3.3 Inference in First-Order Predicate Logic (FOPL)

Inference is the process of deriving new knowledge from existing facts and rules using logical reasoning. In FOPL, two key rule-based inference techniques are:

Forward Chaining (Data-Driven Reasoning)

Starts from known facts and applies rules to derive new facts until the goal is found.

Mechanism:

Start with a knowledge base of known facts.

Apply inference rules (like Modus Ponens) to deduce new facts.

Continue until:

The goal is reached, or

No more new facts can be derived.

Used in:

Expert systems

Real-time monitoring

Rule engines (e.g., Drools, CLIPS)

Example:

Knowledge Base:

1. Human(Socrates)
2. ∀x (Human(x) ⇒ Mortal(x)) Steps:

From 1: Human(Socrates)

From 2: Human(x) ⇒ Mortal(x)

Apply rule: Since Human(Socrates) → conclude Mortal(Socrates)

Backward Chaining (Goal-Driven Reasoning)

Starts from the goal and works backward to see if it can be proven from known facts.

Mechanism:

Start with the goal (query) you want to prove.

Check if goal matches a known fact or can be derived from a rule.

Recursively prove the premises of the rule until:

All are true → Goal is proved, or No rules match → Fail Used in:

Prolog (logic programming)

Theorem provers

Diagnostic systems

Example:

Goal: Mortal(Socrates)

Steps:

Is Mortal(Socrates) known?

Is there a rule? Yes: ∀x (Human(x) ⇒ Mortal(x)) To prove: Human(Socrates) Is Human(Socrates) known?

Therefore, Mortal(Socrates) is proved

Feature	**Forward chaining**	**Backward chaining**
Direction	From facts to conclusion	From goals to facts
Strategy	Data- driven	Goal-driven
When to use	All consequences from facts	Prove specific query
Efficiency	Can generate many irrelevant facts	More efficient for specific goals
Used in	Production system	Logic programming