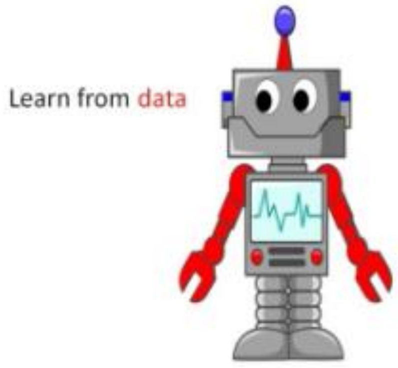
Artificial Intelligence

❑ Propositional Logic in Artificial intelligence

- Facts about Propositional Logic
- Types of Propositional Logic
- Logical Connectives
- Truth Table
- Precedence of Connectives
- Logical Equivalence
- Properties of Operators
- Limitations of Propositional Logic

❑ Rules of Inference in Artificial intelligence

- Inference
- Inference Rules
- Types of Inference Rules



❑ Propositional Logic in Artificial intelligence

Propositional Logic

- Propositional logic (PL) is the simplest form of logic where all the statements are made by propositions.
- A proposition is a declarative statement which is either true or false.
- It is a technique of knowledge representation in logical and mathematical form.

Propositional Logic

Proposition: A proposition (or Statement) is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.

Examples

1. Is the following sentence a proposition? If it is a proposition, determine whether it is true or false.

Paris is the capital of France.

This makes a declarative statement, and hence is a proposition. The proposition is TRUE (T).

Examples (Propositions Cont.)

2. Is the following sentence a proposition? If it is a proposition, determine whether it is true or false.

Are you coming today?.

This is a question not the declarative sentence and hence not a proposition.

Examples (Propositions Cont.)

3. Is the following sentence a proposition? If it is a proposition, determine whether it is true or false.

Take two marker from office.

This is an imperative sentence not the declarative sentence and therefore not a proposition.

Examples (Propositions Cont.)

4. Is the following sentence a proposition? If it is a proposition, determine whether it is true or false.

$$x + 4 > 9.$$

Because this is true for certain values of *x* (such as $x = 6$) and false for other values of *x* (such as $x = 5$), it is not a proposition.

Examples (Propositions Cont.)

5. Is the following sentence a proposition? If it is a proposition, determine whether it is true or false.

He is a university student.

Because truth or falsity of this proposition depend on the reference for the pronoun *he*. it is not a proposition.

Notations

- The small letters are commonly used to denote the propositional variables, that is, variables that represent propositions, such as, p, q, r, s, ….

- The truth value of a proposition is true, denoted by T or 1, if it is a true proposition and false, denoted by F or 0, if it is a false proposition.

Compound Propositions

Producing new propositions from existing propositions.

Logical Operators or Connectives

1. Not $\neg$
2. And $\wedge$
3. Or $\vee$
4. Exclusive or $\oplus$
5. Implication $\rightarrow$
6. Biconditional $\leftrightarrow$

Compound Propositions

Negation of a proposition

Let p be a proposition. The negation of p, denoted by $\neg p$ *(also denoted by ~p)*, is the statement

"It is not the case that p".

The proposition $\neg p$ is read as "not p". The truth values of the negation of p, $\neg p$, is the opposite of the truth value of p.

Examples

1. Find the negation of the following proposition

p : Today is Friday.

The negation of p is

$\neg p$: It is not the case that today is Friday.

This negation can be more simply expressed by

$\neg p$: Today is not Friday.

Examples

2. Write the negation of

“6 is negative”.

The negation is

“It is not the case that 6 is negative”.

or “6 is nonnegative”.

Truth Table (NOT)

- Unary Operator, Symbol: ¬

p	¬p
true	false
false	true

Conjunction (AND)

Definition

Let p and q be propositions. The conjunction of p and q, denoted by $p \wedge q$, is the proposition "*p and q*".

The conjunction $p \wedge q$ is true when *p and q* are both true and is false otherwise.

Examples

1. Find the conjunction of the propositions p and q, where

p : Today is Friday.

q : It is raining today.

The conjunction is

$p \wedge q$: Today is Friday and it is raining today.

Truth Table (AND)

- Binary Operator, Symbol: $\wedge$

p	q	$p \wedge q$
true	true	true
true	false	false
false	true	false
false	false	false

Disjunction (OR)

Definition

Let *p* and *q* be propositions. The disjunction of *p* and *q*, denoted by $p \vee q$, is the proposition "*p or q*".

The disjunction $p \vee q$ is false when both *p* and *q* are false and is true otherwise.

Examples

1. Find the disjunction of the propositions p and q, where

p : Today is Friday.

q : It is raining today.

The disjunction is

$p \vee q$: Today is Friday or it is raining today.

Truth Table (OR)

- Binary Operator, Symbol: $\vee$

p	q	$p \vee q$
true	true	true
true	false	true
false	true	true
false	false	false

Exclusive OR (XOR)

Definition

Let p and q be propositions. The *exclusive or* of p and q, denoted by $p \oplus q$, is the proposition "$p \oplus q$".

The *exclusive or*, $p \oplus q$, is true when exactly one of p or q is true and is false otherwise.

Examples

1. Find the *exclusive or* of the propositions p and q, where

p : Atif will pass the course CSC102.

q : Atif will fail the course CSC102.

The *exclusive or* is

$p \oplus q$: Atif will pass or fail the course CSC102.

Truth Table (XOR)

- Binary Operator, Symbol: $\oplus$

p	q	$p \oplus q$
true	true	false
true	false	true
false	true	true
false	false	false

Examples (OR vs XOR)

The following proposition uses the (English) connective "or". Determine from the context whether "or" is intended to be used in the inclusive or exclusive sense.

1. "Nabeel has one or two brothers".

A person cannot have both one and two brothers.
Therefore, "or" is used in the exclusive sense.

Examples (OR vs XOR)

2. To register for BSC you must have passed the qualifying exam or be listed as an Math major.

Presumably, if you have passed the qualifying exam and are also listed as an Math major, you can still register for BCS. Therefore, “or” is inclusive.

Composite Statements

Statements and operators can be combined in any way to form new statements.

p	q	$\neg p$	$\neg q$	$(\neg p)\vee(\neg q)$
true	true	false	false	false
true	false	false	true	true
false	true	true	false	true
false	false	true	true	true

Translating English to Logic

I did not buy a lottery ticket this week or I bought a lottery ticket and won the million dollar on Friday.

Let p and q be two propositions

p: I bought a lottery ticket this week.

q: I won the million dollar on Friday.

In logic form

$$\neg p \vee (p \wedge q)$$

Conditional Statements

Implication

Definition: Let p and q be propositions. The *conditional statement* $p \rightarrow q$, is the proposition "*If p, then q*".

The *conditional statement* $p \rightarrow q$ is false when p is true and q is false and is true otherwise.

where p is called hypothesis, antecedent or premise.

q is called conclusion or consequence

Implication (if - then)

- Binary Operator, Symbol: $\rightarrow$

P	Q	$P \rightarrow Q$
true	true	true
true	false	false
false	true	true
false	false	true

Conditional Statements

Biconditional Statements

Definition: Let p and q be propositions. The *biconditional statement* $p \leftrightarrow q$, is the proposition "*p if and only if q*".

The *biconditional (bi-implication) statement* $p \leftrightarrow q$ is true when p and q have same truth values and is false otherwise.

Biconditional (if and only if)

- Binary Operator, Symbol: $\leftrightarrow$

P	Q	$P \leftrightarrow Q$
true	true	true
true	false	false
false	true	false
false	false	true

Composite Statements

- Statements and operators can be combined in any way to form new statements.

P	Q	$\neg P$	$\neg Q$	$(\neg P)\vee(\neg Q)$
true	true	false	false	false
true	false	false	true	true
false	true	true	false	true
false	false	true	true	true

Equivalent Statements

P	Q	$\neg(P \wedge Q)$	$(\neg P) \vee (\neg Q)$	$\neg(P \wedge Q) \leftrightarrow (\neg P) \vee (\neg Q)$
true	true	false	false	true
true	false	true	true	true
false	true	true	true	true
false	false	true	true	true

- Two statements are called logically equivalent if and only if (iff) they have identical truth tables
  - The statements $\neg(P \wedge Q)$ and $(\neg P) \vee (\neg Q)$ are logically equivalent, because $\neg(P \wedge Q) \leftrightarrow (\neg P) \vee (\neg Q)$ is always true.

Tautologies and Contradictions

- Tautology is a statement that is always true regardless of the truth values of the individual logical variables
- Examples:
- $R \vee (\neg R)$
- $\neg(P \wedge Q) \leftrightarrow (\neg P) \vee (\neg Q)$
- If $S \rightarrow T$ is a tautology, we write $S \Rightarrow T$.
- If $S \leftrightarrow T$ is a tautology, we write $S \Leftrightarrow T$.

Tautologies and Contradictions

- A Contradiction is a statement that is always false regardless of the truth values of the individual logical variables

Examples

- $R \wedge (\neg R)$
- $\neg(\neg(P \wedge Q) \leftrightarrow (\neg P) \vee (\neg Q))$
- The negation of any tautology is a contradiction, and the negation of any contradiction is a tautology.

Examples

- Proposition p: Ali is smart.
- Proposition q: Ali is honest

<u>Examples</u>

- Ali is not smart but honest.
- Either Ali is smart, or he is not smart but honest.
- If Ali is smart, then he is not honest.
- Ali is either smart or honest, but Ali is not honest if he is smart:

Exercises

- We already know the following tautology:
- $\neg(P \wedge Q) \Leftrightarrow (\neg P) \vee (\neg Q)$
- Nice home exercise:
- Show that $\neg(P \vee Q) \Leftrightarrow (\neg P) \wedge (\neg Q)$.
- These two tautologies are known as De Morgan's laws.

Propositional Logic in Artificial intelligence

Propositional logic (PL) is the simplest form of logic where all the statements are made by propositions. A proposition is a declarative statement which is either true or false. It is a technique of knowledge representation in logical and mathematical form.

Example

a) It is Sunday.
b) The Sun rises from West (False proposition)
c) 3+3= 7(False proposition)
d) 5 is a prime number.

Facts about Propositional Logic

- Propositional logic is also called Boolean logic as it works on 0 and 1.
- In propositional logic, we use symbolic variables to represent the logic, and we can use any symbol for a representing a proposition, such A, B, C, P, Q, R, etc.
- Propositions can be either true or false, but it cannot be both.
- Propositional logic consists of an object, relations or function, and **logical connectives**.
- These connectives are also called logical operators.
- The propositions and connectives are the basic elements of the propositional logic.
- Connectives can be said as a logical operator which connects two sentences.

Facts about Propositional Logic

- A proposition formula which is always true is called **tautology**, and it is also called a valid sentence.
- A proposition formula which is always false is called **Contradiction**.
- A proposition formula which has both true and false values is called **Contingency.**
- Statements which are questions, commands, or opinions are not propositions such as "**Where is Rohini", "How are you", "What is your name**", are not propositions.
- The syntax of propositional logic defines the allowable sentences for the knowledge representation.
- A literal in propositional logic is a variable or its negation

Types of Propositional Logic

There are two types of Propositions:

1. Atomic Propositions
2. Compound propositions

Atomic Proposition: Atomic propositions are the simple propositions. It consists of a single proposition symbol. These are the sentences which must be either true or false.

Example:

- ”2+2 is 4”, it is an atomic proposition as it is a true fact.
- "The Sun is cold" is also a proposition as it is a false fact.

Compound Proposition: Compound propositions are constructed by combining simpler or atomic propositions, using parenthesis and logical connectives.

Example:

- "It is raining today, and the road is slippery."
- “Samar is a doctor, and his clinic is in Karachi."

Logical Connectives

Logical connectives are used to connect two simpler propositions or representing a sentence logically. We can create compound propositions with the help of logical connectives. There are mainly five connectives, which are given as follows:

1. **Negation:**

   A sentence such as ¬ P is called negation of P. A literal can be either Positive literal or negative literal.

   Example:

   P = Today is Sunday.

   ¬ P = Today is not Sunday.

   Q = It is raining.

   ¬ Q = It is not raining.

Logical Connectives

2. **Conjunction:**

A sentence which has ∧ connective such as, **P ∧ Q** is called a conjunction. **Example:**

"Samar is intelligent and hardworking". It can be written as:

P = Samar is intelligent, **Q = Samar is hardworking.**

P∧ Q = Samar is intelligent and hardworking.

3. **Disjunction:**

A sentence which has ∨ connective, such as **P ∨ Q**. is called disjunction, where P and Q are the propositions.

Example:

"Samar is a doctor or an Engineer."

P = Samar is Doctor.

Q= Samar is an Engineer.

P ∨ Q= Samar is a doctor or an Engineer.

Logical Connectives

4. <u>Implication:</u>

A sentence such as $P \rightarrow Q$, is called an implication. Implications are also known as if-then rules. It can be represented as
"If it is raining, **then** the road is slippery."
Let **P= It is raining**, and **Q= Street is wet**, so
$P \rightarrow Q$ = If it is raining, **then** the road is slippery.

5. <u>Biconditional:</u>

A sentence such as **$P \Leftrightarrow Q$ is a Biconditional sentence,**
Example: "If I am breathing, then I am alive.
P= I am breathing, **Q= I am alive**, it can be represented as:
$P \Leftrightarrow Q$ = If I am breathing, then I am alive.

Summarized Table for PL Connectives

Connective symbols	Word	Technical term	Example
$\wedge$	AND	Conjunction	$A \wedge B$
$\vee$	OR	Disjunction	$A \vee B$
$\rightarrow$	Implies	Implication	$A \rightarrow B$
$\Leftrightarrow$	If and only if	Biconditional	$A \Leftrightarrow B$
$\neg$ or $\sim$	Not	Negation	$\neg A$ or $\neg B$

Truth Table

In propositional logic, we need to know the truth values of propositions in all possible scenarios. We can combine all the possible combination with logical connectives, and the representation of these combinations in a tabular format is called Truth table. Following are the truth table for all logical connectives:

For Negation:

P	**¬ P**
True	**False**
False	**True**

For Conjunction:

P	**Q**	**P∧ Q**
True	True	**True**
True	False	**False**
False	True	**False**
False	False	**False**

Truth Table

For disjunction:

P	Q	P V Q.
True	True	**True**
False	True	**True**
True	False	**True**
False	False	**False**

For Implication:

P	Q	P→ Q
True	True	**True**
True	False	**False**
False	True	**True**
False	False	**True**

For Biconditional:

P	Q	P⇔ Q
True	True	**True**
True	False	**False**
False	True	**False**
False	False	**True**

Truth Table with Three Propositions

We can build a proposition composing three propositions P, Q, and R. This truth table is made-up of 8 Tuples as we have taken three proposition symbols.

P	Q	R	¬R	P∨Q	P∨Q→¬R
True	True	True	False	True	False
True	True	False	True	True	True
True	False	True	False	True	False
True	False	False	True	True	True
False	True	True	False	True	False
False	True	False	True	True	True
False	False	True	False	False	True
False	False	False	True	False	True

Precedence of Connectives

Just like arithmetic operators, there is a precedence order for propositional connectors or logical operators. This order should be followed while evaluating a propositional problem. Following is the list of the precedence order for operators:

Precedence	Operators
First Precedence	Parenthesis
Second Precedence	Negation
Third Precedence	Conjunction(AND)
Fourth Precedence	Disjunction(OR)
Fifth Precedence	Implication
Six Precedence	Biconditional

For better understanding use parenthesis to make sure of the correct interpretations. Such as ¬R∨ Q, It can be interpreted as (¬R) ∨ Q.

Logical Equivalence

Logical equivalence is one of the features of propositional logic. Two propositions are said to be logically equivalent if and only if the columns in the truth table are identical to each other.

Let's take two propositions A and B, so for logical equivalence, we can write it as $A \Leftrightarrow B$. In below truth table we can see that column for $\neg A \lor B$ and $A \rightarrow B$, are identical hence A is Equivalent to B.

A	B	¬A	¬A∨ B	A→B
T	T	F	**T**	**T**
T	F	F	**F**	**F**
F	T	T	**T**	**T**
F	F	T	**T**	**T**

Properties of Operators

Commutativity:

$P \wedge Q = Q \wedge P$, or

$P \vee Q = Q \vee P$.

Associativity:

$(P \wedge Q) \wedge R = P \wedge (Q \wedge R)$,

$(P \vee Q) \vee R = P \vee (Q \vee R)$

Identity element:

$P \wedge \text{True} = P$,

$P \vee \text{True} = \text{True}$.

Distributive:

$P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$.

$P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$.

DE Morgan's Law:

$\neg (P \wedge Q) = (\neg P) \vee (\neg Q)$

$\neg (P \vee Q) = (\neg P) \wedge (\neg Q)$.

Double-negation elimination:

$\neg (\neg P) = P$.

Limitations of Propositional Logic

- We cannot represent relations like ALL, some, or none with propositional logic.

  Example:

  All the girls are intelligent.

  Some apples are sweet.

- Propositional logic has limited expressive power.
- In propositional logic, we cannot describe statements in terms of their properties or logical relationships.

❑ Rules of Inference in Artificial intelligence

Inference and Inference Rules

In artificial intelligence, we need intelligent computers which can create new logic from old logic or by evidence, so generating the conclusions from evidence and facts is termed as Inference.

Inference rules are the templates for generating valid arguments.
Inference rules are applied to derive proofs in artificial intelligence, and the **proof** is a sequence of the conclusion that leads to the desired goal.
In inference rules, the implication among all the connectives plays an important role.

Following are some terminologies related to inference rules:

Terminologies

Implication:

It is one of the logical connectives which can be represented as $P \rightarrow Q$.

It is a Boolean expression.

Converse:

The converse of implication, which means the right-hand side proposition goes to the left-hand side and vice-versa. It can be written as $Q \rightarrow P$.

Contrapositive:

The negation of converse is termed as contrapositive, and it can be represented as $\neg Q \rightarrow \neg P$.

Inverse: The negation of implication is called inverse.

It can be represented as $\neg P \rightarrow \neg Q$.

Terminologies

From the above term some of the compound statements are equivalent to each other, which we can prove using truth table:

P	**Q**	**$P \rightarrow Q$**	**$Q \rightarrow P$**	**$\neg Q \rightarrow \neg P$**	**$\neg P \rightarrow \neg Q$.**
T	**T**	T	T	T	T
T	**F**	F	T	F	T
F	**T**	T	F	T	F
F	**F**	T	T	T	T

Hence from the above truth table, we can prove that $P \rightarrow Q$ is equivalent to $\neg Q \rightarrow \neg P$, and $Q \rightarrow P$ is equivalent to $\neg P \rightarrow \neg Q$.

Types of Inference Rules

1. Modus Ponens:

The Modus Ponens rule is one of the most important rules of inference, and it states that if P and P → Q is true, then we can infer that Q will be true. It can be represented as:

Notation for Modus ponens: $$\frac{P \to Q, \quad P}{\therefore Q}$$

Example:
Statement-1: "If I am sleepy then I go to bed" ==> P→ Q
Statement-2: "I am sleepy" ==> P
Conclusion: "I go to bed." ==> Q.
Hence, we can say that,
if P→ Q is true and P is true then Q will be true.

Proof by Truth Table:

P	Q	$P \to Q$
0	0	0
0	1	1
1	0	0
1	1	1

Types of Inference Rules

2. Modus Tollens:

The Modus Tollens rule state that if P→ Q is true and ¬ Q is true, then ¬ P will also true. It can be represented as:

Notation for Modus Tollens: $\dfrac{P \rightarrow Q, \quad \sim Q}{\sim P}$

Example:

Statement-1: "If I am sleepy then I go to bed"
==> P→ Q

Statement-2: "I do not go to the bed."
==> ~Q

Statement-3: Which infers that
"I am not sleepy" => ~P

Proof by Truth Table:

P	Q	~P	~Q	$P \rightarrow Q$
0	0	1	1	1
0	1	1	0	1
1	0	0	1	0
1	1	0	0	1

Types of Inference Rules

3. Hypothetical Syllogism:

The Hypothetical Syllogism rule state that if P→R is true whenever P→Q is true, and Q→R is true.

Example:

Statement-1: If you have my home key then you can unlock my home. P→Q
Statement-2: If you can unlock my home then you can take my money. Q→R
Conclusion: If you have my home key then you can take my money. P→R

Proof by Truth Table:

P	Q	R	$P \to Q$	$Q \to R$	$P \to R$
0	0	0	1	1	1 ←
0	0	1	1	1	1 ←
0	1	0	1	0	1
0	1	1	1	1	1 ←
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	1	1 ←

Types of Inference Rules

4. Disjunctive Syllogism:

The Disjunctive syllogism rule state that if P∨Q is true, and ¬P is true, then Q will be true. It can be represented as:

Notation of Disjunctive syllogism: $\dfrac{P \vee Q, \quad \neg P}{Q}$

Example:

Statement-1: Today is Sunday or Monday. ==>P∨Q

Statement-2: Today is not Sunday. ==> ¬P

Conclusion: Today is Monday. ==> Q

Proof by Truth Table:

P	Q	¬P	P ∨ Q
0	0	1	0
0	1	1	1 ⟵
1	0	0	1
1	1	0	1

Types of Inference Rules

5. Addition:

The Addition rule is one the common inference rule, and it states that If P is true, then P∨Q will be true.

Notation of Addition: $\dfrac{P}{P \lor Q}$

Example:

Statement: I have a vanilla ice-cream.
==> P

Statement-2: I have Chocolate ice-cream.
==> Q

Conclusion: I have vanilla or chocolate ice-cream.
==> (P∨Q)

Proof by Truth Table:

P	Q	P∨Q
0	0	0
1	0	1
0	1	1
1	1	1

Types of Inference Rules

6. Simplification:

The simplification rule state that if P∧ Q is true, then Q or P will also be true. It can be represented as:

Notation of Simplification rule: $\frac{P \land Q}{Q}$ Or $\frac{P \land Q}{P}$

Proof by Truth Table:

P	Q	$P \land Q$
0	0	0
1	0	0
0	1	0
1	1	1

Types of Inference Rules

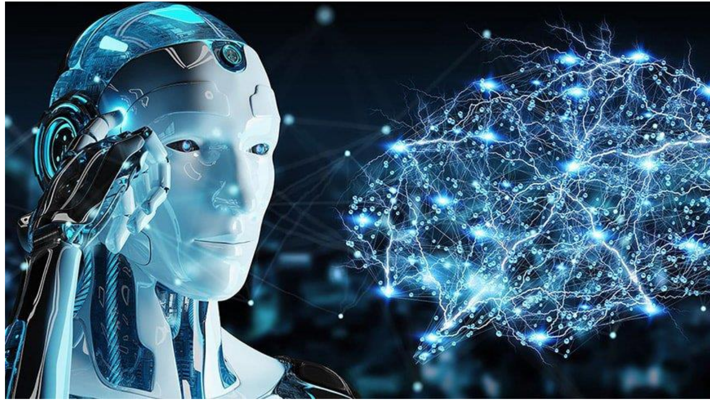
7. Resolution:

The Resolution rule state that if P∨Q and ¬ P∧R is true, then Q∨R will also be true. It can be represented as:

Notation of Resolution $\dfrac{P \lor Q, \quad \neg P \land R}{Q \lor R}$

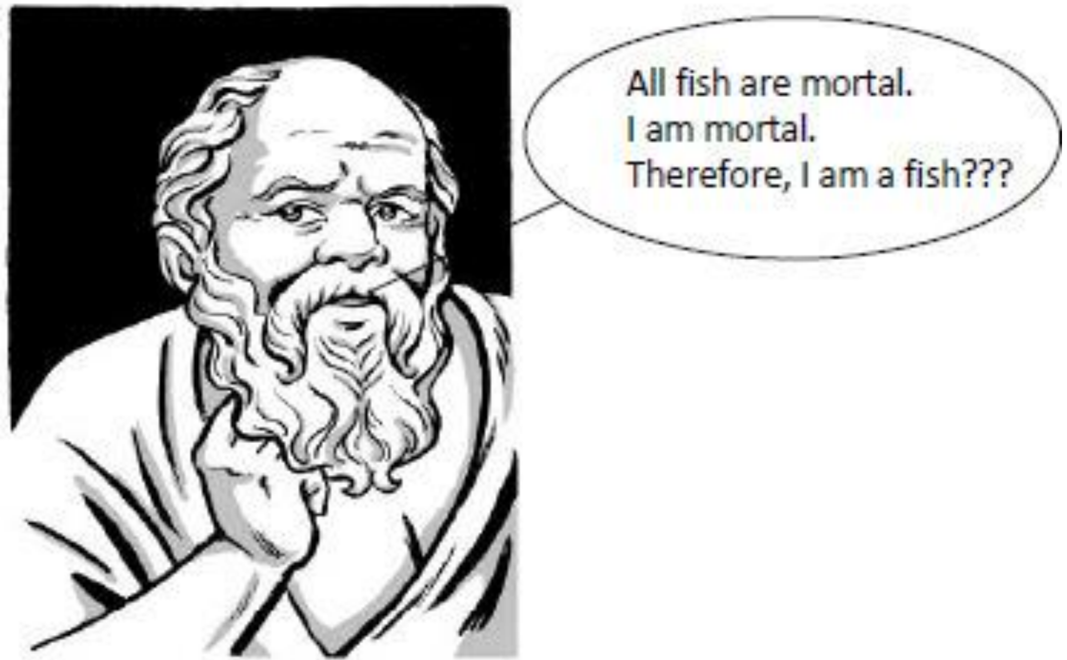
Proof by Truth Table:

P	¬ P	Q	R	$P \lor Q$	¬ P∧R	$Q \lor R$
0	1	0	0	0	0	0
0	1	0	1	0	0	1
0	1	1	0	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	0	0
1	0	0	1	1	0	1
1	0	1	0	1	0	1
1	0	1	1	1	0	1



Artificial Intelligence

- First-Order Logic in Artificial intelligence
- Knowledge Engineering in First-Order Logic



Scope of Propositional Logic

In the topic of Propositional logic, we have seen that how to represent statements using propositional logic. But unfortunately, in propositional logic, we can only represent the facts, which are either true or false. PL is not sufficient to represent the complex sentences or natural language statements.

The propositional logic has very limited expressive power. Consider the following sentence, which we cannot represent using PL logic.

"Some humans are intelligent", or

“John likes cricket."

To represent the above statements, PL logic is not sufficient, so we require some more powerful logic, such as first-order logic.

First-Order Logic

- First-order logic is another way of knowledge representation in artificial intelligence. It is an extension to propositional logic.
- FOL is sufficiently expressive to represent the natural language statements in a concise way.
- First-order logic is also known as **Predicate logic or First-order predicate logic**. It is a powerful language that develops information about the objects in a more easy way and can also express the relationship between those objects.
- First-order logic (like natural language) does not only assume that the world contains facts like propositional logic but also assumes the following things in the world:
  - ✓ **Objects:** A, B, people, numbers, colors, wars, theories, squares, pits, wumpus, ...
  - ✓ **Relations: It can be unary relation such as:** red, round, is adjacent, **or n-ary relation such as:** the sister of, brother of, has color, comes between
  - ✓ **Function:** Father of, best friend, third inning of, end of, ...

Two Main Parts of First-Order Logic

As a natural language, first-order logic also has two main parts:

- ✓ **Syntax, and**
- ✓ **Semantics**

Syntax of First-Order logic:

The syntax of FOL determines which collection of symbols is a logical expression in first-order logic. The basic syntactic elements of first-order logic are symbols. We write statements in short-hand notation in FOL.

The syntax of first-order logic is defined relative to a signature. A signature σ consists of a set of constant symbols, a set of function symbols and a set of predicate symbols. Typically we use letters c, d to denote constant symbols, f, g to denote function symbols and P, Q, R to denote predicate symbols.

Two Main Parts of First-Order Logic

Semantics of First-Order logic:

Semantic concerns the truth of a formula relative to some truth assignment.

Semantics relates the syntax to the world (relational structure).

$A \models \phi$ denotes that formula ϕ is true in the world A.

Here '$\models$' is the semantical relation.

Consider the statement $A \models (x = 2)$.

Does it make sense to ask whether the formula '$x = 2$' is true in the world A?

The truth of the formula depends on the value of x, but x is a variable and can take any value. Some values of x might make the formula true, while others might falsify it. Thus, we need to qualify our answer by saying that the formula is true when x has a certain value.

To do this we can use a function α that assigns values to variables. Such a function is called a 'binding'. Thus, statements about semantics have the form

$$A, \alpha \models \phi$$

where A is a structure, ϕ is a formula and α is a variable assignment.

Basic Elements of First-order logic

Following are the basic elements of FOL syntax:

Constant	1, 2, A, John, Islamabad, cat,....
Variables	x, y, z, a, b,....
Predicates	Brother, Father, >,....
Function	sqrt, LeftLegOf, ...
Connectives	∧, ∨, ¬, ⇒, ⇔
Equality	==
Quantifier	∀, ∃

Atomic & Complex Sentences

Atomic Sentences

- Atomic sentences are the most basic sentences of first-order logic. These sentences are formed from a predicate symbol followed by a parenthesis with a sequence of terms.
- We can represent atomic sentences as**Predicate (term1, term2, ..., term n)**.
- **Example:**

  Akram and Asad are brothers:
  => Brothers(Akram, Asad).
  bella is a cat:
  => cat (bella).

Complex Sentences

Complex sentences are made by combining atomic sentences using connectives.

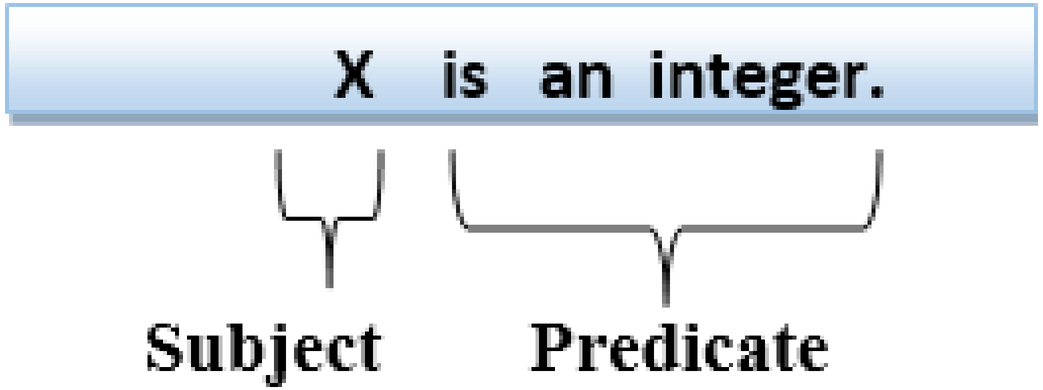
First-order Logic Statements can be Divided into Two Parts

Subject: Subject is the main part of the statement.
Predicate: A predicate can be defined as a relation, which binds two atoms together in a statement.
Consider the statement:

“X is an integer.",

it consists of two parts,

- ✓ the first part X is the **subject** of the statement, and
- ✓ second part "is an integer," is known as **a predicate**.



Quantifiers in First-order Logic

- A quantifier is a language element which generates quantification, and quantification specifies the quantity of specimen in the universe of discourse.
- Quantifiers are the symbols that permit to determine or identify the range and scope of the variable in the logical expression.

Types of Quantifier

There are two types of quantifier:

- Universal Quantifier, (for all, everyone, everything)
- Existential quantifier, (for some, at least one).

Universal Quantifier

Universal quantifier is a symbol of logical representation, which specifies that the statement within its range is true for everything or every instance of a particular thing.

The Universal quantifier is represented by a symbol ∀, which resembles an inverted A.

If x is a variable, then **∀x** is read as:

- **For all x**
- **For each x**
- **For every x.**

Note: In universal quantifier we use implication "→".

Example: Universal Quantifier

All man drink coffee.

Let there be a variable x which refers to a man so all x can be represented in UOD as below:

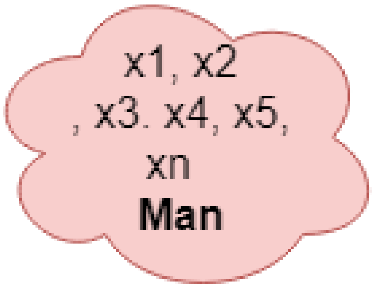
- x1 drinks coffee

  $\wedge$
- x2 drinks

  $\wedge$
- x3 drinks milk

  $\wedge$
- .
- .

  $\wedge$
- xn drinks milk



Universe of Discourse

So in shorthand notation, we can write it as :

Implication (main Connector)

$\forall x$ man(x) $\rightarrow$ drink (x, coffee).

It will be read as:

There are all x where x is a man who drink coffee.

Existential Quantifier

Existential quantifiers are the type of quantifiers, which express that the statement within its scope is true for at least one instance of something.

It is denoted by the logical operator ∃, which resembles as inverted E. When it is used with a predicate variable then it is called as an existential quantifier.

If x is a variable, then existential quantifier will be $\exists x$ or $\exists(x)$.

It will be read as:

- **There exists a 'x.'**
- **For some 'x.'**
- **For at least one 'x.'**

Note: In Existential quantifier we always use AND or Conjunction symbol (∧).

Example: Existential Quantifier

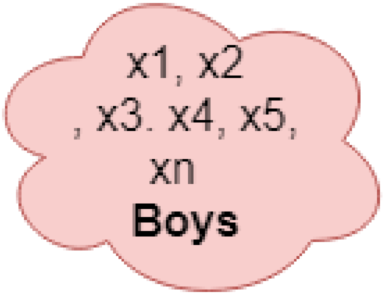
Some boys are intelligent.

- x1 is intelligent

  V
- x2 is intelligentV
- x3 is intelligent

  V
- .
- .

  V
- xn is intelligent



Universe of Discourse

AND (main Connector)

∃x: boys(x) ∧ intelligent(x)

It will be read as:

There are some x where x is a boy who is intelligent.

So in short-hand notation, we can write it as:

Properties of Quantifiers

- <u>**In Universal Quantifier**</u>

  $\forall x \forall y$ is similar to $\forall y \forall x$.

- <u>**In Existential quantifier**</u>

  $\exists x \exists y$ is similar to $\exists y \exists x$

- <u>**However,**</u>

  $\exists x \forall y$ is not similar to $\forall y \exists x$.

Some Examples of FOL using Quantifier

1. **All birds fly.**

In this question the predicate is "**fly(bird)**.“ And since there are all birds who fly so it will be represented as follows.

$$\forall x \ \mathrm{bird}(x) \rightarrow \mathrm{fly}(x).$$

2. **Every man respects his parent.**

In this question, the predicate is "**respect(x, y)," where x=man, and y= parent**.

Since there is every man so will use ∀, and it will be represented as follows:

$$\forall x \ \mathrm{man}(x) \rightarrow \mathrm{respects}\ (x, \mathrm{parent}).$$

3. **Some boys play cricket.**

In this question, the predicate is "**play(x, y)**," where x= boys, and y= game. Since there are some boys so we will use **∃,** and it will be represented as:

$$\exists x \ \mathrm{boys}(x) \wedge \mathrm{play}(x, \mathrm{cricket}).$$

Some Examples of FOL using Quantifier

4. Not all students like both Mathematics and Science.

In this question, the predicate is "**like(x, y)," where x= student, and y= subject**.

Since there are not all students, so we will use **∀ with negation, so** following representation for this:

¬∀ (x) [student(x) → like(x, Mathematics) ∧ like(x, Science)].

5. Only one student failed in Mathematics.

In this question, the predicate is "**failed(x, y)," where x= student, and y= subject**.

Since there is only one student who failed in Mathematics, so we will use following representation for this:

∃(x) [student(x) → failed (x, Mathematics) ∧ ∀ (y) [¬(x==y) ∧ student(y) → ¬failed (x, Mathematics)].

Free and Bound Variables

The quantifiers interact with variables which appear in a suitable way. There are two types of variables in First-order logic which are given below:

Free Variable:

A variable is said to be a free variable in a formula if it occurs outside the scope of the quantifier.

Example: $\forall x\ \exists(y)[P(x, y, z)]$, where z is a free variable.

Bound Variable:

A variable is said to be a bound variable in a formula if it occurs within the scope of the quantifier.

Example: $\forall x\ [A(x)\ B(y)]$, here x and y are the bound variables.

Next Lecture

Quiz Time