Subject Name: **Artificial Intelligence**

Subject Code: **IT-7005**

Semester: **7th**

Unit II

Knowledge Representation

Knowledge is the information about a domain that can be used to solve problems in that domain.
The objective of knowledge representation is to express the knowledge about the world in a computer-tractable form. The Knowledge Representation Framework is illustrated below.

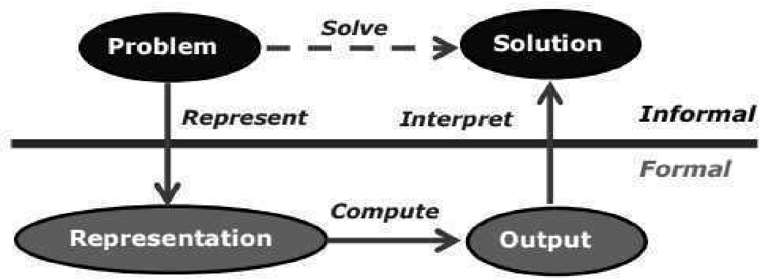

Fig 2.1 Knowledge Representation Framework

The steps are
− The informal formalism of the problem takes place first.
− It is then represented formally and the computer produces an output.
− This output can then be represented in a informally described solution that user understands or checks for consistency.

Knowledge is a progression that starts with data which is of limited utility.
By organizing or analyzing the data, we understand what the data means, and this becomes information.
The interpretation or evaluation of information yield knowledge.
An understanding of the principles embodied within the knowledge is wisdom.

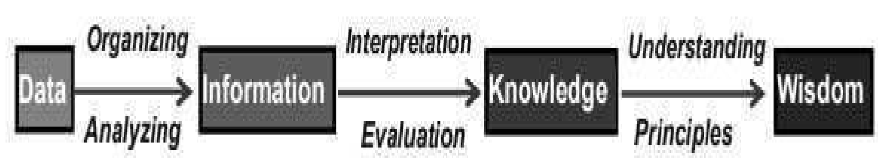

Fig 2.2 Knowledge progression

A knowledge representation language is defined by two aspects:

- Syntax: The syntax of a language defines which configurations of the components of the language constitute valid sentences.
- Semantics: The semantics defines which facts in the world the sentences refer to, and hence the statement about the world that each sentence makes.

Challenges of Knowledge representation:

- representation of commonsense knowledge
- the ability of a knowledge-based system to tradeoff computational efficiency for accuracy ofinferences

- Its ability to represent and manipulate uncertain knowledge and information.

The four fundamental components of a good representation:-

1. The lexical part – that determines which symbols or words are used inthe representation'svocabulary.
2. The structural or syntactic part – that describes the constraints on how the symbols can be arranged, i.e. agrammar.
3. The semantic part – that establishes a way of associating real world meanings with the representations.
4. The procedural part – that specifies the access procedures that enables ways of creating and modifying representations and answering questions using them, i.e. how we generate and compute things with the representation.

Problems in representing knowledge

The fundamental goal of Knowledge Representation is to facilitate inferencing (conclusions) from knowledge. The issues are explained below:-

1. Important Attributes
2. Relationship among attributes
3. Choosing Granularity
4. Set of objects
5. Finding Right structure

1. Important Attributes

There are two attributes "instance" and " IsA ", that are of general significance. These attributes are important because they support property inheritance.

Example : A simple sentence like "Joe is a musician".

- Here "is a" (called IsA) is a way of expressing what logically is called a class-instance relationship between the subjects represented by the terms "Joe" and "musician".
- "Joe" is an instance of the class of things called "musician".
- "Joe" plays the role of instance," musician" plays the role of class in that sentence.
- This is specified as:
  [Joe] IsA [Musician]
  i.e., [Instance] IsA [Class]

2. Relationship among attributes

The attributes we use to describe objects are themselves entities that we represent. The relationship between the attributes of an object, independent of specific knowledge they encode, may hold properties like: Inverses , existence in an isa hierarchy , techniques for reasoning about values and single valued attributes.

3. Choosing Granularity

- At what level of detail should the knowledge be represented?
- At what level should the knowledge be represented and what are the primitives?"
- Should there be a small number or should there be a large number of low-level primitives or High-level facts.
- High-level facts may not be adequate for inference while Low-level primitives may require a lot of storage.

Example of Granularity:
Facts: John spotted Sue.
– This could be represented as
Spotted (agent(John), object (Sue))

4. **Set of objects**
There are certain properties of objects that are true as member of a set but not as individual;
Example: Consider the assertion made in the sentences:
"there are more sheep than people in Australia", and
"English speakers can be found all over the world."
To describe these facts, the only way is to attach assertion to the sets representing people, sheep, and English.
The reason to represent sets of objects is: If a property is true for all or most elements of a set, then it is more efficient to associate it once with the set rather than to associate it explicitly with every elements of the set . This is done,
– in logical representation through the use of universal quantifier, and
– in hierarchical structure where node represent sets and inheritance propagate set level assertion down to individual.

5. **Finding Right structure**
This is about access to right structure for describing a particular situation. This requires, selecting an initial structure and then revising the choice. While doing so, it is necessary to solve following problems:
   - How to perform an initial selection of the most appropriate structure.
   - How to fill in appropriate details from the current situations.
   - How to find a better structure if the one chosen initially turns out not to be appropriate.
   - What to do if none of the available structures is appropriate.
   - When to create and remember a new structure.

Knowledge representation using Propositional and Predicate logic

Propositional Logic
A proposition is a statement, which in English would be a declarative sentence. Every proposition is either TRUE or FALSE. Propositional logic is fundamental to all logic. Propositional logic is also called Propositional calculus, sentential calculus, or Boolean algebra. Propositional logic tells the ways of joining and/or modifying entire propositions, statements or sentences to form more complicated propositions, statements or sentences, as well as the logical relationships and properties that are derived from the methods of combining or altering statements.

The symbols used in propositional logic are:-

∀ Universal quantifier (for all)

∃ Existential quantifier (there exists)

→ Implication

∧ And

∨ Or

¬ Not

Examples:

- Sachin is a cricketer
- Saina is a badminton player
- Saniya is a tennis player
- Jay is a badminton player or a tennis player.
- If Sachin is a cricketer, he is a sportsperson

We can represent them as follows using propositional logic

- SachinCricketer
- SainaBadminton
- SaniyaTennis
- JayBadminton V JayTennis
- SachinCricketer→SachinSportsPerson

Predicate Logic

Predicate allows us to reason about properties of objects and relationships between objects.
In propositional Logic, we could express the English statement “I like cheese” by A. This enables us to create constructs such as ¬A, which means “I do not like cheese,” but it does not allow us to extract any information about the cheese, or me, or other things that I like.

Need for Predicate logic

If we have one more statement, “all cricketers are rich” in our kitty, can we prove Sachin to be rich? It is hard unless we will try a little different way of representing the statements.
For example, we can represent them using first order predicate logic.

1. Cricketer (Sachin)
2. Badminton (Saina)
3. Tennis (Saniya)
4. Badminton (Jay) V Tennis (Jay)

The universal quantifiers are handy when we want to have statements like “All cricketers are rich” and use them for implication as follows.

5. $\forall x$Cricketer(x) -> Rich (x)

In the above statement X is known as a variable which can assume values like Jay and Sachin. Now we can combine 5 and 1 with X = Sachin and can prove Rich (Sachin). This is the power of first order predicate logic. The statements Cricketer (Sachin) are known as predicates. The word Cricketer is a name of that predicate and Sachin is the argument of that predicate.

Resolution

Resolution is a procedure used in proving that arguments which are expressible in predicate logic are correct. Resolution is a procedure that produces proofs by refutation or contradiction. Resolution leads to refute a theorem-proving technique for sentences in propositional logic and first-order logic.

- Resolution is a rule of inference.
- Resolution is a computerized theorem prover.
- Resolution is so far only defined for Propositional Logic. The strategy is that the Resolution techniques of Propositional logic be adopted in Predicate Logic.

Resolution Proof Example

(a) Marcus was a man.

(b) Marcus was a Roman.

(c) All men are people.

(d) Caesar was a ruler.

(e) All Romans were either loyal to Caesar or hated him (or both).

(f) Everyone is loyal to someone.

(g) People only try to assassinate rulers they are not loyal to.

(h) Marcus tried to assassinate Caesar.

Convert to First order Logic

(a) man(marcus)

(b) roman(marcus)

(c) ∀X. man(X) ∀person(X)

(d) ruler(caesar)

(e) ∀X. roman(x) ->loyal(X,caesar)

^hate(X,caesar)

(f) ∀X∀Y. loyal(X,Y)

(g) ∀X∀Y. person(X) ^ruler(Y)

tryassasin(X,Y) -> loyal(X,Y)

(h) tryassasin(marcus,caesar)

Convert to Clausal Form

1. man(marcus)
2. roman(marcus)
3. (¬man(X), person(X))
4. ruler(caesar)
5. (¬roman(X), loyal(X,caesar), hate(X,caesar))
6. (loyal(X,f(X))
7. (¬ person(X), ¬ruler(Y),¬tryassasin(X,Y), ¬loyal(X,Y))
8. tryassasin(marcus,caesar)

Comparison of Propositional and Predicate logic

Propositional Logic	Predicate Logic
1.Propositional logic (also called sentential logic) is logic that includes sentence letters (A,B,C) and logical connectives, but not quantifiers. 2. The semantics of propositional logic uses truth assignments to the letters to determine whether a compound propositional sentence is true. 3. Propositional logic means without ability to do predication. For example, in P ^ Q, both p and q are propositions.	1. Predicate logic is usually used as a synonym for first-order logic, but sometimes it is used to refer to other logics that have similar syntax. It also has variables for individual objects, quantifiers, symbols for functions, and symbols for relations. 2. The semantics include a domain of discourse for the variables and quantifiers to range over, along with interpretations of the relation and function symbols. 3. Predicate logic is the general form of all logics that uses predicates, like *q(x).* For example, ∀x∃y.p(x,y) means "For all x there exists a y such that the proposition p(x,y) is true".

Theorem proving

Reasoning by theorem proving is a weak method, compared to experts systems, because it does not make use of domain knowledge. This, on the other hand, may be strength, if no domain heuristics are available (reasoning from first principles). Theorem proving is usually limited to soundreasoning.

- theorem proverbs: fullyautomatic
- Proof assistants: require steps as input; take care of bookkeeping and sometimes 'easy' proofs.

Theorem proving requires

- a logic(syntax)
- a set of axioms and inferencerules

- a strategy on when how to search through the possible applications of the axioms

Examples:-

p -> (q->p)

(p->(q->r)) -> ((p->q) ->(p->r))

p \/ ~p

p->(~p->q)

Notation: I use ~ for "not"

Inferencing

An Inference Engine is a tool from artificial intelligence. The first inference engines were components of expert systems. The typical expert system consisted of a knowledge base and an inference engine. The knowledge base stored facts about the world. The inference engine applied logical rules to the knowledge base and deduced new knowledge. This process would iterate as each new fact in the knowledge base could trigger additional rules in the inference engine. Inference engines work primarily in one of two modes either special rule or facts: forward chaining and backward chaining. Forward chaining starts with the known facts and asserts new facts. Backward chaining starts with goals, and works backward to determine what facts must be asserted so that the goals can be achieved.

Forward Chaining: Conclude from "A" and "A implies B" to "B.”

A

A -> B

B

Example:

It is raining.

If it is raining, the street is wet.

The street is wet.

Backward Chaining: Conclude from "B" and "A implies B" to "A.”

B

A -> B

A

Example:

The street is wet.

If it is raining, the street is wet.

It is raining.

Monotonic Reasoning

In monotonic systems there is no need to check for in consistencies between new statements and the old knowledge. When a proof is made, the basis of the proof need not be remembered, since the old statements never disappear. But monotonic systems are not good in real problem domains where the information is incomplete, situations change and new assumptions are generated while solving new problems.

- A reasoning process that moves in one directiononly.
- The number of facts in the knowledge base is alwaysincreasing.
- The conclusions derived are valid deductions and they remainso.

Non-Monotonic Reasoning

Non monotonic reasoning is based on default reasoning or "most probabilistic choice". S is assumed to be true as long as there is no evidence to the contrary. For example when we visit a friend's home, we buy biscuits for the children .because we believe that most children like biscuits. In this case we do not have information to the contrary. A computational description of default reasoning must relate the lack of information on X to conclude on Y.

- Non-monotonic reasoning (NMR) is based on supplementing absolute truth withbeliefs.
- These tentative beliefs are generally based on default assumptions that are made in light of lack ofevidence.
- A non-monotonic reasoning (NMR) system tracks a set of tentative beliefs and revises those beliefs when knowledge is observed orderived.
- The reason is, the human reasoning is non-monotonic innature.
- This means, we reach to conclusions from certain premises that we would not reach if certain other sentences are included in ourpremises.
- Conventional reasoning systems such as first order predicate logic are designed to work with information that has three importantproperties.
  1. It is complete with respect to the domain ofinterest.
  2. It isconsistent.
  3. The only way it can change is that new facts can be added as they become available. If the new facts are consistent with all other facts that have already been asserted, then nothing will be ever retracted from the set of facts that are known to be true. This property is calledmonotonicity.

