U-5 (Reasoning & Inferencing using First Order Logic)

2)

Inference in First Order Logic -

- used to deduce new facts or sentences from existing sentences

Termonologies used in FOL

→ Substitution -
- performed on terms & formula
- complex is presence of quantifiers in FOL.
- $F[a/x]$, substitution a constant "a" in place of variable "x".

→ Equality -
- doesn't only use predicate & terms for making atomic sentences.
- another way → equality in FOL.
- equality symbol → specify two terms refer to same object.

FOL inference rules for quantifier -

1) Universal Generalization -
↳ is a valid inference rule which states that if premise $P(c)$ is true for any arbitrary element c in universe of discourse.
conclusion → $\forall x \, P(x)$

$$\frac{P(c)}{\forall x \, P(x)}$$

- If every element has similar property
↳ then used.
- x → not appear as a free variable.

2) Universal Instantiation -

- universal elimination or UI → valid inference rule.
- applied multiple time to add new sentences.
- new KB → logically equivalent to previous KB.
- UI → any sentences obtained by substituting → ground term for variable.
- UI rule state → infer any sentence P(c) by substituting ground term c from $\forall x\, P(x)$.
  any object in universe of discourse.

$$\frac{\forall x\, P(x)}{P(c)}$$

3) Existential Instantiation

- existential elimination
- applied only once to replace existential sentence.
- new KB → = old KB, if it is satisfiable if old KB was satisfiable
- one can infer P(c) from formula given in form of $\exists x\, P(x)$ for a new constant symbol c.
- restriction with this rule is c used in rule → new term for P(c) is true.

$$\frac{\exists x\, P(x)}{P(c)}$$

4) Existential introduction -

- existential generalization.
- some element c in universe of discourse property P, P

$$\frac{P(c)}{\exists x\, P(x)}$$

Generalized Modus Ponens Rule—

- for inference process in FOL we have single inference rule.
- lifted version of Modus Ponens.
- generalized modus Ponen
  - ↳ "P implies Q & P is asserted to be true, therefore Q must be true".

for atomic sentences → p_i, p_i', q.
where substitution θ such that $SUBST(\theta, p_i') = SUBST(\theta, p_i)$

$$\frac{p_1', p_2' \ldots p_n', (p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)}{SUBST(\theta, q)}$$

eg →
we will use this rule for Kings are evil, so will find some x such that x is king, & x is gredy so we infer that x is evil.

Unification—

- process of finding a substitute that makes two seperate logical atomic expression identical.
- substitution process.
- accepts two literals as input & uses substitution → make them identical
- Let Ψ_1 & Ψ_2 → two ~~automatic~~ atomic sentences.
  σ → unifier → $\Psi_1\sigma = \Psi_2\sigma$ → UNIFY (Ψ_1, Ψ_2)

* Find MGU for Unify {King (x), King (John)}

Let Ψ_1 = King (x), Ψ_2 = King (John),
substitution θ = {John / x}

- unifer for atoms → both phrases → identical after placement.

- UNIFY algo → employed, takes two automic statements & returns a unifer for each of them.
- First order techniques → heavily reliable on unification.
- expression don't match → result is failure
- replacement variables → MGU (Most General Unifier)

condition -

- Atoms or expressions → various predicate symbols → never united since they have different predicate symbols.
- ~~Both phrases → same no of arguments~~
- two comparable variables appear → same operation → unification fails
- no of arguments → both expression → identical
-

#

* Forward Chaining & Backward Chaining.

* Inference engine -

	Backward Chaining	Forward Chaining.
Known as	Goal - driven	Data Driven
starts from	possible conclusion	new data
processing	efficient	somewhat wasteful.
aims for	Necessary data	Any conclusion(s)
Approach	Conservative/Cautious	Opportunistic
Practical if	no. of possible final answer is reasonable. or a set of known alternatives is available.	combinatorial explosion creates an infinite no. of possible right answers
Appropriate for	Diagnostic, prescription & debugging appln	planning, monitoring, control & interpretation appln.
Reasoning	Top - down reasoning	Bottom - up reasoning
Type of Search	Depth - First Search	Breadth - ~~First~~ Search
who determine search	Consequents determine search	Antecedents determine search
Flow	consequences to antecedent	Antecedent to consequent

Comparision b/w Propositional logic & FOL:

Prepositional Logic	Predicate Logic (FOL).
• can't represent small worlds like vaccum cleaner world.	• well represents small world's problem.
• weak knowledge represn language	• strong knowledge language.
• uses prepositions in which complete sentence → by symbol.	• involves constants, variables, functions, relations.
• can't directly represent properties of individual entities or reln b/w individual entities. eg- Meera is short.	• can directly represent properties of individual entia or reln b/w individuals entities using individual predicts using function. eg- short (Meera).
• can't express specialization, generalization or patterns, etc. eg- All rectangles have 4 sides	• can express, generalization eg- no_of_sides (rectangle, 4)
• fondation level logic.	• higher level logic.
• not sufficiently expressive to represent complex statement	• can represent complex statement
• assumes world contains facts.	• assumes world contains, objects, relations, functions like natural ~~process~~ language
• meaning of facts is context independant unlike natural language	• context dependent like Natural language.
• declarative in nature	• derivate in nature

Page No.
Date / /

Ontological Engineering -

→ field of engineering → how to make representations that are more broad & adaptable.

- Actions, time, physical objects & beliefs are examples of concepts.
- Works on far scale than K.E.
- Instances classes & axioms are used to represent concept of ontologies
- knowledge engineering → ontologies → knowledge representation.

- Upper ontology

→ limitations of logic representation -
- red, green & yellow tomatoes → exceptions & uncertainity.

→ process of representing abstract concepts

Categories & Objects -

- sorting inform items into categories.

- using predicates & objects of first order logic ∴ ∀x → Apple (x)
- using reification of categories into objects
  ∴ preposition into an object. eg → Apples.

Categories are used to simplify & organize knowledge base.
help of inheritance.

subclass, sub-category relⁿ are generally used in categories, automatically follow rules of inheritance.

Page No.
Date / /

- subcategory → inherits properties of super category.

Knowledge → 1) food is good for health 2) Apple is Fruit.
subclass → Food & Apple → Fruit.

Deduce that → Apple is good for health.

\# Few rules that category needs.

↳ 1) an object is member of a category eg - MemberOf(Apple, Fruit)

2) category of subclass of another category eg - subsetOf(Fruit, Food).

3) members of category have properties eg - $\forall x [\text{MemberOf}(x, \text{Food}) \Rightarrow \text{Good-For-Health}(x)]$

4) all member of category can be recognized by some properties.

5) category as a whole has some properties.