Unit 4

Knowledge Representation

- The objective of knowledge representation is to express knowledge in a computer tractable form such that it can be used by any algorithm to perform any task.

- Key aspects of knowledge representation languages are:

i. Syntax: describes how sentences are formed in the language.

ii. Semantics: describes the meaning of sentences, what is it the sentence refers to in the real world.

iii. Computational aspect: describes how sentences and objects are manipulated.

- A formal language is required to represent knowledge in a computer tractable form and reasoning processes are required to manipulate this knowledge to deduce non-obvious (unseen) facts.

•General features of a representation

•Representational adequacy

The ability to represent all kinds of knowledge that are needed in a certain domain.

•Inferential adequacy

The ability to represent all of the kinds of inferential procedures (procedures that manipulate the representational structures in such a way as to derive new structures corresponding to new knowledge inferred from old).

•Inferential efficiency

The ability to represent efficient inference procedures (for instance, by incorporating into the knowledge structure additional information that can be used to focus the attention of the inference mechanisms in the most promising directions).

•Acquisitional efficiency

The ability to acquire new information easily.

- **Knowledge Representation using Logic**

- Logics are formal languages for representing information such that conclusions can be drawn. Logic makes statements about the world which are true (or false).

Logic is:

- concise
- unambiguous
- context insensitive
- expressive
- effective for inferences

•Logic is defined by the following:

1. Syntax - describes the possible configurations that constitute sentences.

2. Semantics - determines what facts in the world the sentences refer to i.e. the interpretation. Each sentence makes a claim about the world.

3. Proof theory - set of rules for generating new sentences that are necessarily true given that the old sentences are true. The relationship between sentences is called entailment. The proof can be used to determine new facts which follow from the old.

Types of logic:

Different types of logics are: Propositional logic and Predicate logic (First –order Predicate Logic).

•Propositional logic

- A propositional logic is a declarative sentence which can be either true or false but not both or either.

- Propositional logic is a mathematical model that allows us to reason about the truth or falsehood of logical expression.

- In propositional logic, there are atomic sentences and compound sentences built up from atomic sentences using logical connectives.

Formal Logical Connectives

- In logic, a logical connective (also called a logical operator) is a symbol or word used to connect two or more sentences (of either a formal or a natural language) in a grammatically valid way, such that the sense of the compound sentence produced depends only on the original sentences. The most common logical connectives are binary connectives (also called dyadic connectives) which join two sentences which can be thought of as the function's operands. Also commonly, negation is considered to be a unary connective. Logical connectives along with quantifiers are the two main types of logical constants used in formal systems such as propositional logic and predicate logic.

- List of Logical connectives

Commonly used logical connectives include:

- Negation (not): $\neg$, ~
- Conjunction (and): $\wedge$, & , •
- Disjunction (or): $\vee$
- Material implication (if...then): $\longrightarrow$, $\Rightarrow$, $\supset$
- Biconditional (if and only if): $\leftrightarrow$, $\equiv$, $=$

Alternative names for biconditional are "iff", "xnor" and "bi-implication".

For example, the meaning of the statements *it is raining* and *I am indoors* is transformed when the two are combined with logical connectives:

- It is raining **and** I am indoors ($P \land Q$)
- **If** it is raining, **then** I am indoors ($P \rightarrow Q$)
- **If** I am indoors, **then** it is raining ($Q \rightarrow P$)
- I am indoors **if and only if** it is raining ($P \leftrightarrow Q$)
- It is **not** raining ($\neg P$)

For statement P = *It is raining* and Q = *I am indoors*.

Sentences in the propositional logic:

i. **Atomic sentences**: – Constructed from constants and propositional symbols – True, False are atomic sentences. Light in the room is on. It rains outside are (atomic) sentences.

ii. **Composite sentences**: – Constructed from valid sentences via connectives eg: $(A \wedge B)$ $(A \vee B)$ $(A \Rightarrow B)$ $(A \Leftrightarrow B)$ $(A \vee B) \wedge (A \vee \neg B)$

•Propositional logic is the simplest logic. We use the symbols like P1, P2 to represent sentences. A sentence (well-formed formula) is defined as follows

If S is a sentence, $\neg$S is a sentence (negation)
If S1 and S2 are sentences, S1 ^ S2 is a sentence (conjunction)
If S1 and S2 are sentences, S1 $\vee$ S2 is a sentence (disjunction)
If S1 and S2 are sentences, S1 $\Rightarrow$ S2 is a sentence (implication)
If S1 and S2 are sentences, S1 $\Leftrightarrow$ S2 is a sentence (biconditional)

Formal grammar for propositional logic can be given as below:

Sentennce	$\rightarrow$ AutomicSentence | ComplexSedntence
AutomicSentence	$\rightarrow$ True | False | Symbol
Symbol	$\rightarrow$ P | Q | R ....
ComplexSentence	$\rightarrow$ $\neg$Sentence
	| (Sentence ^ Sentence)
	| (Sentence $\vee$ Sentence)
	| (Sentence $\Rightarrow$ Sentence)
	| (Sentence $\Leftrightarrow$ Sentence)

P	Q	$\neg$P	P$\wedge$Q	P$\vee$Q	P$\Rightarrow$Q	P$\Leftrightarrow$Q
false	false	true	false	false	true	true
false	true	true	false	true	true	false
true	false	false	false	true	false	false
true	true	false	true	true	true	true

• Limitations of Propositional logic:

- We cannot represent relations like ALL, some, or none with propositional logic. Example:
  1. **All the girls are intelligent.**
  2. **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.

Tautology/Validity

A sentence is *valid* if it is true in all models,

$$\text{e.g., } True,\ A \vee \neg A,\ A \Rightarrow A,\ (A \wedge (A \Rightarrow B)) \Rightarrow B$$

Satisfiability

A sentence is satisfiable if it is true in some model

$$\text{e.g., } A \vee B_{\text{(OR)}}$$

Contradiction/Unsatisfiable

A sentence is unsatisfiable whose truth values are always false.

$$\text{e.g., } A \neg \wedge A$$

Logically Contigent: A formula or statement that is neither a tautology nor a contradiction is said to be logically contigent.

$$
\begin{aligned}
(\alpha \wedge \beta) &\equiv (\beta \wedge \alpha) \quad \text{commutativity of } \wedge \\
(\alpha \vee \beta) &\equiv (\beta \vee \alpha) \quad \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) &\equiv (\alpha \wedge (\beta \wedge \gamma)) \quad \text{associativity of } \wedge \\
((\alpha \vee \beta) \vee \gamma) &\equiv (\alpha \vee (\beta \vee \gamma)) \quad \text{associativity of } \vee \\
\neg(\neg\alpha) &\equiv \alpha \quad \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) \quad \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \vee \beta) \quad \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination} \\
\neg(\alpha \wedge \beta) &\equiv (\neg\alpha \vee \neg\beta) \quad \text{de Morgan} \\
\neg(\alpha \vee \beta) &\equiv (\neg\alpha \wedge \neg\beta) \quad \text{de Morgan} \\
(\alpha \wedge (\beta \vee \gamma)) &\equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) \quad \text{distributivity of } \wedge \text{ over } \vee \\
(\alpha \vee (\beta \wedge \gamma)) &\equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) \quad \text{distributivity of } \vee \text{ over } \wedge
\end{aligned}
$$

First-Order Logic in Artificial intelligence

- 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

"Sachin likes cricket."

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

•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**.
- First-order logic 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,......
  - **Relations: It can be unary relation such as:** red, round, is adjacent, **or n-any relation such as:** the sister of, brother of, has color, comes between
  - **Function:** Father of, best friend, third inning of, end of, ...
- As a natural language, first-order logic also has two main parts:
  - **Syntax**
  - **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.
- Basic Elements of First-order logic:
- Following are the basic elements of FOL syntax:

Constant	1, 2, A, John, Mumbai, cat,....
Variables	x, y, z, a, b,....
Predicates	Brother, Father, >,....
Function	sqrt, Left Leg Of, ...

Connectives	$\wedge$, $\vee$, $\neg$, $\Rightarrow$, $\Leftrightarrow$
Equality	==
Quantifier	$\forall$, $\exists$

- 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:**

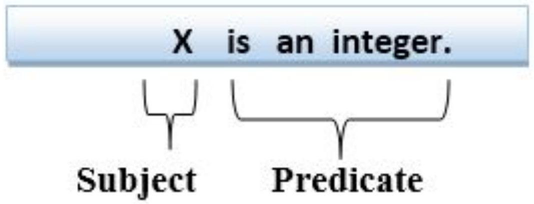
Ravi and Ajay are brothers: =>Brothers (Ravi, Ajay).

Chinky is a cat: => cat (Chinky).

- 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.

- These are the symbols that permit to determine or identify the range and scope of the variable in the logical expression. There are two types of quantifier:

1. **Universal Quantifier, (for all, everyone, everything)**

- **Existential quantifier, (for some, at least one).**

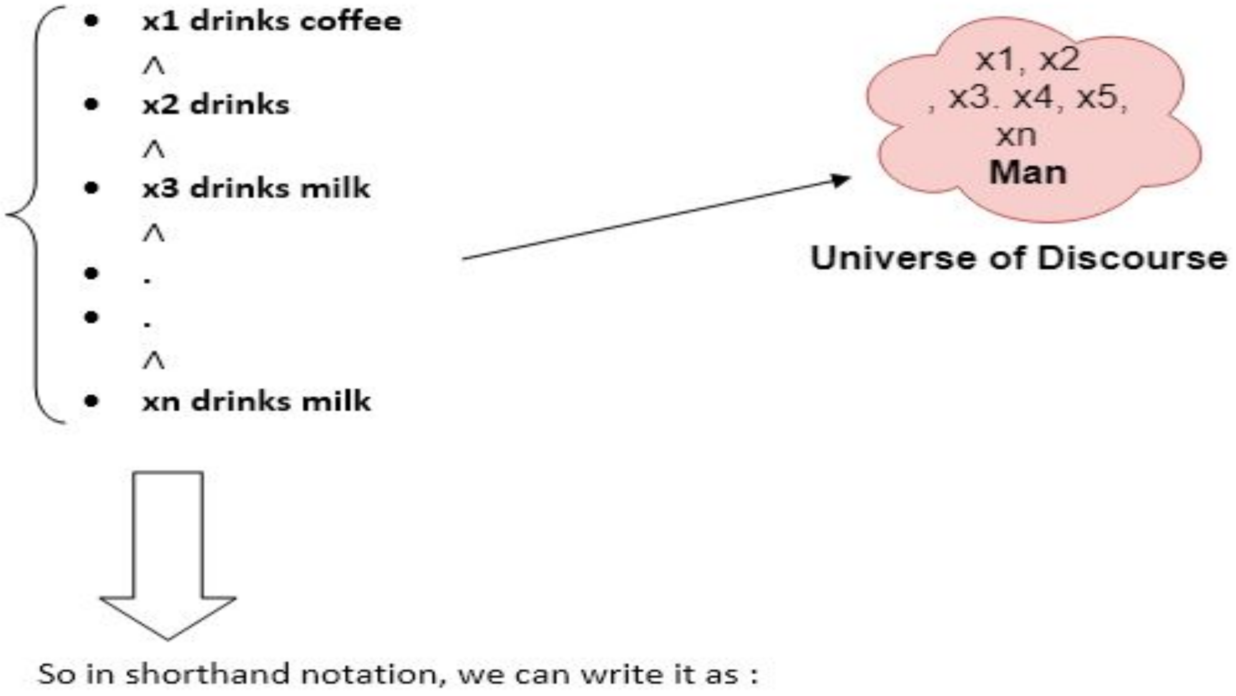
Universal Quantifier:

- 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 $\forall$, which resembles an inverted A.
- If x is a variable, then $\forall x$ is read as:

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

Example:

- **All man drink coffee.**
- Let a variable x which refers to a cat so all x can be represented in UOD as below



- **$\forall$ x man(x) → 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 ∃x or ∃(x). And it will be read as:

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

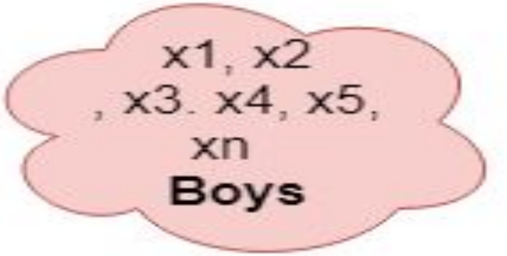
- Example:
- **Some boys are intelligent**

- **x1 is intelligent**

  V
- **x2 is intelligentV**
- **x3 is intelligent**

  V
- **.**
- **.**

  V
- **xn is intelligent**



Universe of Discourse

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