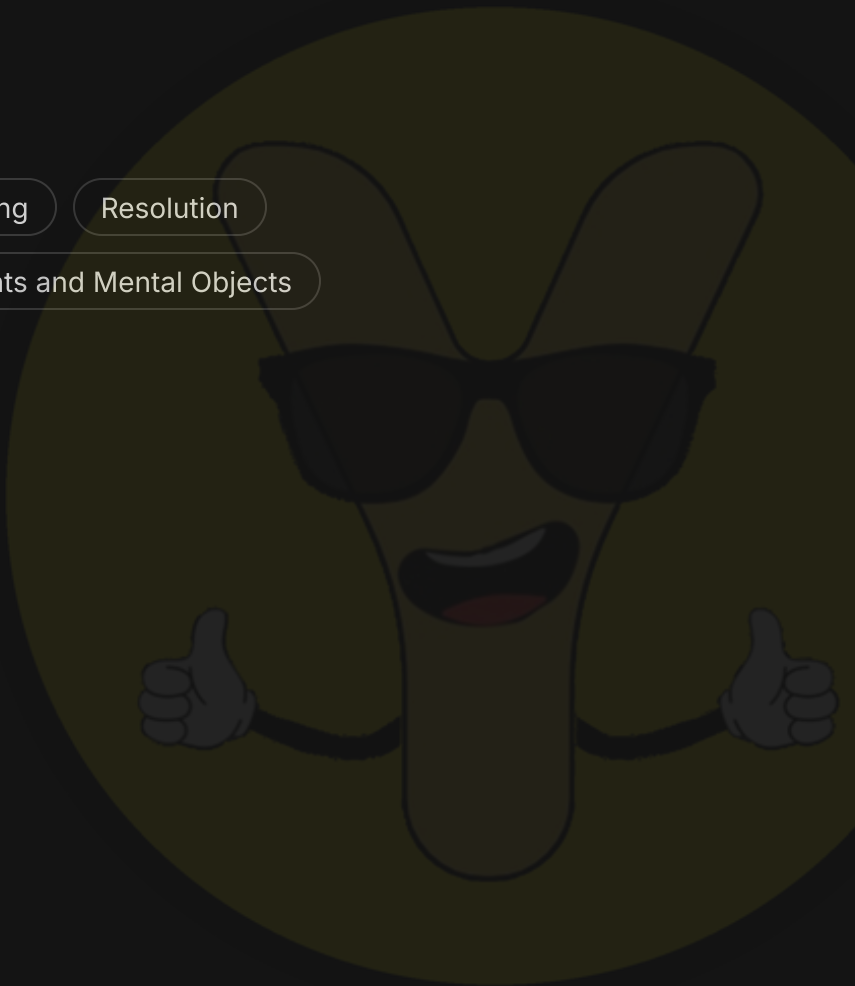
AKTU · 3RD YEAR · SEMESTER 5 · BCAI501

Artificial Intelligence

Unit 3 — Knowledge Representation

- First Order Predicate Logic
- Prolog Programming
- Unification
- Forward Chaining
- Backward Chaining
- Resolution
- Knowledge Representation
- Ontological Engineering
- Categories and Objects
- Events
- Mental Events and Mental Objects
- Reasoning Systems for Categories
- Reasoning with Default Information



DECODED BY

Chirag Jain

Founder, Ycotes · Gold Medalist in Artificial Intelligence, IIT Madras · taught 10,000+ students

First Order Predicate Logic (FOPL)

FIRST ORDER PREDICATE LOGIC

A formal symbolic logic system that extends propositional logic by introducing predicates, functions, constants, variables, and quantifiers to represent complex world knowledge.

- Objects: Represent domain entities, e.g., $\mathrm{John}, 5, \mathrm{India}$.
- Predicates: Represent relations or properties of objects, e.g., $\mathrm{IsBrother}(X, Y)$ or $\mathrm{IsGreen}(X)$.
- Functions: Map objects to another object, e.g., $\mathrm{FatherOf}(X)$.
- Quantifiers: Universal $\forall$ ('for all') and Existential $\exists$ ('there exists').

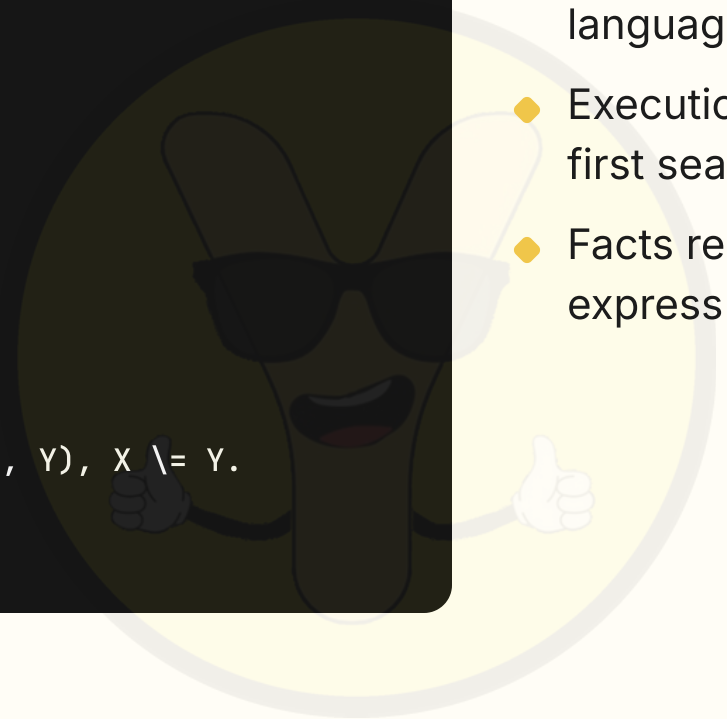
Prolog Programming Basics

```
% Prolog Knowledge Base

% Facts
parent(john, mary).
parent(john, tom).
male(john).
female(mary).

% Rules
father(X, Y) :- parent(X, Y), male(X).
sibling(X, Y) :- parent(Z, X), parent(Z, Y), X \= Y.

% Query example: ?- father(john, mary).
```



- Prolog is a declarative logic programming language based on Horn clauses.
- Execution relies on backward chaining, depth-first search, and automatic backtracking.
- Facts represent static assertions; rules express conditional logical inference.

Unification Algorithm

UNIFICATION

An algorithmic procedure in automated theorem proving that finds a substitution θ making two first-order logic expressions syntactically identical.

- Substitution: Mapping set of variables to terms, written as $\theta = \{X/a, Y/f(b)\}$.
- Most General Unifier (MGU): The minimal variable-binding substitution that unifies two terms.
- Occurs Check: Verifies a variable does not occur within its substituting term, preventing infinite structures.

Unification Worked Example

Find the Most General Unifier (MGU) for the expressions $P(X, f(Y))$ and $P(a, f(g(z)))$.

1. Match predicate symbols: Both use P with arity 2. Proceeds to arguments.
2. Compare 1st argument pair (X, a): Bind variable X to constant a, yields $\theta_1 = \{X/a\}$.
3. Compare 2nd argument pair $(f(Y), f(g(z)))$: Matching functor f, proceed to inner arguments $(Y, g(z))$.
4. Bind variable Y to term $g(z)$, yields $\theta_2 = \{Y/g(z)\}$.
5. Combine substitution sets to obtain MGU: $\theta = \{X/a, Y/g(z)\}$.

MGU $\theta = \{X/a, Y/g(z)\}$, resulting in unified expression $P(a, f(g(z)))$.

Forward Chaining Process



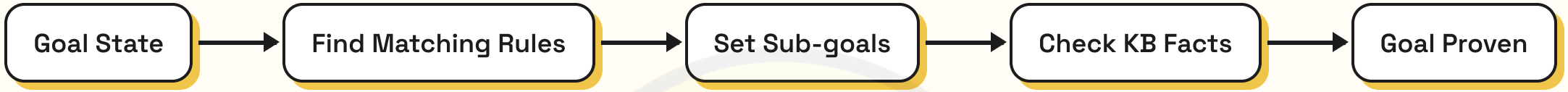
Data-driven forward chaining execution flow starting from facts to infer goal.

Forward Chaining Inference Strategy

Forward chaining is a data-driven reasoning strategy working forward from known facts to derive new conclusions.

- Working Principle
  - Starts with known atomic facts in the Knowledge Base (KB).
  - Evaluates rule premises matching known facts in data-driven manner.
  - Fires active rules and adds conclusions into KB as new facts.
- Key Advantages
  - Efficiently derives all possible conclusions from available input data.
  - Ideal for dynamic monitoring, control systems, and synthesis tasks.
- Disadvantages
  - May derive many irrelevant facts unrelated to the target goal.

Backward Chaining Strategy



Goal-driven backward chaining flow recursively resolving sub-goals.

Backward Chaining Inference Strategy

Backward chaining is a goal-driven reasoning strategy used in Prolog and diagnostic engines.

- Working Principle
  - Starts with a target hypothesis/goal to be proven.
  - Searches KB for rules whose consequents (THEN-part) match the goal.
  - Extracts rule antecedents (IF-part) as new sub-goals to prove recursively.
- Key Advantages
  - Focuses computation strictly on facts relevant to target hypothesis.
  - Efficient when target goal is explicitly defined.
- Disadvantages
  - Can explore redundant proof branches without proper memoization.

Comparison: Forward vs Backward Chaining

Feature	Forward Chaining	Backward Chaining
Strategy	Data-driven (Bottom-Up)	Goal-driven (Top-Down)
Start Point	Given initial facts in KB	Target goal or hypothesis
Direction	Premises to conclusions	Goals to sub-goals and facts
Execution Style	Fires all matching rules	Recursively satisfies sub-goals
Applications	Control systems, monitoring	Diagnostic tools, Prolog, parsing
Efficiency	May derive unnecessary facts	Focuses only on goal-relevant facts

Resolution & CNF Conversion

Resolution is a sound and refutation-complete inference rule operating on Conjunctive Normal Form (CNF).

- Resolution Rule
  - From clauses $A \lor B$ and $\neg A \lor C$, infer resolvent $B \lor C$.
- CNF Conversion Steps
  - 1. Eliminate implications: $P \to Q \equiv \neg P \lor Q$.
  - 2. Move negations inward using De Morgan laws.
  - 3. Standardize variables (rename overlapping scope).
  - 4. Skolemize: Replace existential quantifiers $\exists$ with Skolem constants/functions.
  - 5. Drop universal quantifiers $\forall$ and distribute $\lor$ over $\land$.

Resolution Refinement Example

Prove $\text{Likes}(John, Peanut)$ given: 1. $\forall x\, \text{Food}(x) \rightarrow \text{Likes}(John, x)$, 2. $\text{Food}(Peanut)$.

1. Negate goal to prove: $\neg\text{Likes}(John, Peanut)$ (Clause 1).
2. Convert KB Axiom 1 to CNF: $\neg\text{Food}(x) \vee \text{Likes}(John, x)$ (Clause 2).
3. KB Axiom 2 in CNF: $\text{Food}(Peanut)$ (Clause 3).
4. Resolve Clause 1 and Clause 2 with $\theta = \{x/Peanut\}$: yields $\neg\text{Food}(Peanut)$ (Clause 4).
5. Resolve Clause 4 and Clause 3: yields empty clause $\square$ (Contradiction).

Contradiction $\square$ reached; hence, goal $\text{Likes}(John, Peanut)$ is proven TRUE.

Knowledge Representation (KR) Approaches

Knowledge representation formalizes domain facts into computational structures for automated reasoning.

- KR Requirements
  - Representational Adequacy: Ability to represent all necessary domain knowledge.
  - Inferential Adequacy: Ability to manipulate structures to infer new knowledge.
  - Inferential Efficiency: Ability to direct inference mechanisms towards solutions.
- Major KR Schemes
  - Logical KR: Propositional and First Order Predicate Logic.
  - Network KR: Semantic Networks and Conceptual Graphs.
  - Frame-based KR: Structured frames with attributes and inheritance.
  - Rule-based KR: Production rules in IF-THEN structure.

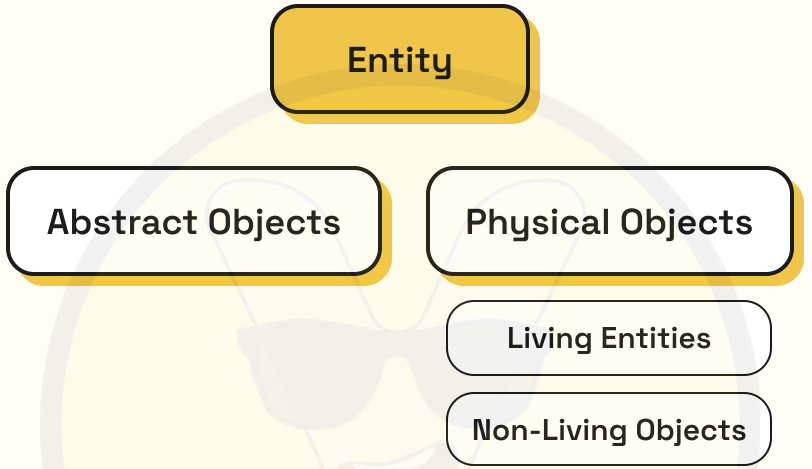
Ontological Engineering

ONTOLOGICAL ENGINEERING

The subfield of AI that studies methods for constructing, organizing, and maintaining explicit formal specifications of domain concepts and relationships (ontologies).

- Upper Ontology: Defines generic, high-level concepts like Objects, Events, Time, Space, and Actions.
- Domain Ontology: Models specialized knowledge unique to specific fields like Medicine or Finance.
- Enables interoperability, knowledge reuse, and standardized communication between heterogeneous AI systems.

Categories and Objects Hierarchy



Taxonomic categorization tree supporting inheritance of properties down object hierarchies.

Representing Events and Time

Dynamic domains require explicit formalisms to represent actions, events, and temporal changes over time.

- Situation Calculus
  - Represents continuous world states as discrete sequences of situations.
  - Fluents: State-dependent relations like $\mathrm{At}(Robot, Location, s)$.
  - Actions: Map situation s to new situation $\mathrm{Result}(a, s)$.
- Event Calculus
  - Time-point based representation of events and changing fluents.
  - $\mathrm{Initiates}(e, f, t)$: Event e causes fluent f to become true at time t.
  - $\mathrm{Terminates}(e, f, t)$: Event e terminates fluent f at time t.

Mental Events and Mental Objects

MENTAL EVENTS AND OBJECTS

Formal representations of internal cognitive states, beliefs, desires, and intentions held by intelligent agents regarding facts or propositions.

- Mental Objects: Abstract internal concepts, propositions, or mental sentences (e.g., belief in 'Raining').
- Mental Events: Internal cognitive processing events such as thinking, deciding, or believing.
- Propositional Attitudes: Represented using modal operators or reification: $\mathrm{Believes}(\mathrm{Agent}_A, \mathrm{IsRaining})$.
- Referential Opacity: Expressing contexts where equal terms cannot be substituted without altering truth value.

Reasoning Systems for Categories

Specialized reasoning frameworks organize category taxonomies and speed up logical subsumption.

- Semantic Networks
  - Graph structures consisting of nodes (objects/categories) connected by directed labeled edges.
  - Inheritance: Sub-classes automatically inherit attributes from super-classes via `is-a` links.
- Description Logics (DL)
  - Decidable formal fragments of FOL designed for structural taxonomy management.
  - Key Concepts: Concept definitions, Role assertions, and TBox/ABox separation.
  - Reasoning Services: Subsumption checking (if concept $A \subseteq B$) and instance classification.

Reasoning with Default Information

Non-monotonic reasoning mechanisms handle incomplete domain knowledge by adopting plausible default assumptions.

- Non-Monotonic Reasoning
  - Standard logic is monotonic: adding new facts never invalidates derived conclusions.
  - Non-monotonic systems revise or retract conclusions when new counter-evidence is introduced.
- Default Logic (Reiter)
  - Default rules format: $\frac{A:B}{C}$ (If A is true, and B is consistent with KB, infer C).
  - Example: Most birds fly; unless an entity is established as a Penguin.
- Closed World Assumption (CWA)
  - Assumes any atomic fact not explicitly asserted in the KB is false.

Exam Corner: Likely AKTU Questions

- **Q1** Explain First Order Predicate Logic (FOPL) syntax and represent 'All birds fly except penguins' in FOPL.
- **Q2** Write the Unification algorithm and find the MGU for expressions P(x, f(g(z))) and P(a, f(y)).
- **Q3** Differentiate between Forward Chaining and Backward Chaining with suitable diagrams and comparative points.
- **Q4** Explain the step-by-step process of converting arbitrary FOPL formulas into Conjunctive Normal Form (CNF).
- **Q5** Explain Resolution Refinement. Prove a given target goal using Resolution Refinement method with an example.
- **Q6** What is Ontological Engineering? Explain how categories, objects, and events are formally structured.
- **Q7** Write detailed notes on: (a) Mental Events and Mental Objects, (b) Non-Monotonic Reasoning & Default Logic.

Unit 3 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 4 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

