USN

Internal Assessment Test 2 – Dec 2024

Sub:	**Artificial Intelligence**					Sub Code:	**BCS515B**	Branch:	**CSE**
Date:	**17/12/2024**	Duration:	90 mins	Max Marks:	50	Sem/Sec:	**V/A,B&C**		OBE

Answer any FIVE FULL Questions

		MARKS	CO	RBT
1	a. Match the Statement with Propositional Logic Expressions **Statements** / **Propositional Logic** i) If it rains, then the ground is wet. — $\neg P$ ii) It is not raining. — $V \leftrightarrow P$ iii) It is sunny or it is cloudy. — $P \to Q$ iv) The light is on and the fan is off. — $R \vee S$ v) The train is late if and only if it is raining. — $T \wedge \neg U$ **Answer :** i) $P \to Q$ ii) $\neg P$ iii) $R \vee S$ iv) $T \wedge \neg U$ v) $V \leftrightarrow P$ (each correct match carrys 1 mark)	5	CO3	L2
	b. Explain the steps involved in the knowledge-engineering process. Answer: Short explanations of the 7 step process: 1. Identify the Task 2. Assemble Relevant Knowledge 3. Decide on Vocabulary 4. Encode General Knowledge 5. Encode Specific Problem Instances 6. Pose Queries and Get Answers **7.** Debug the Knowledge Base	5	CO3	L2
2	a. Convert the following English statements into FOL i) No mountain is higher than itself $\forall x(Mountain(x) \to \neg Higher(x,x))$ ii) There are atleast two mountains in India $\exists x \, \exists y \, (Mountain(x) \wedge Mountain(y) \wedge InIndia(x) \wedge InIndia(y) \wedge x \neq y)$ iii)There is exactly one coin in the box $\exists x \, (Coin(x) \wedge InBox(x) \wedge \forall y \, (Coin(y) \wedge InBox(y) \to y = x))$	5	CO3	L2

	iv) All students get good grades if they study $\forall x\,(Student(x) \wedge Studies(x) \rightarrow GoodGrades(x))$ v) All humans are mortal $\forall x\,(Human(x) \rightarrow Mortal(x))$			
	b.Convert the following FOL in to English statements i) ∃x ∃y (BankAccount(x) -> Deposit(y,x)) ii) $\forall x\,(Student(x) \rightarrow \exists y\,(Classroom(y) \wedge Enrolled(x,y)))$ iii) $\forall x\,(Teacher(x) \rightarrow \exists y\,(Subject(y) \wedge Teaches(x,y)))$ iv) $\exists x\,(Student(x) \wedge Plays(x, Football))$ v) $\forall x\,(Employee(x) \rightarrow \exists y\,\exists z\,(Company(y) \wedge Computer(z) \wedge WorksIn(x,y) \wedge Uses(x,z)))$ **Answer:** I) There exists an account x and a deposit y such that if x is a bank account, then y is deposited into x. II) For every exist x, if x is a student, then there exists a y such that y is a classroom and x is enrolled in y. III) For every x, if xx is a teacher, then there exists a y such that y is a subject and x teaches y. Iv) There exists an x such that xxx is a student and x plays football." Or : *"There is at least one student who plays football."* *v)* Every employee works in some company and uses some computer	5	CO3	L2

3	Consider the following problem: 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. a) Represent the above facts as first-order definite clauses.(3) b) Using Forward Chaining approach prove that **West is a criminal**.(3) c) Write the forward-chaining algorithm(4) Answer:a) 1. **Crime definition**: Selling weapons to hostile nations is a crime for Americans: $$American(x) \land Weapon(y) \land Hostile(z) \land Sells(x, y, z) \rightarrow Criminal(x)$$ 2. **Nono is hostile**: Nono is a hostile nation: $$Hostile(Nono)$$ 3. **Missiles are weapons**: All missiles are weapons: $$Missile(y) \rightarrow Weapon(y)$$ 4. **Nono has some missiles**: Nono has a missile called $M1$: $$Owns(Nono, M1) \land Missile(M1)$$ 5. **Colonel West sold missiles to Nono**: Colonel West sold $M1$ to Nono: $$Sells(West, M1, Nono)$$ 6. **West is an American**: Colonel West is American: $$American(West)$$ **b) Using Forward Chaining to Prove West is a Criminal** **Goal**: Prove that **West is a criminal** using Forward Chaining. 1. **Initial facts** (given or inferred directly): • $Hostile(Nono)$ • $Owns(Nono, M1)$ • $Missile(M1)$ • $American(West)$ • $Sells(West, M1, Nono)$ 2. **Rules to use**: • $Missile(y) \rightarrow Weapon(y)$ • $American(x) \land Weapon(y) \land Hostile(z) \land Sells(x, y, z) \rightarrow Criminal(x)$	10	CO4	L3

Forward Chaining Steps:

1. Start with $Missile(M1)$.
   Using the rule $Missile(y) \rightarrow Weapon(y)$, infer:

$$Weapon(M1)$$

2. Now we know:
   - $American(West)$
   - $Weapon(M1)$
   - $Hostile(Nono)$
   - $Sells(West, M1, Nono)$
3. Apply the rule:
   $American(x) \land Weapon(y) \land Hostile(z) \land Sells(x, y, z) \rightarrow Criminal(x)$.

Substituting $x = West$, $y = M1$, $z = Nono$:

- $American(West)$ is true.
- $Weapon(M1)$ is true.
- $Hostile(Nono)$ is true.
- $Sells(West, M1, Nono)$ is true.

Therefore, we can infer:

$$Criminal(West)$$

C) Forward Chaining Algorithm

```
function FOL-FC-ASK(KB, α) returns a substitution or false
  inputs: KB, the knowledge base, a set of first-order definite clauses
          α, the query, an atomic sentence
  local variables: new, the new sentences inferred on each iteration

  repeat until new is empty
     new ← { }
     for each rule in KB do
        (p1 ∧ . . . ∧ pn ⇒ q) ← STANDARDIZE-VARIABLES(rule)
        for each θ such that SUBST(θ, p1 ∧ . . . ∧ pn) = SUBST(θ, p'1 ∧ . . . ∧ p'n)
                    for some p'1, . . . , p'n in KB
           q' ← SUBST(θ, q)
           if q' does not unify with some sentence already in KB or new then
              add q' to new
              φ ← UNIFY(q', α)
              if φ is not fail then return φ
     add new to KB
  return false
```

4	a. Write the limitations of Propositional logic	5	CO3	L2

Answer: (ant 5 points- each point carrys 1 mark)

1. We cannot represent relations like ALL, some, or none with propositional logic. Example:All the girls are intelligent.
   Some apples are sweet.

2. Propositional logic has limited expressive power.
3. In propositional logic, we cannot describe statements in terms of their properties or logical relationships.
4. Scalability issues
5. No support for representing uncertainty
6. Uses static representation

b. Compare forward chaining with backward chaining reasoning approach — 5 — CO4 — L2

Answer:

Aspect	Backward Chaining	Forward Chaining
Direction	Starts from the goal and works backward.	Starts from known facts and works forward.
Goal-Oriented	Yes, only focuses on proving the specific goal.	No, it generates all possible inferences.
Efficiency	Generally more efficient for specific goals.	Can be inefficient when many facts are processed.
Use Case	Diagnosis, planning, expert systems.	Data analysis, rule-based systems.

5 — 10 — CO4 — L3

Everyone who loves all animals is loved by someone.
Anyone who kills an animal is loved by no one.
Jack loves all animals.
Either Jack or Curiosity killed the cat, who is named Tuna.

Prove that the Curiosity kill the cat using Resolution.

Write all the steps for the proof by resolution.(10)

Answer:

First, we express the original sentences, some background knowledge, and the negated goal G in first-order logic:

A. $\forall x\ [\forall y\ Animal(y) \Rightarrow Loves(x,y)] \Rightarrow [\exists y\ Loves(y,x)]$

B. $\forall x\ [\exists z\ Animal(z) \land Kills(x,z)] \Rightarrow [\forall y\ \neg Loves(y,x)]$

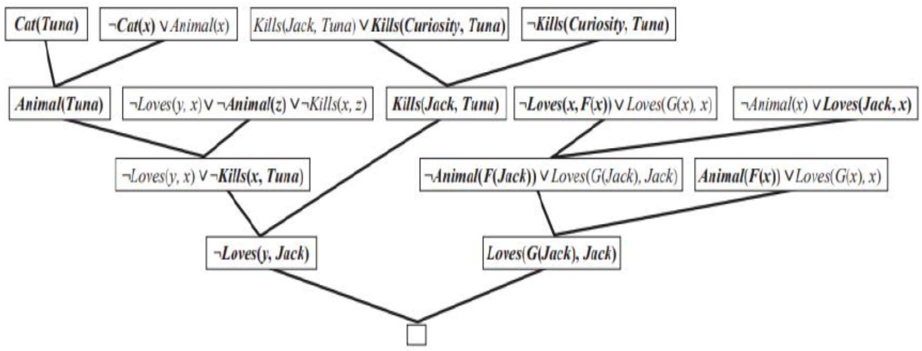
C. $\forall x\ Animal(x) \Rightarrow Loves(Jack,x)$

D. $Kills(Jack,Tuna) \lor Kills(Curiosity,Tuna)$

E. $Cat(Tuna)$

F. $\forall x\ Cat(x) \Rightarrow Animal(x)$

¬G. $\neg Kills(Curiosity,Tuna)$

	Now we apply the conversion procedure to convert each sentence to CNF: A1. $Animal(F(x)) \vee Loves(G(x), x)$ A2. $\neg Loves(x, F(x)) \vee Loves(G(x), x)$ B. $\neg Loves(y, x) \vee \neg Animal(z) \vee \neg Kills(x, z)$ C. $\neg Animal(x) \vee Loves(Jack, x)$ D. $Kills(Jack, Tuna) \vee Kills(Curiosity, Tuna)$ E. $Cat(Tuna)$ F. $\neg Cat(x) \vee Animal(x)$ $\neg$G. $\neg Kills(Curiosity, Tuna)$ Suppose Curiosity did not **kill Tuna**. We know that either Jack or Curiosity did; thus Jack must have. Now, Tuna is a cat and cats are animals, so Tuna is an animal. Because anyone who kills an animal is loved by no one, we know that no one loves Jack. On the other hand, Jack loves all animals, so someone loves him; so we have a contradiction. Therefore, Curiosity killed the cat. 			
6	Let's use the following **Knowledge Base (KB)** and query. **Knowledge Base:** 1. `Parent(x, y) ⇒ Ancestor(x, y)` (Rule 1) 2. `Parent(x, z) ∧ Ancestor(z, y) ⇒ Ancestor(x, y)` (Rule 2) 3. `Parent(John, Mary)` (Fact) 4. `Parent(Mary, Sam)` (Fact) **Query: `Ancestor(John, Sam)`** Use backward chaining algorithm to get the reasoning for the given query. Answer:	10	CO4	L3

Execution Steps

Step 1: FOL-BC-ASK

- **Query:** Ancestor(John, Sam)
- **Calls:** FOL-BC-OR(KB, Ancestor(John, Sam), { }).

Step 2: FOL-BC-OR

- **Goal:** Ancestor(John, Sam)
- **Fetch rules for** Ancestor:
  1. **Rule 1:** Parent(x, y) ⇒ Ancestor(x, y)
  2. **Rule 2:** Parent(x, z) ∧ Ancestor(z, y) ⇒ Ancestor(x, y)

Case 1: Use Rule 1

- lhs = Parent(x, y), rhs = Ancestor(x, y).
- **Unify** Ancestor(John, Sam) **with** Ancestor(x, y):
  - **Substitution:** θ = {x=John, y=Sam}.
- **Sub-goal:** Parent(John, Sam).

Step 3: FOL-BC-AND

- **Goals:** [Parent(John, Sam)]
- **Calls:** FOL-BC-OR(KB, Parent(John, Sam), {x=John, y=Sam}).

Step 4: FOL-BC-OR

- **Goal:** Parent(John, Sam)
- **Check the KB:**
  - **Facts:** Parent(John, Mary) **(no match for** Sam**).**
  - **Rule 1 fails.**

Case 2: Use Rule 2

- lhs = Parent(x, z) ∧ Ancestor(z, y), rhs = Ancestor(x, y).
- **Unify** Ancestor(John, Sam) **with** Ancestor(x, y):
  - **Substitution:** θ = {x=John, y=Sam}.
- **Sub-goals:**
  - goals = [Parent(John, z), Ancestor(z, Sam)].

Step 5: FOL-BC-AND

- **Goals:** [Parent(John, z), Ancestor(z, Sam)].

1. **First sub-goal (Parent(John, z)):**
   - **Calls:** FOL-BC-OR(KB, Parent(John, z), θ).
   - **Matches:** Parent(John, Mary).
   - **Substitution:** {z=Mary}.
   - **Update θ':** {x=John, y=Sam, z=Mary}.

2. **Second sub-goal (`Ancestor(z, Sam)`):**
   - Calls: `FOL-BC-OR(KB, Ancestor(Mary, Sam), θ')`.
   - Unify with Rule 1: `Parent(x, y) ⇒ Ancestor(x, y)`.
   - Sub-goal: `Parent(Mary, Sam)`.

Step 6: FOL-BC-AND

- Goal: `[Parent(Mary, Sam)]`.
- Matches fact: `Parent(Mary, Sam)`.
- Substitution: `{x=Mary, y=Sam}`.
- Satisfies all sub-goals.

Final Result

- Combine all substitutions:
  - `{x=John, y=Sam, z=Mary}`.
- The query `Ancestor(John, Sam)` is **true.**

CI CCI HoD
