First order logic
(Predicate Logic)

First order logic(Predicate Logic)

It is another type of logic that allows us to express more complex ideas more succinctly than propositional logic.

First order logic uses two types of symbols: Constant Symbols and Predicate Symbols. Constant symbols represent objects, while predicate symbols are like relations or functions that take an argument and return a true or false value.

Predicates are properties, additional information to better express the subject of the sentence.

A quantified predicate is a proposition , that is, when you assign values to a predicate with variables it can be made a proposition.

Propositional Logic	Predicate Logic
Propositional logic is the logic that deals with a collection of declarative statements which have a truth value, true or false.	Predicate logic is an expression consisting of variables with a specified domain. It consists of objects, relations and functions between the objects.
It is the basic and most widely used logic. Also known as Boolean logic.	It is an extension of propositional logic covering predicates and quantification.
A proposition has a specific truth value, either true or false.	A predicate's truth value depends on the variables' value.
Scope analysis is not done in propositional logic.	Predicate logic helps analyze the scope of the subject over the predicate. There are three quantifiers : Universal Quantifier (∀) depicts for all, Existential Quantifier (∃) depicting there exists some and Uniqueness Quantifier (∃!) depicting exactly one.
Propositions are combined with Logical Operators or Logical Connectives like Negation(¬), Disjunction(∨), Conjunction(∧), Exclusive OR(⊕), Implication(⇒), Bi-Conditional or Double Implication(⇔).	Predicate Logic adds by introducing quantifiers to the existing proposition.
It is a more generalized representation.	It is a more specialized representation.

Universal Quantification

Quantification is a tool that can be used in first order logic to represent sentences without using a specific constant symbol.

Universal quantification uses the symbol ∀ to express "for all".

The sentence $\forall x$: BelongsTo(x, Gryf) $\rightarrow$ $\neg$BelongsTo(x, Huffle) expresses the idea that it is true for every symbol that if this symbol belongs to Gryff, it does not belong to Huffle.

Existential quantification

While universal quantification was used to create sentences that are true for all x, existential quantification is used to create sentences that are true for at least one x. It is expressed using the symbol ∃.

For example, the sentence ∃x: House(x) ∧ BelongsTo(Minerva, x) means that there is at least one symbol that is both a house and that Minerva belongs to it.

In other words, this expresses the idea that Minerva belongs to a house.

Existential and universal quantification can be used in the same sentence.

For example, the sentence $\forall x.\ Person(x) \rightarrow (\exists y.\ House(y) \land BelongsTo(x, y))$ expresses the idea that if x is a person, then there is at least one house, y, to which this person belongs.

In other words, this sentence means that every person belongs to a house.

Using Predicate Logic

1. Marcus was a man.
2. Marcus was a Pompeian.
3. All Pompeians were Romans.
4. Caesar was a ruler.
5. All Pompeians were either loyal to Caesar or hated him.
6. Every one is loyal to someone.
7. People only try to assassinate rulers they are not loyal to.
8. Marcus tried to assassinate Caesar.

Using Predicate Logic

1. Marcus was a man.

   man(Marcus)

Using Predicate Logic

2. Marcus was a Pompeian.

   Pompeian(Marcus)

Using Predicate Logic

3. All Pompeians were Romans.

   $\forall x$: Pompeian(x) $\rightarrow$ Roman(x)

Using Predicate Logic

4. Caesar was a ruler.

   ruler(Caesar)

Using Predicate Logic

5. All Pompeians were either loyal to Caesar or hated him.

   It can be represented in any of the following two forms:

   inclusive-or

   $\forall x$: Roman(x) $\rightarrow$ loyalto(x, Caesar) $\vee$ hate(x, Caesar)

   exclusive-or

   $\forall x$: Roman(x) $\rightarrow$ (loyalto(x, Caesar) $\wedge$ $\neg$hate(x, Caesar)) $\vee$

   ($\neg$loyalto(x, Caesar) $\wedge$ hate(x, Caesar))

Using Predicate Logic

6. Every one is loyal to someone.

   $\forall x: \exists y: \text{loyalto}(x, y)$ $\exists y: \forall x: \text{loyalto}(x, y)$

Using Predicate Logic

7. People only try to assassinate rulers they are not loyal to.

$$\forall x: \forall y: person(x) \wedge ruler(y) \wedge tryassassinate(x, y) \rightarrow \neg loyalto(x, y)$$

Using Predicate Logic

8. Marcus tried to assassinate Caesar.

   tryassassinate(Marcus, Caesar)

Using Predicate Logic

Was Marcus loyal to Caesar?

man(Marcus)

ruler(Caesar)

tryassassinate(Marcus, Caesar)

$\Downarrow$ $\forall x$: man(x) $\rightarrow$ person(x)

$\neg$loyalto(Marcus, Caesar)

Using Predicate Logic

- Many English sentences are ambiguous.
- There is often a choice of how to represent knowledge.
- Obvious information may be necessary for

  reasoning
- We may not know in advance which statements to deduce (P or $\neg$P).

Reasoning

1. Marcus was a Pompeian.
2. All Pompeians died when the volcano erupted in 79 A.D.
3. It is now 2008 A.D.

Is Marcus alive?

Reasoning

1. Marcus was a Pompeian.

   Pompeian(Marcus)

2. All Pompeians died when the volcano erupted in 79 A.D.

   $\forall x$: Pompeian(x) $\rightarrow$ died(x, 79)

   erupted(volcano, 79)

3. It is now 2008 A.D.

   now = 2008

4. If someone dies, then he is dead at all later times
   $\forall x$: $\forall t_1$: $\forall t_2$: died(x, t_1) ^ greater-than(t_2, t_1) $\rightarrow$ dead(x, t_2)

4. Alive means not dead
   $\forall x$: $\forall t$:[alive(x,t)-> $\neg$dead(x,t)] ^ [$\neg$dead(x,t) $\rightarrow$ alive(x, t)]

Resolution

- $\neg A \lor B \Leftrightarrow A \Rightarrow B$ *(implication)*
- $A \Rightarrow B \Leftrightarrow \neg B \Rightarrow \neg A$ *(contraposition)*
- $(A \Rightarrow B) \land (B \Rightarrow A) \Leftrightarrow (A \Leftrightarrow B)$ *(equivalence)*
- $\neg(A \land B) \Leftrightarrow \neg A \lor \neg B$ *(De Morgan's law)*
- $\neg(A \lor B) \Leftrightarrow \neg A \land \neg B$
- $A \lor (B \land C) \Leftrightarrow (A \lor B) \land (A \lor C)$ *(distributive law)*
- $A \land (B \lor C) \Leftrightarrow (A \land B) \lor (A \land C)$
- $A \lor \neg A \Leftrightarrow w$ *(tautology)*
- $A \land \neg A \Leftrightarrow f$ *(contradiction)*

Conversion to Clause Form

1. Eliminate $\rightarrow$.

   $$P \rightarrow Q \equiv \neg P \vee Q$$

2. Reduce the scope of each $\neg$ to a single term.

   $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$
   $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$
   $\neg \forall x: P \equiv \exists x: \neg P$
   $\neg \exists x: p \equiv \forall x: \neg P$
   $\neg\neg P \equiv P$

3. Standardize variables so that each quantifier binds a unique variable.

   $$(\forall x: P(x)) \vee (\exists x: Q(x)) \equiv (\forall x: P(x)) \vee (\exists y: Q(y))$$

Conversion to Clause Form

4. Move all quantifiers to the left without changing their relative order.

   $(\forall x: P(x)) \vee (\exists y: Q(y)) \equiv \forall x: \exists y: (P(x) \vee (Q(y))$

5. Eliminate $\exists$ (Skolemization).

   $\exists x: P(x) \equiv P(c)$ Skolem constant

   $\forall x: \exists y\ P(x, y) \equiv \forall x: P(x, f(x))$ Skolem function

6. Drop $\forall$.

   $\forall x: P(x) \equiv P(x)$

7. Convert the formula into a conjunction of disjuncts.

   $(P \wedge Q) \vee R \equiv (P \vee R) \wedge (Q \vee R)$

8. Create a separate clause corresponding to each conjunct.
9. Standardize apart the variables in the set of obtained clauses.

Example

1. Marcus was a man.
2. Marcus was a Pompeian.
3. All Pompeians were Romans.
4. Caesar was a ruler.
5. All Pompeians were either loyal to Caesar or hated him.
6. Every one is loyal to someone.
7. People only try to assassinate rulers they are not loyal to.
8. Marcus tried to assassinate Caesar.

Prove: Marcus hate Caesar

Example

1. Man(Marcus).
2. Pompeian(Marcus).
3. $\forall x$: Pompeian(x) $\rightarrow$ Roman(x).
4. ruler(Caesar).
5. $\forall x$: Roman(x) $\rightarrow$ loyalto(x, Caesar) $\vee$ hate(x, Caesar).
6. $\forall x$: $\exists y$: loyalto(x, y).
7. $\forall x$: $\forall y$: person(x) $\wedge$ ruler(y) $\wedge$ tryassassinate(x, y)
   $\rightarrow$ $\neg$loyalto(x, y).
8. tryassassinate(Marcus, Caesar).

Clause form..

1.*man(Marcus)*

2.*Pompiean(Marcus)*

3.- *Pompiean(x1)* v *Roman(x1)*

4.*ruler(Caesar)*

5.- *Roman(x2)* v *loyalto(x2,Caesar)* v *hate(x2,Caesar)*

6. *loyal(x3,f(x3))*

7.-*man(x4)* v -*ruler(y1)* v - *tryassassinate(x4,y1)* v

-*loyalto(x4,y1)*

8.*tryassassinate(Marcus,Caesar)*

Prove: hate(Marcus,Caesar)