PART - A

UNIT – I

1. **What is Intelligence?**
   Intelligence is the ability to acquire, understand and apply knowledge, or the ability to exercise thought and reason.

2. **Describe the four categories under which AI is classified with examples.**
   AI definitions may be organized into four categories.
   1. Systems that think like humans (General Problem Solver).
   2. Systems that act like humans (natural language processing)
   3. Systems that think rationally (Formal logic).
   4. Systems that act rationally (Intelligent agents).

3. **Define Artificial Intelligence.**
   The art of creating machines that perform functions that require intelligence when performed by human beings.

4. **List the fields that form the basis for AI.**
   Mathematics, Psychology and Linguistics

5. **What is a Knowledge Based System? Explain.**
   A **knowledge-based system** is a program for extending and/or querying a knowledge base. The dictionary defines **a knowledge-based system** as a computer system that is programmed to imitate human problem-solving by means of artificial intelligence and reference to a database of knowledge on a particular subject. Knowledge-based systems are systems based on the methods and techniques of Artificial Intelligence. Their core components are the knowledge base and the inference mechanisms.

6. **List a few of the task domains of AI.**
   Mundane tasks – perception, NLP, Robot control, Common sense reasoning
   Formal tasks – Games, Mathematics
   Expert tasks – Scientific analysis, Medical diagnosis, Financial analysis

7. **Describe the components of a KBS.**
   The general block diagram of a KBS is

Input / Output Device	**Inference Engine**	**Knowledge Base**

8. **What id meta-knowledge?**

Meta-knowledge may be loosely defined as "knowledge about knowledge". Meta-knowledge includes information about the knowledge the system possesses, about the efficiency of certain methods used by the system, the probabilities of the success of past plans, etc. The meta-knowledge is generally used to guide future planning or execution phases of a system.

9. Expand LISP and PROLOG.

LISP – LISt Processing
PROLOG - PROgramming LOGic

10. What is a Production System?

Production systems are a good way to describe the operations that can be performed in search for a solution to a problem. They consist of

1. A set of rules each consisting of a left side (a pattern) that determines the applicability of the rule and a right side that describes the operation to be performed if the rule is applied.
2. One or more knowledge/databases that contain whatever information is appropriate for the particular task.
3. A control strategy that specifies the order in which the rules will be compared to the database and a way of resolving the conflicts that arise when several rules match at once.
4. A rule applier.

11. Define state-space search technique.

State space search involves the use of a graph to keep track of the relationships between states. Each node of the graph represents a *state* of the problem. Each arc in the graph represents the application of an operator in the search process. A state must encapsulate all the relevant information necessary to decide "what to do next", and none of the irrelevant information. The representation of states is critical when defining a problem as state space search. (Clever representations may reduce computation). The solution is the sequence of operators which transform the state state to the goal state

12. List the steps in performing a state-space search.

State space search involves finding a path from the initial state of a search problem to a goal state. To do this, we build a *search graph*, starting from the initial state (or the goal state). We *expand* a state by applying the search operators to that state, generating ALL of its successor states. These successors are in the next level down of the search graph. The order in which we choose states for expansion is determined by the *search strategy.* Different strategies result in (sometimes massive) different behaviour. KEY CONCEPT: We want to find the solution while realizing in memory as few as possible of the nodes in the search space.

13. What is heuristic search?

The basic idea of heuristic search is that, rather than trying all possible search

paths, you try and focus on paths that seem to be getting you nearer your goal state. To use heuristic search you need an *evaluation function* that scores a node in the search tree according to how close to the target/goal state it seems to be. This will just be a guess, but it should still be useful.

14. Differentiate Informed & Uninformed search. Give examples.

Uninformed search – also called as 'blind search'. The search operators test for a solution. The search should proceed in a systematic way by exploring in some predefined order by simply selecting the nodes at random. Eg. DFS, BFS, Bidirectional search.

Informed search – more information than the initial state, the operators and goal state are available. The search space is thus constrained making search efficient. They use and depend upon heuristic information.

15. Define the logic behind – Hill climbing, Best-First Search, BFS and DFS.

In **hill climbing** the basic idea is to always head towards a state which is better than the current one. So, if you are at town A and you can get to town B and town C (and your target is town D) then you should make a move IF town B or C appear nearer to town D than town A does. In *steepest ascent* hill climbing you will always make your next state the best successor of your current state, and will only make a move if that successor is better than your current state.

BFS explores all nodes at a given depth before proceeding to he next level i.e. all immediate children of nodes are explored before any of the children's children are considered.

DFS performs the search by diving downward into a tree as quickly as possible. It generates a child node from the most recently expanded node, then generating that child's children and so on until a goal is found or some cutoff depth d is reached. If a goal is not found when a leaf node is reached or a cutoff point, the program backtracks to the most recently expanded node and generates another of its children.

16. What do you mean by Game Playing?

Games naturally require search procedures and mostly they are of heuristic nature.

17. What are the components of a Game software?

The two important knowledge-based components of a *good* game-playing program are:

- a good plausible-move generator and
- a good static evaluation function.

18. What is a plausible-move generator? What is its role?

The plausible move generator is one in which only some small number of promising moves are generated. This is useful as it reduces the search space.

19. Define alpha & beta values in a game tree.

A modified strategy in game trees is called *alpha-hero pruning.* It requires the maintenance of two threshold values, one representing a lower bound on the

value that a maximizing node may ultimately be assigned (we call this *alpha)* and a value representing an upper bound on the value that a minimizing node may be assigned (we call this beta*).*

20. Mention some of the knowledge representation techniques.

Logic, Semantic nets, Frames, Scripts, Conceptual dependency.

UNIT - II

1. Differentiate prepositional & predicate logic.

Propositional Logic is a tool for reasoning. It provide basic concepts used in many computer science fields and also used in many medical applications. The components are:

- Proposition
- Basic operators
- Language
- Truth table
- Boolean Algebra

Predicate Logic provides a formalism for performing this analysis of prepositions and additional methods for reasoning with quantified expressions. The term predicate logic derives from the fact that we analyze prepositions into predicate-argument compositions.

2. What is clausal form? How is it useful?

Logic studies the relationship of implication between assumptions and conclusions. It does not tell whether the assumptions are correct, likely, meaningful, etc., but can determine what follows from making those assumptions (what else would be true if the assumptions also were). To demonstrate that assumptions imply a conclusion, it is helpful to construct a proof consisting of inference steps. For the proof to be convincing, the steps need to be direct and obvious, so the sentences should be unambiguous and their grammar should be as simple as possible. The clausal form of logic fits this simplicity requirement. Its simplest sentences are *atomic sentences*, which name relationships between individuals:

likes(sweetie, rover)
it_is_raining.
gives(eve, adam, apple).

3. Define a well-formed formula (wff).

A Formula is a term (string) in prepositional logic. A Well-formed formula (WFF) is a term that is constructed correctly according to propositional logic syntax rules. A WFF consists of

Constants: *False, True*

Variables: *P, Q, R*

If *a* is WFF, ¬*a* is WFF

If *a* and *b* are WFF, *aΛb* are WFF

If *a* and *b* are WFF, *avb* are WFF

If *a* and *b* are WFF, $a \rightarrow b$ are WFF

If *a* and *b* are WFF, $a \leftrightarrow b$ are WFF

Any formula that cannot be constructed using these rules are not WFF

Precedence of Logical Operators

$\neg$ $\wedge$ $\vee$ $\rightarrow$ $\leftrightarrow$

4. List some of the rules of inference.

The sound rules are:

a) **Modus ponens** (or implication-elimination) - From an implication and the premise of the implication you can infer the conclusion. If there is an axiom E F and an axiom E, then F follows logically.

b) **Modus tolens** - If there is an axiom E F and an axiom F, then E follows logically.

c) **Resolution -** If there is an axiom E F and an axiom F G then E G follows logically. In fact, resolution can subsume both modus ponens and modus tolens. It can also be generalized so that there can be any number of disjuncts in either of the two resolving expressions, including just one. The only requirement is that one expression contains the negation of one disjunct from the other.

5. What is resolution /refutation?

Resolution produces proofs by

Converting problems into a canonical form. All axioms are converted to clauses in **Conjunctive Normal Form**.

Using **refutation** - ie, by attempting to show that the negated assertion produces a contradiction with known axioms in the database. This requires using the **resolution inference rule** and **unification**.

6. Define unification.

Substitution of variables to make resolution possible is called Unification and proof in predicate logic can be based on this joint method of Unification and resolution.

7. What are semantic nets?

Semantic networks are knowledge representation schemes involving nodes and links (arcs or arrows) between nodes. The nodes represent objects or concepts and the links represent relations between nodes. The links are directed and labeled; thus, a semantic network is a directed graph. In print, the nodes are usually represented by circles or boxes and the links are drawn as arrows between the circles.

8. What are frames? How do they differ from semantic nets.

The idea of ***semantic networks*** started out as a natural way to represent labeled connections between entities. But, as the representations are expected to support increasingly large ranges of problem solving tasks, the representation

schemes necessarily become increasingly complex. In particular, it becomes necessary to assign more structure to nodes, as well as to links.
For example, in many cases we need node labels that can be computed, rather than being fixed in advance. It is natural to use database ideas to keep track of everything, and the nodes and their relations begin to look more like ***frames***.
In the literature, the distinction between frames and semantic networks is actually rather blurred. However, the more structure a system has, the more likely it is to be termed a frame system rather than a semantic network. For our purposes, we shall use the ***practical distinction*** that semantic networks look like networks, and frames look like frames.

9. What are script? What is its use?

Scripts are frame-like structures used to represent commonly occurring experiences such as going to the movies, shopping in a super market, etc. Like in a play, script structures are designed in terms of actors, roles, properties and scenes.

10. List the components of a script.

Entities – Picture Producers (PP) and Picture Aiders (PA)
Actions
Conceptual cases
Conceptual dependencies
Conceptual tenses

11. Mention the frame manipulation primitives.

1. Frame invocation
2. Frame switching
3. Frame modification
4. Frame addition and deletion
5. Frame linking
6. Frame merging and splitting

12. Define forward and backward chaining. Differentiate the same.

There are two main methods of reasoning when using inference rules: backward chaining and forward chaining.
Forward chaining starts with the data available and uses the inference rules to conclude more data until a desired goal is reached. An inference engine using forward chaining searches the inference rules until it finds one in which the if-clause is known to be true. It then concludes the then-clause and adds this information to its data. It would continue to do this until a goal is reached. Because the data available determines which inference rules are used, this method is also called *data driven*.

Backward chaining starts with a list of goals and works backwards to see if there is data which will allow it to conclude any of these goals. An inference engine using backward chaining would search the inference rules until it finds one which has a then-clause that matches a desired goal. If the if-clause of that inference rule is not known to be true, then it is added to the list of goals.

13. What is means-end analysis?

The purpose of means-end analysis is to identify a procedure that causes a transition from the current state to the goal state or at least an intermediate state that is closer to the goal state. The identified procedure reduces the observed difference between the current state and the goal state.

14. Mention the strategies used in resolving clauses (unit-preference, set-of-support, best first)

Unit-preference strategy – gives preference to resolutions involving the clauses with the smallest number of literals.

Set-of-support strategy – allows only resolutions involving the negated theorem.

Best first strategy – resolves all possible pairs of the initial clauses, then resolves all possible pairs level by level.

UNIT - III

1. Describe Bayes theorem.

Bayes theorem is a result in probability theory, which relates the conditional and marginal probability distributions of random variables. In some interpretations of probability, Bayes' theorem tells how to update or revise beliefs in light of new evidence: *a posteriori*.

To derive the theorem, we start from the definition of conditional probability. The probability of event *A* given event *B* is

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)}.$$

Likewise, the probability of event *B* given event *A* is

$$\Pr(B|A) = \frac{\Pr(A \cap B)}{\Pr(A)}.$$

Rearranging and combining these two equations, we find

$$\Pr(A|B)\ \Pr(B) = \Pr(A \cap B) = \Pr(B|A)\ \Pr(A).$$

This lemma is sometimes called the product rule for probabilities. Dividing both sides by Pr(*B*), providing that it is non-zero, we obtain Bayes' theorem:

$$\Pr(A|B) = \frac{\Pr(B|A)\ \Pr(A)}{\Pr(B)}.$$

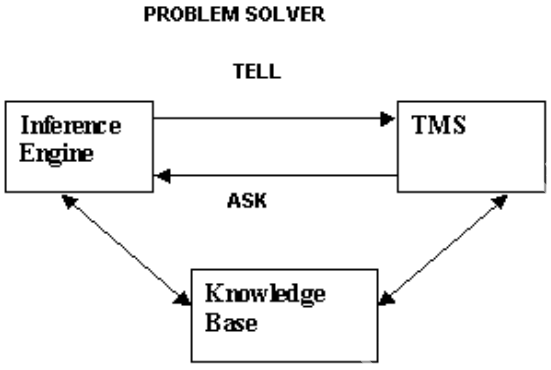
2. What are the disadvantages of Closed World Assumption (CWA). How will you overcome it?

In CWA consistency is maintained by using a KB consisting of Horn clauses. If a KB is consistent and Horn, then its CWA augmentation is consistent. (Horn clauses are clauses with at most one positive literal.). The global nature of the negation is failure assumption and this a serious drawback of CWA. And indeed there are many applications where it is not appropriate.

Truth maintenance systems (also known as belief revision and revision maintenance systems) are companion components to inference systems. The main job of the TMS is to maintain consistency of the knowledge being used by the problem solver and not to perform any inference functions.

Fig. Illustrates the role played by the TMS as part

of the problem solver. The inference engine (IE) solves domain problems based on its current belief set, while the TMS maintains the currently active belief set. The updating process is incremental. After each inference, information is exchanged between the two components. The IE tells the TMS what deductions it has made. The TMS, in turn, asks questions about current beliefs and reasons for failures. It maintains a consistent set of beliefs for the IE to work with even if new knowledge is added and removed.



Architecture of the problem solver with a TMS

This is however overcome by using completion formulas such as modal and temporal logics. CWA is essentially the formalism under which Prolog operates and Prolog has been shown to be effective in numerous applications.

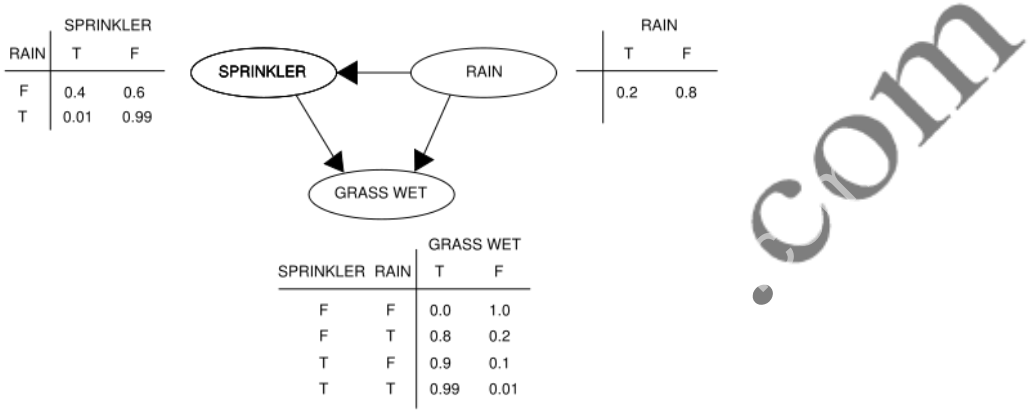
3. Define Non monotonic reasoning.

In many situations, new facts become known which contradict and invalidate old knowledge. The old knowledge when retracted causes other dependent knowledge to become invalid, thereby requiring further retractions. The retractions lead to a shrinkage or nonmonotonic growth in the knowledge. This is shrinkage of knowledge is called as nonmonotonic reasoning.

4. **What are Truth Maintenance Systems? Draw its block diagram.**

5. **What are Bayesian networks? Give an example.**
   A **bayesian network** (or a **belief network**) is a directed acyclic graph which represents independencies embodied in a given joint probability distribution over a set of variables. Nodes can represent any kind of variable, be it a measured parameter, a latent variable or a hypothesis. They are not restricted to representing random variables; which forms the "Bayesian" aspect of a Bayesian network. In this graph, nodes correspond to variables of interest, and edges between two nodes correspond to a possible dependence between variables, one that does not disappear regardless of what information we may receive. An absence of an edge between two nodes *X* and *Y* means that there is some set of variables *Z* such that *X* and *Y* become independent given *Z* (*X* and *Y* are conditionally independent given *Z*). If a given *X* and *Y* are not connected by an

edge, the sets Z which render X and Y independent can be determined by a graphical condition called d-separation.



6. **What is fuzzy logic? What is its use?**
   Fuzzy logic allows us to represent set membership as a possibility of probability distribution. (E.g. facts such as 'very tall', 'slightly cold', etc.)
   It is useful in defining reasoning systems based on techniques for combining distributions.

UNIT – IV

1. **Construct parse trees for given sentences. (sentences will be given)**
2. **Define linguistics. List the general classification of languages.**
   Linguistics is the scientific study of language, which can be theoretical or applied. Someone who engages in this study is called a linguist.

Theoretical (or general) linguistics encompasses a number of sub-fields, such as the study of language structure (grammar), and meaning (semantics). The study of grammar encompasses morphology (formation and alteration of words) and syntax (the rules that determine the way words combine into phrases and sentences). Phonology, the study of sound systems and abstract sound units, and phonetics, which is concerned with the actual properties of speech sounds (phones) as well as those of non-speech sounds, and how they are produced and perceived, also form part of this field.

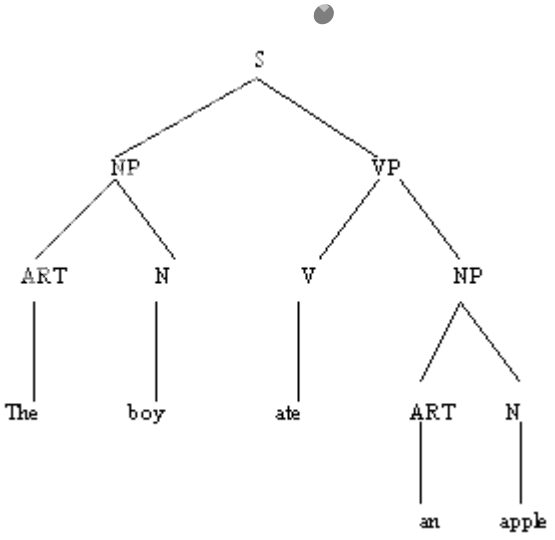
Linguistics compares languages (comparative linguistics) and explores their histories, in order to find universal properties of language and to account for its development and origins (historical linguistics). **Applied linguistics** puts linguistic theories into practice in areas such as foreign language teaching, speech therapy, translation and speech pathology.

3. **What are grammars?**

A grammar has the following components.

- A set of **terminal** symbols.
  These are the symbols that make up the strings generated by the grammar. Terminals are written with lower case letters. By definition, terminals cannot be replaced in the course of a derivation.
- A set of **non-terminal** symbols.
  These are the symbols that can be replaced during a derivation. Non-terminals are written with upper case letters. By definition, non-terminals cannot appear in the strings generated by a given grammar.
- A **start symbol** (usually **S**).
  This is the symbol that the derivations start with.
- A set of (context-free) **rules**
  Each rule consists of an arrow that has a non-terminal symbol to its left and a sequence of one or more symbols to its right. This sequence may contain terminals and non-terminals, as well as the special symbol **#.**

4. **Give the syntactic tree for the sentence „The boy ate the apple".**



5. **List the types of grammars.**

Regular, Context-free, Context-sensitive, Recursively enumerable.

6. **What is parsing? What is its importance?**

Parsing is the process of analyzing a sentence by taking it apart word by word and determining its structure from its constituent parts and subparts. This structure can be represented with a syntactic tree or a list. The parsing process is generally the inverse of the sentence generation process as it involves finding a grammatical sentence structure from an input string.

When given an input string, parsing helps in finding the lexical parts to be identified and analyzed for its meanings and role they play.

7. **Differentiate – Top down & Bottom Up parsing, Deterministic & Non deterministic parsing,.**

   A **top-down parser** begins by hypothesizing a sentence (the symbol S) and successively predicting lower level constituents until individual preterminal symbols are written. These are then replaced by the input sentence words which match the terminal categories.

   A **bottom-up parse**, on the other hand, begins with the actual words appearing in the sentence and is, therefore, data driven.

   Parsers may also be classified as **deterministic or nondeterministic** depending on the parsing strategy employed. A deterministic parser permits only one choice (arc) for each word category. Thus, each arc will have a different test condition. Consequently, if an incorrect test choice is accepted from some state, the parse will fail since the parser cannot backtrack to an alternative choice.

   Nondeterministic parsers permit different arcs to be labeled with the same test. Consequently, the next test from any given state may not be uniquely determined by the state and the current input word. The parser must guess at the proper constituent and then backtrack if the guess is later proven to be wrong. This will require saving more than one potential structure during parts of the network traversal.

8. **What are Recursive transition networks (RTN), Augmented Transition Networks (ATN)?**

   A recursive transition network (RTN) is a transition network which permits arc labels to refer to other networks (including the network's own name), and they in turn may refer back to the referring network rather than just permitting word categories used previously.

   When an RTN is given additional semantic features, it is called augmented transition network or ATN.

9. **What is the role of semantic analysis in NLP?**

   Content determination is concerned with what details to include in an explanation, a request, a question or argument in order to convey the meanings set forth by the goals of the speaker. This means the speaker must know what the hearer already knows, what the hearer needs to know, and what the hearer wants to know. These topics are related to the domain, task, and discourse context described above.

   Text planning is the process of organizing the content to be communicated so as to best achieve the goals of the speaker.

   Realization is the process of mapping the organized content to actual text. This requires that specific words and phrases be chosen and formulated into a syntactic structure.

10. **Define Natural Language generation.**

Natural Language Generation (NLG) is the natural language processing task of generating natural language from a machine representation system such as a knowledge base or a logical form.

Some people view NLG as the opposite of natural language understanding. The difference can be put this way: whereas in natural language understanding the system needs to disambiguate the input sentence to produce the machine representation language, in NLG the system needs to make decisions about how to put a concept into words.

11. List any two NLP systems.

LUNAR, LIFER, and SHRDLU.

12. What is distributed reasoning?

A *distributed reasoning system* is one that is composed of separate modules (called *agents*) and a set of communication paths between them.

- Each agent usually acts as a problem solving entity itself.
- Many kinds of system exist ranging across:
  - Centralised control systems with shared knowledge,
  - Totally distributed control and knowledge.

Such a system must provide:

- Good coordination between the agents.
- Good communication structure.
- Distributed reasoning techniques.

13. What are Intelligent Agents? What are its use?

An **intelligent agent** is one that does the right thing. As a first approximation, the right action is the one that will cause the agent to be most successful.

For each possible percept sequence, an ideal intelligent agent does whatever action is expected to maximize its performance measure, on the basis of the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

UNIT – V

1. What are Expert Systems?

An ***expert system*** is a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice.

2. Briefly explain the knowledge acquistion process.

Knowledge acquisition process is an important task in building a KBS and the encoding of knowledge has to be derived from expert sources such as experts, journal articles, databases, etc.

3. List the characteristic features of a expert system.

- Goal driven reasoning or backward chaining - an inference technique which uses IF THEN rules to repetitively break a goal into smaller sub-goals which are easier to prove;

- Coping with uncertainty - the ability of the system to reason with rules and data which are not precisely known;
- Data driven reasoning or forward chaining - an inference technique which uses IF THEN rules to deduce a problem solution from initial data;
- Data representation - the way in which the problem specific data in the system is stored and accessed;
- User interface - that portion of the code which creates an easy to use system;
- Explanations - the ability of the system to explain the reasoning process that it used to reach a recommendation.

4. **Mention some of the key applications of ES.**
   Medical diagnosis, Mining, NLP, Automated learning

5. **What is learning? What are its types?**
   Learning is acquiring knowledge. The types of learning are:
   1. Memorization
   2. Direct instruction
   3. Analogical learning
   4. Induction
   5. Deduction

6. **Define generalization, specialization. Give examples.**
   Generalization is the process that makes a general or estimate of a large class, the target concept be made after observing only some fraction of the objects belonging to that class.
   Specialization –

7. **Define Inductive Bias.**
   Bias is collectively all of those factors that influence the selection of hypotheses excluding factors directly related to the training examples. When induction is associated it becomes inductive bias.

8. **What is Explanation Based Learning? How is it useful?**
   EBL is the form of deductive generalization.
   It is useful as it can formulate a generalized explanation of the goal concept.

PART – B

UNIT – I

1. Given an example game tree perform min-max search. Find the alpha –beta values and also give the algorithm to find the same.
2. Solve the 8 puzzle problem using hill climbing heuristics. Find the heuristic function and trace the problem for a given condition.
3. Give a brief note on the ways AI systems are organized (thinking humanly, acting humanly, thinking rationally, acting rationally)
4. Write a detailed note on the importance of AI and the various fields it interacts with.
5. What is a KBS? Give a detailed description.
6. Describe the state space search technique for Water Jug problem, cannibals and Missionaries problem, 8 puzzle problem.
7. Give a detailed note on the types of Production systems.
8. Describe any one of the Ai programming languages with suitable examples and language constructs.
9. Write short notes on (or) discuss the algorithms for – heuristic search, BFS, DFS, hill climbing, Best-First search.
10. Describe any procedure used in Game Playing (Minmax Search procedure).
11. Write a detailed note on Alpha-Beta cutoffs.

UNIT – II

1. Problems in Propostional and Predicate logic (to write formulae)
2. With an example prove the concept of Resolution.
3. Explain the steps involved in converting a wff into clausal form with an example.
4. What are inference rules? Explain in detail.
5. Discuss the process of unification with an example. Also give its algorithm.
6. Given a situation – to draw a semantic net.
7. For a given description – give the frame representation.
8. Give a detailed note on the various components in Conceptual dependency.
9. With the aid of an example explain the representation of knowledge using rules.
10. Explain the functionality of Partitioned semantic networks with an example.
11. To represent given situation using Conceptual dependency.
12. To write a script for a given situation.

UNIT – III

1. Write short notes on – Dependency directed backtracking, Modal & Temporal Logics, Default Logic, Abduction.
2. Compare “rule based approach to uncertainty” and “Dempster-Shafer theory”. Illustrate it with an example.
3. Discuss Non monotonic reasoning with suitable examples.

4. **Discuss the role of Bayes theorem for reasoning in AI.**
5. **Explain the role of Bayesian networks in inference.**
6. **Write short notes on Fuzzy Logic.**

UNIT – IV

1. **Give an example of each of the four types of 0, 1, 2 and 3 for Chomsky"s hierarchy of Grammar.**
2. **Describe the components of natural language processing in detail.**
3. **What are the advantages of transformational grammars?**
4. **Write short notes on – Case grammars, Systemic grammars, Semantic grammars. Give examples.**
5. **Explain the transition network for any English sentence.**
6. **Discuss the various parsing techniques in detail.**
7. **With an example explain a simple parser in Prolog.**
8. **Write a detailed note on RTN and ATN. Give suitable examples.**
9. **Describe in detail the approaches of semantic analysis in NLP (lexical & Compositional).**
10. **Discuss the process of Natural Language Generation with examples.**
11. **Give a detailed note on any two NLP systems (LUNA, LIFER< SHRDLU).**
12. **Explain the process of Distributed Reasoning in detail.**
13. **Explain the role of Intelligent agents in AI.**
    Also could be asked to
14. **draw parse trees for given sentences.**
15. **draw RTN, ATN for given descriptions.**
16. **acceptance of given phrases by given grammar.**

UNIT – V

1. **Design an expert system for the military to use in decision making when one"s forces are under nuclear attack. Draw the architecture and explain its components.**
2. **Consider the problem of building a program to learn the user preferences while searching for a given data / information. Construct a learning model for this application.**
3. **Give a detailed note on the various non production expert system architectures. (associated networks, frame based, black board based, analogy, neural networks, etc.)**
4. **Write short notes on KAS, Knowledge building tools.**
5. **With a neat diagram explain the functionality of a general learning model.**
6. **Give detailed notes on Learning by Induction.**
7. **Discuss the generalization rules with suitable examples.**
8. **Explain Inductive Bias with an example.**
9. **Explain EBL with an example.**