AI Question Bank for Test2

Module 3.

1. Define Logical agents and explain how the knowledge can be represented using sematic networks and Propositional Logic.
2. List and explain the Key Components of Knowledge base.
3. Define Logical Reasoning and explain Syllogism with example. Discuss Key Components of Knowledge based agents.
4. Write and explain the function for Knowledge based agents.
5. Describe Wumpus World and its PEAS description
6. What is Logic? What are the different types of Logic? Give Examples.
7. Give the formal definition of Logical entailment and explain with Wumpus World as example.
8. Discuss the following with suitable examples:
    a. Logical Entailment
    b. Logical Inference
    c. Model Checking
    d. Sound or Truth Preserving
    e. Completeness
    f. Correspondence between world and representation
    g. Grounding
9. Discuss the following with suitable examples:
    a. Key elements and components of Propositional logic
    b. Syntax and BNF Grammar of Propositional logic
    c. Semantics and Rules for Atomic and complex sentences in Propositional logic
10. Express the simple Knowledge base of Wumpus World using syntax and semantics of Propositional logic

Module 4.

1. **What is First Order Logic (FOL)? Explain basic elements of FOL with examples.**
2. **Explain the following with examples:**
   a. The syntax of first-order logic with equality, specified in Backus–Naur form
   b. Syntax and Semantics of FOL
3. **For each of the following English sentences, write a corresponding sentence in FOL.**
   a. The only good extraterrestrial is a drunk extraterrestrial.
   b. The Barber of Seville shaves all men who do not shave themselves.
   c. There are at least two mountains in England.
   d. There is exactly one coin in the box.
   e. There are exactly two coins in the box.
   f. The largest coin in the box is a quarter.
   g. No mountain is higher than itself.
   h. All students get good grades if they study.

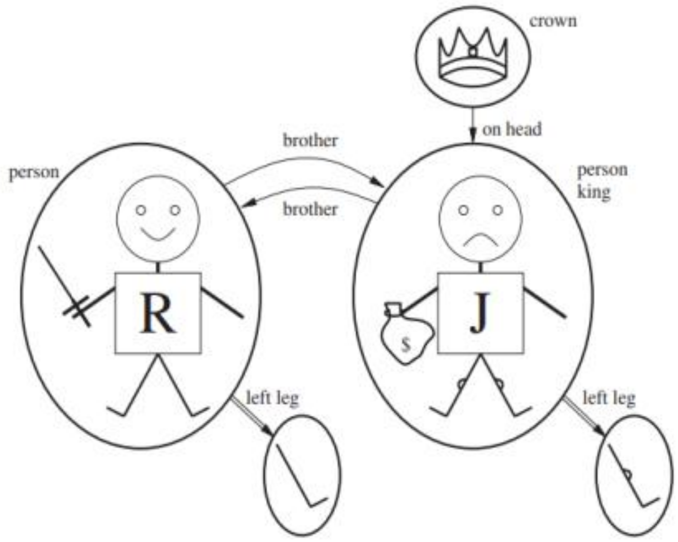
Ans:

First Order Logic Sentences

For each of the following English sentences, write a corresponding sentence in FOL.

1. The only good extraterrestrial is a drunk extraterrestrial.
   $\forall x.ET(x) \wedge Good(x) \rightarrow Drunk(x)$
2. The Barber of Seville shaves all men who do not shave themselves.
   $\forall x.\neg Shaves(x, x) \rightarrow Shaves(BarberOfSeville, x)$
3. There are at least two mountains in England.
   $\exists x, y.Mountain(x) \wedge Mountain(y) \wedge InEngland(x) \wedge InEngland(y) \wedge x \neq y$
4. There is exactly one coin in the box.
   $\exists x.Coin(x) \wedge InBox(x) \wedge \forall y.(Coin(y) \wedge InBox(y) \rightarrow x = y)$
5. There are exactly two coins in the box.
   $\exists x, y.Coin(x) \wedge InBox(x) \wedge Coin(y) \wedge InBox(y) \wedge x \neq y \wedge \forall z.(Coin(z) \wedge InBox(z) \rightarrow (x = z \vee y = z))$
6. The largest coin in the box is a quarter.
   $\exists x.Coin(x) \wedge InBox(x) \wedge Quarter(x) \wedge \forall y.(Coin(y) \wedge InBox(y) \wedge \neg Quarter(y) \rightarrow Smaller(y, x))$
7. No mountain is higher than itself.
   $\forall x.Mountain(x) \rightarrow \neg Higher(x, x)$
8. All students get good grades if they study.
   $\forall x.Student(x) \wedge Study(x) \rightarrow GetGoodGrade(x)$

4. List the properties and draw backs of Propositional Language
5. With example explain the following components of FOL
   a. Objects
   b. Relations
   c. Functions
6. Compare different types of languages
7. Discuss the following components of Syntax and Semantics of FOL
   a. Models for FOL
   b. Symbols and Interpretations
   c. Terms
   d. Atomic Sentences
   e. Complex Sentence
   f. Quantifiers
   g. Equality
   h. Data base Semantics
8. List all Objects, Tuples, Relations and Functions in the following Model:



9. Write the syntax of FOL with equality specified in Backus Naur form.
10. Give examples for Nested Quantifiers
11. Write DeMorgans Rules for quantified and unquantified sentences.
12. Write a FOL statements for the following :
    a. One’s mother is one’s female parent: ∀ m,c Mother (c)= m ⇔ Female(m) ∧ Parent(m,c) .
    b. One’s husband is one’s male spouse: ∀ w,h Husband(h,w) ⇔ Male(h) ∧ Spouse(h,w)

c. Male and female are disjoint categories: ∀ x Male(x) ⇔ ¬Female(x) .
d. Parent and child are inverse relations: ∀ p,c Parent(p,c) ⇔ Child(c,p)
e. A grandparent is a parent of one's parent: ∀ g,c Grandparent(g,c) ⇔ ∃ p Parent(g,p) ∧ Parent(p,c) .
f. A sibling is another child of one's parents: ∀ x,y Sibling(x,y) ⇔ x != y ∧ ∃ p Parent(p,x) ∧ Parent(p,y)
g.

13. Discuss the Following with examples
   a. Assertions and queries in first-order logic
   b. Representing Numbers and List in FOL
   c. Representing Sets in FOL
   d. Representing Wumpus World using FOL

14. What is Knowledge Engineering? Explain the different steps of Knowledge Engineering projects by considering the Electronic Circuit (of 1-bit full adder) with example.

15. What is Modus Ponens? Give Example.

16. Compare Propositional Logic with First Order logic with examples

17. Discuss / Illustrate the following with examples :
   a. The rule of Universal Instantiation
   b. The rule of Existential Instantiation
   c. Unification
   d. Unification Algorithm
   e. Storage and Retrieval in FOL
   f. Rules for Unification in FOL
   g. Forward Chaining
   h. Forward Chaining Algorithm

18. Consider the following problem: The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American. Using FOL Forward Chaining prove that West is Criminal/

19. Unify the following two predicates using Unification Algorithm
   a. Predicate P(x,y)
   b. Predicate Q(f(z),a)

20.

Module 5

1. Write an Algorithm and explain with example for the following of FOL
   c. Backward Chaining in FOL
   d. Resolution in FOL
2. Consider the following problem: The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American. Using FOL Forward Chaining prove that West is Criminal/
3. Explain the Steps involved in Resolution of FOL statements with examples.
4. State and Explain the following
   e. Herbrands Theorem
   f.
5. Convert the following sentences to conjunctive normal form.
   a. (A → B) → C

      Answer:

      ¬(¬A ∨ B) ∨ C (A ∧ ¬B) ∨ C (A ∨ C) ∧ (¬B ∨ C)
   b. A → (B → C)

      Answer:

      ¬A ∨ ¬B ∨ C
   c. (A → B) ∨ (B → A)

      Answer:

   (¬A ∨ B) ∨ (¬B ∨ A)
   d. 4. (¬P → (P → Q))

      Answer:

      ¬¬P ∨ (¬P ∨ Q) P ∨ ¬P ∨ Q
   e. (P → (Q → R)) → (P → (R → Q))

      Answer:

      ¬(¬P ∨ ¬Q ∨ R) ∨ (¬P ∨ ¬R ∨ Q)

      (P ∧ Q ∧ ¬R) ∨ (¬P ∨ ¬R ∨ Q)

      (P ∨ ¬P ∨ ¬R ∨ Q) ∧ (Q ∨ ¬P ∨ ¬R ∨ Q) ∧ (¬R ∨ ¬P ∨ ¬R ∨ Q)

      ¬P ∨ Q ∨ ¬R

6. Explain the following with examples

a. Algorithm To convert a given formula to CNF
b. Algorithm/Steps To prove by resolution
c. Horn Clauses and Definite Clauses
d. Herbrands Theorem
e. Ground Term and Herbrands Universe
f. Lifting Lemma

7. **Apply resolution steps and prove the validity ¬P(a)** for the following Knowledge base:
   1. P(a)∨Q(b)
   2. ¬Q(b)∨R(c)
   3. ¬R(c)

8. Discuss with examples how equality is handled in First order Logic
9. Define Classical Planning? What are the limitations of earlier approach and how it is overcome in Classical Planning discuss with examples?
10. What is the closed-world assumption in classical planning?
11. Provide an example of how states are represented as conjunctions of fluents.
12. What is the purpose of Planning Domain Definition Language (PDDL)?
13. Design the classical planning with PDDL description (Schema) and solution plan for the following examples
    a. Air Cargo Transport
    b. The Spare Tire Problem
    c. The Blocks World
14. Discuss the complexity of Classical Planning with reference to PlanSAT and bounded PlanSAT
15. With suitable examples compare the Forward State Space Search with Backward space Search.
16. Discuss and Apply the ignore-delete-lists heuristic to simplify problem with suitable examples.
17. Define Planning graph and construct planning graph for the problem Have cake and Eat Cake problem
18. Design and Descibe GraphPlan Algorithm
19. Apply the GRAPHPLAN algorithm to extract a solution for the spare tyre problem.
20. Discuss how planning graphs aids heuristic estimation.
21. Design and illustrate GraphPlan Algorithm with examples.

22. Discuss the termination of GraphPlan Algorithm.

23. Define logic programming? What is Prolog. Give Examples for Prolog Programs.

24. Discuss how Prolog executes with example.What are the Key features of Prolog Execution.

25. Discuss the Efficient implementation of Logic Programs with examples.

26. Discuss the following with suitable examples :

a. Parallelism and Dynamic Programming in Prolog
b. Redundant and Inininte inference loops in Prolog
c. Database semantics of Prolog
d. Constraint Logic Programming

Additional Questions to Practice on Classical Planning

RBT Level 1 (Remembering) – 20 Questions

1. Define **Classical Planning**.
2. What is the **Planning Domain Definition Language (PDDL)** used for in classical planning?
3. How are states represented in classical planning?
4. What does the **closed-world assumption** imply?
5. What is the **unique names assumption** in classical planning?
6. Give an example of a fluent representing a state.
7. What are **non-ground fluents**? Provide an example.
8. Why are **negations** not allowed in state representation in classical planning?
9. Explain the role of **action schemas** in planning.
10. What is the purpose of preconditions in an action schema?
11. What are the effects of an action schema?
12. Provide an example of an action schema for flying a plane.
13. What is a **planning domain**?
14. What is a **planning problem**?
15. Give an example of a **planning problem** and its solution.
16. List three limitations of early classical planning approaches.
17. What is the purpose of **factored representations** in classical planning?
18. Define the **PlanSAT** decision problem.
19. What is the difference between **PlanSAT** and **Bounded PlanSAT**?
20. Describe the basic setup of the **blocks world** problem.

RBT Level 2 (Understanding) – 10 Questions

21. Why is **PDDL** preferred for describing actions and states in classical planning?
22. Explain the **delete list** and its role in determining the effect of an action.
23. Why are **structured state representations** more efficient than atomic state representations?
24. How does classical planning address the combinatorial explosion of actions in large domains?
25. Differentiate between **progression search** and **regression search** in state-space search.
26. What are the **types of heuristics** used in classical planning, and why are they important?
27. Explain the concept of **mutex links** in planning graphs.
28. How does the **GRAPHPLAN algorithm** ensure termination?
29. Describe the **spare tire problem** and its solution in classical planning.
30. Why are spurious actions problematic in classical planning, and how can they be avoided?

RBT Level 3 (Applying) – 10 Questions

31. Represent the state At(Truck1,Melbourne)∧At(Truck2,Sydney) as a conjunction of fluents.
32. Create an action schema for loading a cargo onto a truck.
33. Using the air cargo domain, plan a sequence of actions to move Cargo1 from SFO to JFK using Plane1.
34. Write an action schema for **MoveToTable(b, x)** in the blocks world problem.
35. Apply the **ignore-delete-lists heuristic** to simplify the **spare tire problem**.
36. Construct a planning graph for a simple transportation problem where a truck needs to move from Location A to B and back.
37. Use regression search to derive the preconditions for unloading a cargo at a destination airport.
38. Solve a simplified version of the **blocks world problem**: Move Block A from Table to Block B.
39. Show how inequality preconditions can resolve a spurious action Fly(P1,JFK,JFK).
40. Apply the **GRAPHPLAN algorithm** to extract a solution for the air cargo transport problem involving two cargos and two airports.

RBT Level 4 (Analyzing) – 10 Questions

41. Analyze the complexity of classical planning for a domain with 3 planes, 5 cargos, and 10 airports.
42. Evaluate the advantages of **heuristic-based planning** over brute-force state-space search.
43. Discuss the monotonic properties of planning graphs and their importance in the GRAPHPLAN algorithm.
44. Explain how **level-cost heuristics** estimate the cost of achieving a goal in planning graphs.
45. Analyze the scalability challenges of atomic state representations in dynamic environments.
46. Investigate how the use of **mutex links** helps prevent conflicts in planning graphs.
47. Explain the computational challenges of using **domain-independent heuristics** in classical planning.
48. Compare the efficiency of **progression search** and **regression search** in solving complex planning problems.
49. Explore the trade-offs between **PlanSAT** and **Bounded PlanSAT** for practical applications.
50. Analyze how PDDL enables concise and reusable action descriptions across multiple domains.

Additional Topic Wise Questions on Logic Programming

1. Introduction to Logic Programming

- **Level 1**: Define logic programming.
- **Level 1**: What is Robert Kowalski's principle of algorithm design?
- **Level 2**: Explain how logic programming differs from procedural programming.
- **Level 3**: Discuss the advantages and limitations of using logic programming for AI applications.

---

2. Prolog Basics

- **Level 1**: What is Prolog, and what are its primary use cases?
- **Level 1**: How are variables and constants represented in Prolog?
- **Level 2**: Write a Prolog program to find the factorial of a number.
- **Level 3**: Compare the declarative nature of Prolog with imperative programming languages like Python or Java.

---

3. Prolog Syntax and Rules

- **Level 1**: Describe the clause structure in Prolog with an example.
- **Level 1**: What does the Prolog predicate `append([], Y, Y)` mean?
- **Level 2**: Explain the role of recursive definitions in Prolog programs using the `append` predicate.
- **Level 3**: Design a Prolog rule to identify prime numbers and discuss its efficiency.

---

4. Execution Mechanism in Prolog

- **Level 1**: What is depth-first backward chaining in Prolog?
- **Level 1**: How does Prolog handle arithmetic operations?
- **Level 2**: Analyze the implications of Prolog's depth-first search on query performance.
- **Level 3**: Propose improvements to Prolog's execution strategy to handle large-scale reasoning problems efficiently.

---

5. Efficient Implementation of Logic Programs

- **Level 1**: What are the two modes in which Prolog programs can run?
- **Level 1**: What is the purpose of the Warren Abstract Machine (WAM)?
- **Level 2**: How do choice points and continuations improve Prolog's efficiency?
- **Level 3**: Discuss how parallelization techniques like OR-parallelism and AND-parallelism can enhance Prolog's performance.

---

6. Handling Redundancy and Infinite Loops in Prolog

- **Level 1**: What causes infinite loops in Prolog programs?
- **Level 1**: How does Prolog handle redundant computations?
- **Level 2**: Compare Prolog's depth-first backward chaining with forward chaining in terms of redundancy.
- **Level 3**: Evaluate the role of memoization and Tabled Logic Programming in addressing Prolog's inefficiencies.

---

7. Database Semantics of Prolog

- **Level 1**: What is the closed-world assumption in Prolog?
- **Level 1**: How does Prolog's database semantics simplify reasoning compared to first-order logic?
- **Level 2**: Illustrate the unique names assumption with an example.
- **Level 3**: Critique the limitations of Prolog's database semantics in dynamic knowledge bases.

---

8. Constraint Logic Programming (CLP)

- **Level 1**: What is Constraint Logic Programming (CLP)?
- **Level 1**: How does CLP extend Prolog's capabilities?
- **Level 2**: Write a CLP rule to verify if a triangle can be formed from three given lengths.
- **Level 3**: Evaluate the effectiveness of CLP for solving constraint satisfaction problems like Sudoku or scheduling.

***9.* Applications of Prolog**

- **Level 1**: Name three areas where Prolog is commonly used.
- **Level 1**: How is Prolog used in expert systems?
- **Level 2**: Design a simple Prolog program for family tree queries.
- **Level 3**: Analyze the potential of Prolog for real-time AI tasks like speech recognition or robotics.

…Dr.Thyagaraju G S

9480123526