UNIT III KNOWLEDGE INFERENCE

PART - A

1. Define FOL.
FOL is a first order logic. It is a representational language of knowledge which is powerful than propositional logic (i.e.) Boolean Logic. It is an expressive, declarative, compositional language.

2. Define a knowledge Base:
Knowledge base is the central component of knowledge base agent and it is described as a set of representations of facts about the world.

3. With an example, show objects, properties functions and relations.
Example
"EVIL KING JOHN BROTHER OF RICHARD RULED ENGLAND IN 1200"
Objects : John, Richard, England, 1200
Relation : Ruled
Properties : Evil, King
Functions : BROTHER OF

4. Define a Sentence?
Each individual representation of facts is called a sentence. The sentences are expressed in a language called as knowledge representation language.

5. Define an inference procedure
An inference procedure reports whether or not a sentence is entiled by knowledge base provided a knowledge base and a sentence .An inference procedure 'i' can be described by the sentences that it can derive. If i can derive from knowledge base, we can write. KB --Alpha is derived from KB or i derives alpha from KB.

6. Define Ontological commitment.
The difference between propositional and first order logic is in the ontological commitment. It assumes about the nature of reality.

7. Define Epistemological commitment.
The logic that allows the possible states of knowledge with respect to each fact.

8. Define domain and domain elements.
The set of objects is called domain, sometimes these objects are referred as domain elements.

9. What are the three levels in describing knowledge based agent?
- Logical level
- Implementation level
- Knowledge level or epistemological level

10. Define Syntax?
Syntax is the arrangement of words. Syntax of a knowledge describes the possible configurations that can constitute sentences. Syntax of the language describes how to make sentences.

11. Define Semantics
The semantics of the language defines the truth of each sentence with respect to each possible world. With this semantics, when a particular configuration exists with in an agent, the agent believes the corresponding sentence.

12. Define Logic
Logic is one which consist of

i. A formal system for describing states of affairs, consisting of a) Syntax b)Semantics.
ii. Proof Theory – a set of rules for deducing the entailment of a set sentences.

13. What are the 3 types of symbol which is used to indicate objects, relations and functions?
i) Constant symbols for objects
ii) Predicate symbols for relations
iii) Function symbols for functions

14. Define terms.
A term is a logical expression that refers to an object. We use 3 symbols to build a term.

15. Define Atomic sentence.
Atomic sentence is formed by both objects and relations.
Example
Brother (William, Richard)
William is the brother of Richard.

16. Define Quantifier and it's types.
Quantifiers are used to express properties of entire collection of objects rather than representing the objects by name.
Types:
i. Universal Quantifier
ii. Existential Quantifier
iii. Nested Quantifier.

17. What are the two we use to query and answer in knowledge base?
ASK and TELL.

18. Define kinship domain.
The domain of family relationship is called kinship domain which consists of objects unary predicate, binary predicate, function, relation.

19. Define syntactic sugar.
Extension to the standard syntax (i.e.) procedure that does not change the semantics (i.e.) meaning is called syntactic sugar.

20. Define synchronic and diachronic sentence.
Sentences dealing with same time are called synchronic sentences. Sentences that allow reasoning "a cross time" are called diachronic sentence.
21. What are the 2 types of synchronic rules?
i. Diagnostic rules
ii. Casual rules.

22. Define skolem constant.
The existential sentence says there is some object satisfying a condition, and the instantiation process is just giving a name to that object. That name must not belong to another object. The new name is called skolem constant.

23.What is truth Preserving
An inference algorithm that derives only entailed sentences is called sound or truth preserving
.
24 .Define a Proof
A sequence of application of inference rules is called a proof. Finding proof is exactly finding solution to search problems. If the successor function is defined to generate all possible applications of inference rules then the search algorithms can be applied to find proofs.

25.Define a Complete inference procedure
An inference procedure is complete if it can derive all true conditions from a set of premises.

26.Define Interpretation
Interpretation specifies exactly which objects, relations and functions are referred to by the constant predicate, and function symbols.

27.Define Validity of a sentence
A sentence is valid or necessarily true if and only if it is true under all possible interpretation in all posssible world.

28.Define Satistiability of a sentence
A sentence is satisfiable if and only if there is some interpretation in some world for which it is true.

29. Define true sentence
A sentence is true under a particular interpretation if the state of affairs it represents is the case.

30. What are the basic Components of propositional logic?
i. Logical Constants (True, False)

31 .Define Modus Ponen's rule in Propositional logic?
The standard patterns of inference that can be applied to derive chains of conclusions that lead to the desired goal is said to be Modus Ponen's rule.

32 .Define AND –Elimination rule in propositional logic
AND elimination rule states that from a given conjunction it is possible to inference any of the conjuncts. OR-Introduction rule states that from, a sentence, we can infer its disjunction with anything.
35. Define Unification.
Lifted Inference rule require finding substitutions that make different logical expressions look identical (same). This is called Unification.

36. Define Occur check.
When matching a variable in 2 expressions against a complex term, one must check whether the variable itself occurs inside the term, If it does the match fails. This is called occur check.

37. Define pattern matching.
The inner loop of an algorithm involves finding all the possible unifiers with facts in the KB. This is called pattern matching.

38. Explain the function of Rete Algorithm?
This algorithm preprocess the set of rules in KB to constant a sort of data flow network in which each node is a literals from rule a premise.

39. Define magic set.
To rewrite the rule set, using information from the goal, so that only relevant variable bindings called magic set.

40. Define backward chaining.
This algorithm works backward from the goal, chaining through rules to find known facts that support the proof.

41. Define Prolog program.
It is a set of definite clauses written in a notation somewhat different from standard FOL.

42. What are the divisions of knowledge in OTTER theorem?
i. Set of Support (SOS)

ii. Usable axioms
iii. Rewrites (or) Demodulators
iv. A set of parameters and sentences

43. What are the 2 types of frame problem?
i. Representational Frame Problem
ii. Inferential Frame Problem

44. What are the 2 types of processes?
i. Discrete events – it have definite structure
ii. Liquid events - Categories of events with process.

45. Define fluent calculus.
Discard Situation Calculus and invent a new formalism for writing axioms is Called Fluent Calculus.

46. What is important for agent?
Time (i.e.) intervals is important for agent to take an action.
There are 2 kinds;
i. Moments
ii. Extended Intervals

47. Define runtime variables.
Plans to gather and use information are represented using short hand
Notation called runtime variables (n).
Example
[Look up (Agent, "Phone number (Divya)".N), Dial (n)]

PART – B

1. **Describe about types of Knowledge representation?**

There are four types of Knowledge representation :
Relational, Inheritable, Inferential, and Declarative/Procedural.

◊ **Relational Knowledge :**

provides a framework to compare two objects based on equivalent attributes.
any instance in which two different objects are compared is a relational type of knowledge.

◊ **Inheritable Knowledge**

− is obtained from associated objects.
− it prescribes a structure in which new objects are created which may inherit all or a subset of attributes from existing objects.

◊ **Inferential Knowledge**

− is inferred from objects through relations among objects.
− e.g., a word alone is a simple syntax, but with the help of other
words in phrase the reader may infer more from a word; this inference within linguistic is called semantics.

◊ **Declarative Knowledge**

− a statement in which knowledge is specified, but the use to which that knowledge is to be put is not given.
− e.g. laws, people's name; these are facts which can stand alone, not
dependent on other knowledge;

Procedural Knowledge

− a representation in which the control information, to use the knowledge, is embedded in the knowledge itself.
− e.g. computer programs, directions, and recipes; these indicate specific use or implementation.

2. Discuss in details about inferential knowledge?

This knowledge generates new information from the given information.
This new information does not require further data gathering form soue, but does require analysis of the given information to generate new knowledge.
− given a set of relations and values, one may infer other values or relations.
− a predicate logic (a mathematical deduction) is used to infer from a set of attributes.
− inference through predicate logic uses a set of logical operations to relate individual data.
− the symbols used for the logic operations are :

" → " (implication), " ¬ " (not), " V " (or), " Λ " (and),
" ∀ " (for all), " ∃ " (there exists).

Examples of predicate logic statements :
1. *"Wonder"* is a name of a dog : **dog (wonder)**
2.All dogs belong to the class of animals : **∀ x : dog (x) → animal(x)**
3.All animals either live on land or in **∀ x : animal(x) → live (x,**

water : **land) V live (x, water)**
From these three statements we can infer that :
" Wonder **lives either on land or on water."**

3. ***Discuss about*** **Predicate Logic ?**

Predicate Logic

The propositional logic, is not powerful enough for all types of assertions;
Example : The assertion **"x > 1"**, where **x** is a variable, is not a proposition because it is neither true nor false unless value of **x** is defined.
For **x > 1** to be a proposition ,
− either we substitute a specific number for **x** ;
− or change it to something like
"There is a number x for which x > 1 holds";
− or **"For every number x, x > 1 holds"**.
Consider example :

" All men are mortal.
Socrates is a man.
Then Socrates is mortal" ,

These cannot be expressed in propositional logic as a finite and logically valid argument (formula).
We need languages : that allow us to describe properties (*predicates*) of objects, or a relationship among objects represented by the variables .
Predicate logic satisfies the requirements of a language.
− *Predicate logic* is powerful enough for expression and reasoning.
− *Predicate logic* is built upon the ideas of *propositional logic.*

Predicate :

Every complete "sentence" contains two parts : a "subject" and a "predicate".
The *subject* is what (or whom) the sentence is about.
The *predicate* tells something about the subject;

Example :

A *sentence* **"Judy {runs}".**
The subject is **Judy** and the predicate is **runs** .
Predicate, always includes verb, tells something about the subject.

Predicate is a verb phrase template that describes a property of objects, or a relation among objects represented by the variables.

Example:

"The car Tom is driving *is blue*"; "The sky *is blue*" ;"The cover of this book *is blue*"
Predicate is **"is blue"** , describes property.
Predicates are given names; Let **„B"** is name for predicate **"is_blue"**.
Sentence is represented as **"B(x)"** , read as **"x is blue"**;

Symbol **"x"** represents an arbitrary Object .

4. Comparison between procedural and declarative language?

Comparison between Procedural and Declarative Language :

Procedural Language	Declarative Language
• Basic, C++, Cobol, etc.	• SQL
• Most work is done by interpreter of the languages	• Most work done by Data Engine within the DBMS
• For one task many lines of code	• For one task one SQL statement
• Programmer must be skilled in translating the objective into lines of procedural code	• Programmer must be skilled in clearly stating the objective as a SQL statement
• Requires minimum of management around the actual data	• Relies on SQL-enabled DBMS to hold the data and execute the SQL statement .
• Programmer understands and has access to each step of the code	• Programmer has no interaction with the execution of the SQL statement
• Data exposed to programmer during execution of the code	• Programmer receives data at end as an entire set
• More susceptible to failure due to changes in the data structure	• More resistant to changes in the data structure
• Traditionally faster, but that is changing	• Originally slower, but now setting speed records
• Code of procedure tightly linked to front end	• Same SQL statements will work with most front ends Code loosely linked to front end.
• Code tightly integrated with structure of the data store	• Code loosely linked to structure of data; DBMS handles structural issues
• Programmer works with a pointer or cursor	• Programmer not concerned with positioning
• Knowledge of coding tricks applies only to one language	• Knowledge of SQL tricks applies to any language using SQL