known facts or data, by starting from the initial state and reaches the goal state.
Forward-chaining approach is also called as data-driven as we reach to the goal using available data.
Forward -chaining approach is commonly used in the expert system, such as CLIPS, business, and production rule systems.
Consider the following famous example which we will use in both approaches:
Backward-chaining:
Backward-chaining is also known as a backward deduction or backward reasoning method when using an inference engine. A backward chaining algorithm is a form of reasoning, which starts with the goal and works backward, chaining through rules to find known facts that support the goal.
Properties of Backward Chaining:
It is known as a top-down approach.
Backward-chaining is based on modus ponens inference rule.
In backward chaining, the goal is broken into sub-goal or sub-goals to prove the facts true.
It is called a goal-driven approach, as a list of goals decides which rules are selected and used.
Backward -chaining algorithm is used in game theory, automated theorem proving tools, inference engines, proof assistants, and various AI applications.
The backward-chaining method mostly used a **depth-first search** strategy for proof.

4b) Explain in detail about wumpus world problem? 7m
Description:3m
Explanation:4m
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. Now we will explore the Wumpus world and will determine how the agent will find its goal by applying logical reasoning.
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.



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.

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].



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.

5a) Explain in detail about knowledge based engineering? 7m
Description:3m
Explanation:4m

- 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).**
- ✓ 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.**
- The agents 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 **three things**.
- 1) **First**, it **TELLS** the knowledge base what it perceives.
- 2) **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 current state** of the world, about the **outcomes** of possible action sequences, and so on.
- 3) **Third**, 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.

5b) What is basis of Resolution? Explain about Unification Algorithm. 7m
Description:3m
Algorithm:4m
Resolution is a theorem proving technique that proceeds by building refutation proofs, i.e., proofs by contradictions. Resolution is used, if there are various statements are given, and we need to prove a conclusion of those statements. Unification is a key concept in proofs by resolutions. Resolution is a single inference rule which can efficiently operate on the **conjunctive normal form or clausal form**.

We can apply the resolution procedure to a very simple inference in the wumpus world. When the agent is in [1,1], there is no breeze, so there can be no pits in neighboring squares. The relevant knowledge base is

$$KB = R_2 \wedge R_4 = (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$$

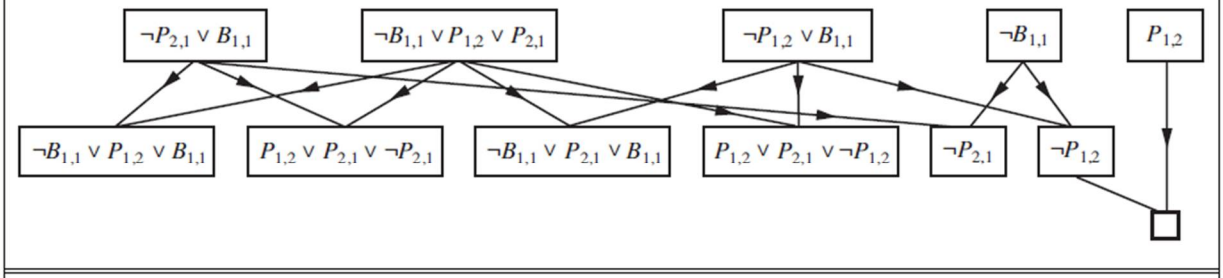


Figure 7.13 Partial application of PL-RESOLUTION to a simple inference in the wumpus world. $\neg P_{1,2}$ is shown to follow from the first four clauses in the top row.

When we convert (KB ∧ ¬α) into CNF, we obtain the clauses shown at the top of Figure 7.13.
The second row of the figure shows clauses obtained by resolving pairs in the first row. Then, when P1,2 is resolved with ¬P1,2, we obtain the empty clause, shown as a **small square**.

function PL-RESOLUTION(KB, α) **returns** *true* or *false*
  inputs: KB, the knowledge base, a sentence in propositional logic
    α, the query, a sentence in propositional logic

  clauses ← the set of clauses in the CNF representation of $KB \wedge \neg\alpha$
  new ← { }
  loop do
    for each pair of clauses C_i, C_j **in** *clauses* **do**
      resolvents ← PL-RESOLVE(C_i, C_j)
      if *resolvents* contains the empty clause **then return** *true*
      new ← *new* ∪ *resolvents*
    if *new* ⊆ *clauses* **then return** *false*
    clauses ← *clauses* ∪ *new*

Figure 7.12 A simple resolution algorithm for propositional logic. The function PL-RESOLVE returns the set of all possible clauses obtained by resolving its two inputs.

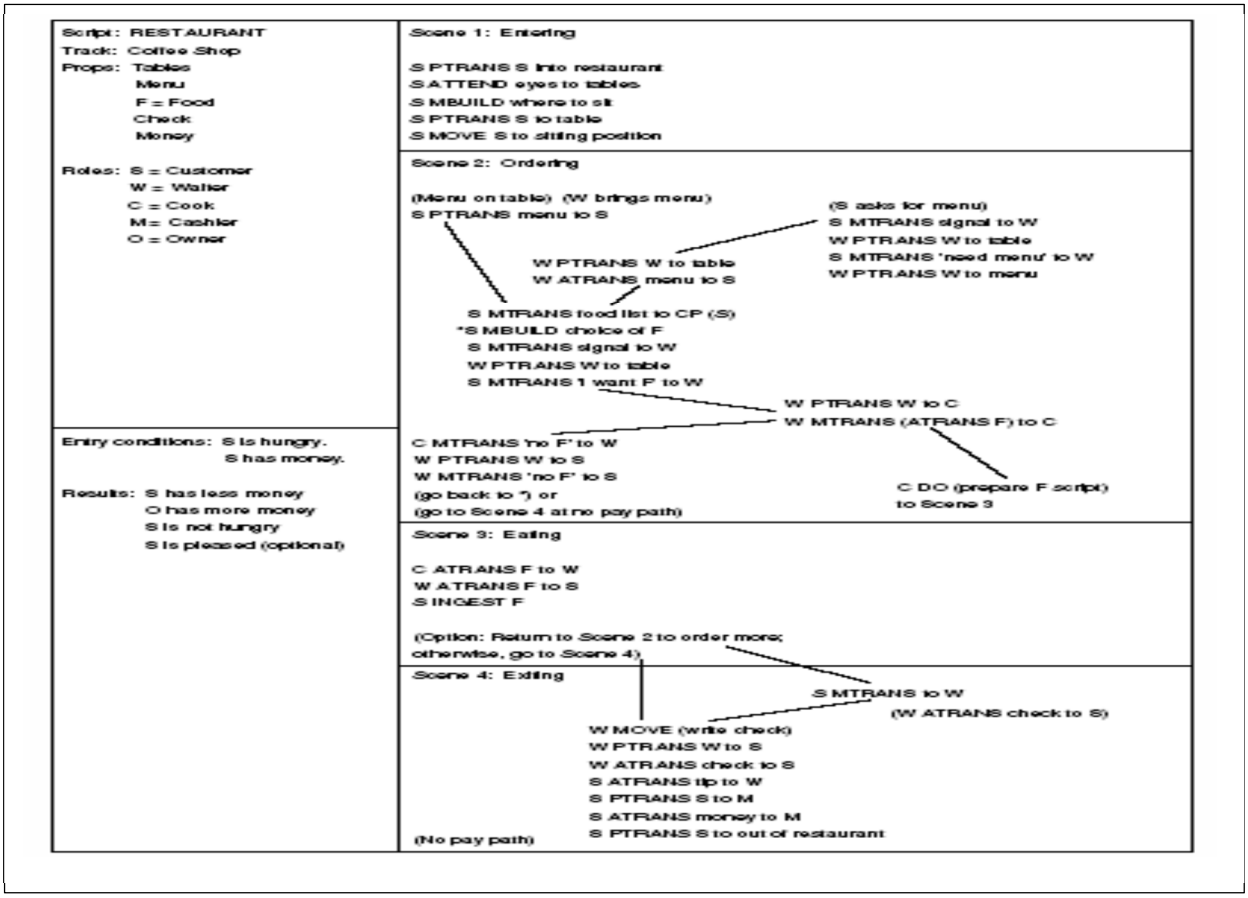
Unit-III

6a) Explain Script, Components of script and restaurant Script example. 7m
Description:3m
Example: 4m

Script (1977):- (Schank and Abelson, 1977). Script is a structure which is used to represent the knowledge . Script is a structure that describe a sequence of events in particular context scripts are frame like structures used to represent commonly occurring experiences such as going to movies, shopping in supermarket, eating in restaurant, banking. A script consist set of slots and information (knowledge) contained in it

Various components of Script are

Script Name: - Title
Track: - Special situation, specific variation
Roles:- peoples involve in the event described in script
Entry condition: - required pre situation to execute the script
Props: - non live object involve in the Script
Scenes: - The actual sequence of events that occur
Result: - Condition that will be True after execution of script



6b) What are primitives for Conceptual dependencies? Construct the CD Relationship for the sentence 'Since Smoking can kill you, I stopped'. 7m

Description:4m

Example:3m

Conceptual Dependency is a high level strong filler structure which is used to represent high level complicated knowledge which requires solving complicated problems. CD is collection of symbols which are used to represent knowledge. CD is the graphical presentation of high level knowledge. CD is a theory of how to represent the kind of natural about events that is usually contained in natural language sentences

Various Primitives (Symbols) used in CD

The set of primitive actions in CD, proposed by Shank and Abelson, is presented below

1. ATRANS: - Transfer of an abstract entity (e.g. Give)
2. PTRANS: - Transfer of an physical location of an object (Go)
3. PROPEL:- application of physical force to an object (Push)
4. MOVE:- movement of body part by it's owner (Kick)
5. GRASP: - Grasping or holding the object tightly by an actor(Clutch , Grasp)
6. INGEST: - Ingestion of an object by an animal (eat)
7. XPEL:- expulsion of something from the body of animal (Cry)
8. MTRANS: - transfer of an mental information (Tell)
9. MBUILD:- Building a new information out of old (Decide)
10. SPEAK: - Production of sound
11. ATTEND: - Focusing of a sense organ towards a stimulus (listen)

(OR)

7a) Explain about goal stack planning with example? 7m

Description:3M

Example:4M

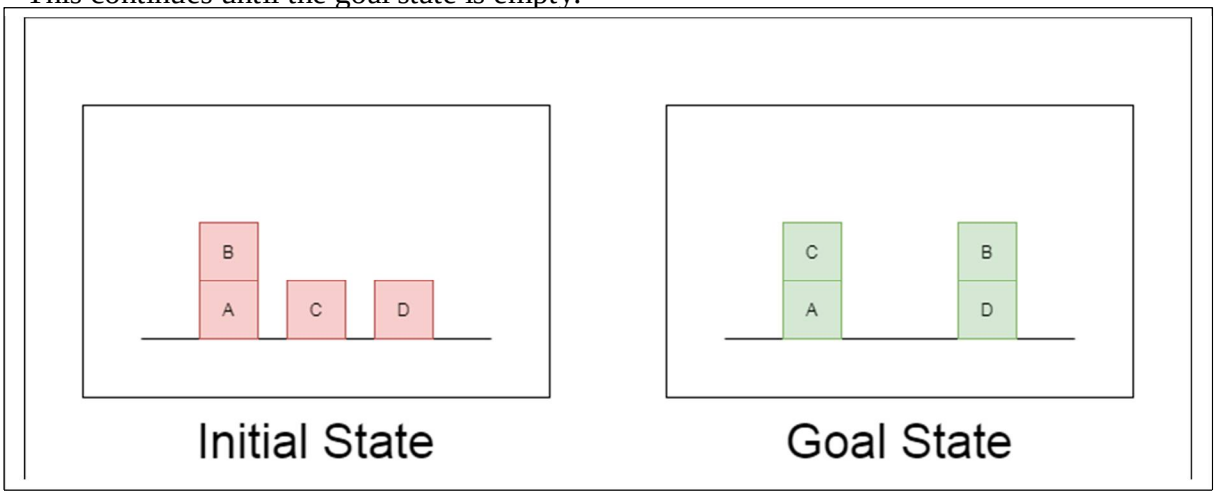
Basic Idea to handle interactive **compound goals** uses goal stacks, Here the **stack** contains :

- **goals,**
- **operators** -- ADD, DELETE and PREREQUISITE lists
- **a database** maintaining the current situation for each operator used.

Consider the following where wish to proceed from the ***start* to *goal* state**.

The method is to

- Investigate the **first node** on the stack ie **the top goal.**
- If a **sequence of operators** is found that satisfies this goal it is removed and the next goal is attempted.
- This continues until the goal state is empty.

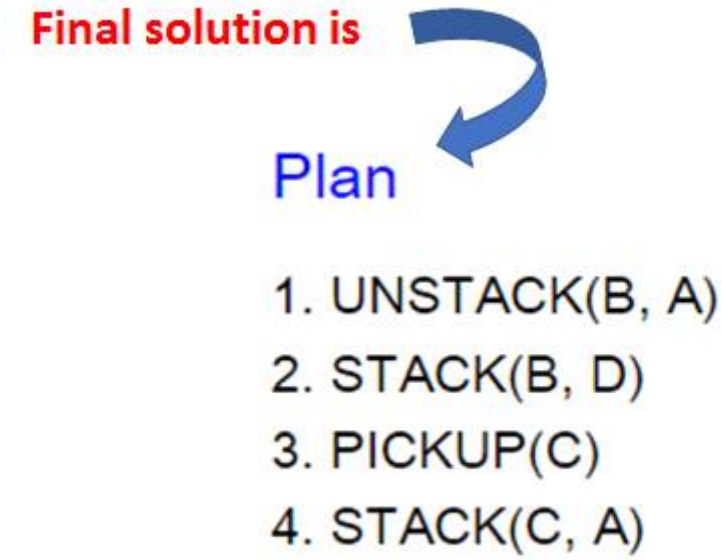


sider **alternative 1** above further:

- The first goal ON(*C*,*A*) is **not true** and the only operator that would make it true is STAC which replaces ON(*C*,*A*) giving:
    - **STACK(C, A)**
    - **ON(B, D)**
    - **ON(C, A) ON(B, D) OTAD**

But in order to apply **STACK(C, A), its preconditions must hold, so we establish them as sub goals.** This produces the new goal stack as

CLEAR (A)
HOLDING (C)



erator that could make it true is

7b) Describe Hierarchical Planning with an Example? 7m

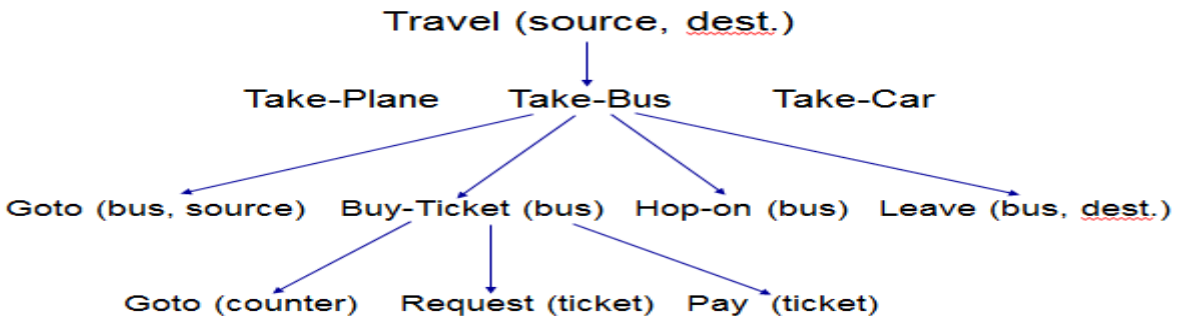
Principle ----2m
Procedure---2m
Example-----3m

Principle

- hierarchical organization of 'actions'
- complex and less complex (or: abstract) actions
- lowest level reflects directly executable actions

Procedure

- planning starts with complex action on top
- plan constructed through action decomposition
- substitute complex action with plan of less complex actions (pre-defined plan schemata; or learning of plans/plan abstraction)
- overall plan must generate effect of complex action
  Hierarchical Planning / Plan Decomposition
  Plans are organized in a hierarchy. Links between nodes at different levels in the hierarchy denote Decomposition of a “complex action” into more primitive actions (operator expansion).
  Example:



Unit-IV

8a) Discuss the MYCIN Expert system focusing on the certainty factor used. 7m
Description:3M
Conditions: 4M

MYClN attempts to solve its goal of recommending a therapy for a particular patient by first finding the cause of the patient's illness. It uses its production rules to reason backward from goals to clinical observations.

To solve the top-level diagnostic goal, it looks for rules whose right sides suggest diseases. It then uses the left sides of those rules (the preconditions) to set up sub goals whose success would enable the rules to be invoked. These sub goals are again matched against rules, and their preconditions are used to set up additional sub goals.

Whenever a precondition describes a specific piece of clinical evidence, MYCIN uses that evidence if it already has access to it. Otherwise, it asks the user to provide the information.

MYCIN is reasoning backward, the question can easily be answered by examining the goal tree. Doing so provides two kinds of information:

- What higher-level question might the system be able to answer if it had the requested piece of information? (In this case, it could help determine the category of ORGANISM-l.)
- What other information does the system already have that makes it think that the requested piece of knowledge would help"? (In this case, facts [2.1] to [2.4].)

Is the patient's illness with OHGANiSM-1 a hospital-acquired infection? WHY
[i.e., WHY is it important to determine whether or not the infection with ORGANISM-1 was acquired while the patient was hospitalized?]
[2.0] This will aid in determining the category of ORGANISM-i. it has already been established that
[2.1] the gram stain of OHGANlSM-l is gramneg, and {2.2} the morphology of ORGANISM-1 is rod, and
[2.3] the aerobicity of ORGANISM-1 is facultative Therefore, it [2.4] the infection with ORGANISM-1 was not acquired while the patient was hospitalized. Then there is strongly suggestive evidence (.8) that the category of ORGANlSM-t is enter bacteriaceae {ALSO there is weakly suggestive evidence (.1) that
the identity oiORGANlSM-'l is pseudomonasaaerUQinosa} [HULED37] HOW [2-3] [i.e., HOW was it established that the aerobicity of ORGANISM-1 is facuitative?] The toilowing were used: [3.1] FlULEo2? indicated there is strongly suggestive evidence (B) that the aerobicity of ORGANISM-1 is facultative.
[3.2] RULEDSS indicated there is weakly suggestive evidence (.2) that the aerobicity of ORGANISM-i
is not facultative. Since this gave a cumulative CF of (fa) for iacultative, the evidence indicates that the aerobicity of ORGANlSM-I is facultative.

8b) Describe learning in problem solving with example? 7m
Description for each method----2m
List of methods:1m
There are three basic methods in which a system can learn from its own experiences.

- Learning by Parameter Adjustment.
- Learning by Macro Operators.
- Learning by Chunking.

Learning by Parameter Adjustment
So the basic idea of idea of parameter adjustment is to:

- Start with some estimate of the correct weight settings.
- Modify the weight in the program on the basis of accumulated experiences.
- Features that appear to be good predictors will have their weights increased and bad ones will be decreased.

Learning by Macro Operators
The basic idea here is similar to Rote Learning:
Avoid expensive computation
Macro-operators can be used to group a whole series of actions into one.
For example: Making dinner can be described a lay the table, cook dinner, serves dinner. We could treat laying the table as on action even though it involves a sequence of actions.
Learning by Chunking
Chunking involves similar ideas to Macro Operators and originates from psychological ideas on memory and problem solving.
The computational basis is in production systems (studied earlier).
SOAR is a system that use production rules to represent its knowledge. It also employs chunking to learn from experience.

(OR)

9a) Explain about knowledge acquisition in building expert systems? 7m
Description----3m
Explanation---------4m

Typically, a knowledge engineer interviews a domain expert to elucidate expert knowledge, which is then translated into rules. After the initial system is built, it must be iteratively refined until it approximates expert-level performance. This process is expensive and time-consuming, so it is worthwhile to look for more automatic ways of constructing expert knowledge bases. While no totally automatic knowledge acquisition systems yet exist, there are many programs that interact with domain experts to extract expert knowledge efficiently.
These programs provide support for the following activities:

- Entering knowledge.
- Maintaining knowledge base consistency.
- Ensuring knowledge base completeness

The most useful knowledge acquisition programs are those that are restricted to a particular problem-solving paradigm, e.g., diagnosis or design. It is important to be able to enumerate the roles that knowledge can play in the problem-solving process.
The acquisition proceeds through several steps:

1. Initial knowledge base construction. MOLE asks the expert to list common symptoms or complaints that might require diagnosis, For each symptom, MOLE prompts for a list of possible explanations.
2. Refinement of the knowledge base.
MOLE now tries to identify the weaknesses of the knowledge base- One approach is to find holes and prompt the expert to fill them.
It is difficult, in general, to know whether a knowledge base is complete, so instead MOLE lets the expert watch MOLE-p solving sample problems.

9b) Explain each Component of Expert System and how Explanation is provided by it? 7m
Diagram---3m
Characteristics, capabilities and explanation—4m

What are Expert Systems?

The expert systems are the computer applications developed to solve complex problems in a particular domain, at the level of extra-ordinary human intelligence and expertise.

Characteristics of Expert Systems

- High performance
- Understandable
- Reliable
- Highly responsive

Capabilities of Expert Systems

The expert systems are capable of −

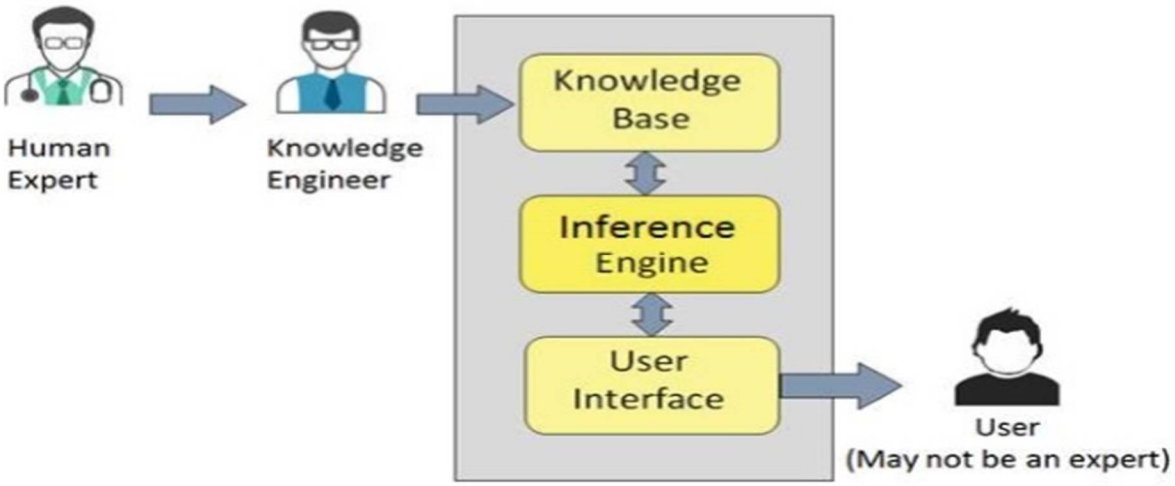
- Advising
- Instructing and assisting human in decision making
- Demonstrating
- Deriving a solution
- Diagnosing
- Explaining
- Interpreting input
- Predicting results
- Justifying the conclusion
- Suggesting alternative options to a problem

Components of Expert Systems

The components of ES include −

- Knowledge Base
- Inference Engine
- User Interface

Let us see them one by one briefly −



EXPLANATION:

In order for an expert system to be an effective tool, people must be able to interact with it easily. To facilitate this interaction, the expert system must have the following two capabilities in addition to the ability to perform its underlying task:

Explain its reasoning. In many of the domains in which expert systems operate, people will not accept results unless they have been convinced of the accuracy of the reasoning process that produced those results.

Acquire new knowledge and modifications of old knowledge. Since expert systems derive their power from the richness of the knowledge bases they exploit

Signature of the internal Examiner (B. Krishnaiah)

Name of the external Examiners	Name of the college	Dept.	Signature

Signature of the HOD