Knowledge Base Components:

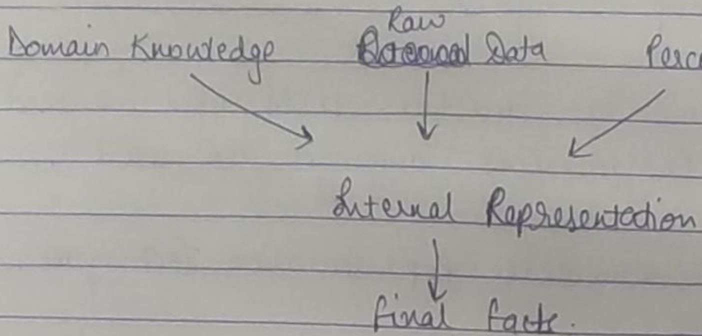
① Factual Knowledge: widely accepted by Engineers, Scholars.

② Heuristic Knowledge: Practice, accurate judgement & one's ability of guessing.

Knowledge representation: Either using Rules or logic.

Rules → If - then - else rules.

Limited / Partial Info in the beginning and then later on, info. is evolved



Approaches of KR:

① While designing a system, we would go for a system that allows representation of entire knowledge.

② Knowledge must be efficient.

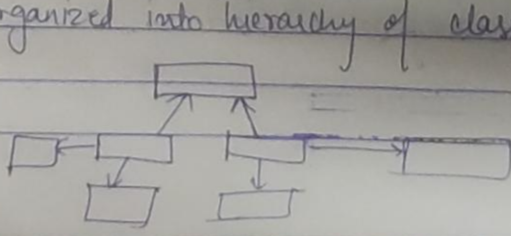
③ The Representation needs to be adaptive and also available at any point of time.

Approaches

Relational Knowledge Structure: facts are mapped into relations and stored in the database

Inheritable Knowledge Structure: Elements inherit values by being a member of the class.

Data must be organized into hierarchy of classes. Also known as slot-filler structure



Inferential Knowledge Structure: First order logic is used for better utilization of knowledge.

eg. If animal has hair then species is mammal — $\forall x$ Has hair$(x) \Rightarrow$ Mammal(x)

If animal is mammal and animal has hooves, then Mammal group is ungulate. — $\forall x$ Mammal $\wedge$ Has hooves$(x) \Rightarrow$ Ungulated (x)

Procedural Knowledge Structure: used when we need detailed information. Programming languages are used.

(4) Issues with Knowledge Representation Techniques:

→ Important attributes :- Most attributes have an impact on the information. It is necessary to identify these attributes.

→ Relationship among attributes: Identifying the relationship among attributes is equally Important.

→ Choosing Granularity: It is difficult to choose upto what level of detail should the knowledge be represented?

→ Representing Sets of Objects: If a property is true for most or all of the elements of the set, then it is better to associate it once with the set rather than associating it with single elements of the set.

→ Finding the right structures as needed: Given a large amount of knowledge stored in the database, how can relevant parts be accessed when they are needed?

* **Knowledge Based Agent**: Agents who have the capability of maintaining Knowledge, reason over that knowledge, update the knowledge after observations and take actions.

These agents are composed of → Knowledge base
Inference System.

The agent → tells the knowledge base what it percieves.
Asks the knowledge base what action it should perform
Takes action.

(*) **Logic**

(*) → **Propositional Logic**: Simplest. Works on 0 & 1, also known as Boolean Logic.
→ Symbols represent facts and they are ~~connected~~ joined by logical connectives. eg: $P \wedge Q$; $Q \Rightarrow R$
→ Some statements are given and we can deduce facts using these.

$S1 \wedge S2$ → Conjunction
$S1 \vee S2$ → Disjunction
$S1 \Rightarrow S2$ → Implication
$S1 \Leftrightarrow S2$ → Biconditional.

Syntactic Properties.

Semantic Properties-
→ $S1 \wedge S2$ is true iff $S1$ is true & $S2$ is true.
$S1 \vee S2$ is true if $S1$ is true or $S2$ is true
$S1 \Rightarrow S2$ is true iff $S1$ is false or $S2$ is true.
$S1 \Leftrightarrow S2$ is true iff $S1 \Rightarrow S2$ is true and $S2 \Rightarrow S1$ is true.

Inference Properties:

P	Q	P → Q	Q → P
T	T	T	T
T	F	F	T
F	T	T	F
F	F	T	T

(*) Predicate logic: also called first order logic.
An extension to propositional logic.
Develops info. about the objects and expresses relationship b/w them.

Eg: ① Marcus was a man ⇒ man(Marcus)
② All cars are red ⇒ $\forall x$ Cars(x) $\rightarrow$ Red(x)
③ All Romans were either loyal to Caesar or hated him

⇓

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

ⓐ Universal Quantifiers (for all, for each, for every x)
In UQ, we use implication "$\rightarrow$"
eg. All men drink coffee

⇓

$\forall x$ men(x) $\rightarrow$ drink(x, coffee)

ⓑ Existential Quantifiers: (for some x, there exist a x, for atleast one x).
denoted by $\exists(x)$
Eg: Some boys are intelligent
$\exists x$: boys(x) $\wedge$ intelligent(x)

eg → All birds fly $\forall x$ bird(x) $\rightarrow$ fly(x)

Syntax & Semantics:
① Variables → denoted by lower case letters
② Constants → denoted by upper case letters.
③ Connectives → ($\wedge, \vee, \sim, \rightarrow, \leftrightarrow$)
④ Quantifiers.
Is relationship → relates a subject & object (represented using binary relationship)
instance → describes property of object (unary)

Unification:

→ The process of making two different statements look identical by the process of substituting.
→ It takes two literals as input and makes them look identical by substitution.

Eg: $P(x, f(y))$ — ①
$P(a, f(g(z)))$ — ②

$$\boxed{\begin{array}{l} x = a \\ y = g(z) \end{array}}$$

Unification: $[a/x, g(z)/y]$

Substitute x with a and y with g(z).
With both the substitutions, the first expression will be identical to the 2nd expression and the substitution set ⇒ $[a/x, g(z)/y]$

Conditions for unification:

1. Predicate symbols must be the same. Expressions with different predicate symbols can never be unified
2. No. of arguments in both expressions must be same.
3. Unification will fail if there are two similar variables present in same expression.

Algo: unify(L_1, L_2)

1. If L_1 or L_2 is a variable/constant, → if $L_1 = L_2$, return nil
   - If L_1 occurs in L_2 return fail
   - else return L_2/L_1

   Repeat same for L_2.
2. If predicate is not same, return fail.
3. If no of arguments not same, return fail.

④. Else substitute to w/l
⑤. loop
⑥. Return.

eg: $\underbrace{Q(a, g(x, a), f(y)) \; ; \; Q(a, g(f(b), a), x)}$

①. substitute x with $f(b)$. ~~same substitute~~ in both.
$[f(b)/x]$
⇓

$Q(a, g(f(b), a), f(y)), \; Q(a, g(f(b), a), f(b))$

②. Substitute $[b/y]$

$Q(a, g(f(b), a), f(b)), \; Q(a, g(f(b), a), f(b))$

eg②. $P(x, g(x)), \; P(\text{prime}, f(\text{prime}))$

We cannot unify because in 1st statement we are using function of g and in 2nd statement we are using function of f.

(*) · Resolution:

→ A theorem proving technique that proves by contradiction.
→ It is used when there are various statements given and we need to prove conclusion of those statements.
→ Unification is a key concept in proofs by resolution.
→ We use both Propositional as well as first order logic in different ways.
→ Resolution can efficiently operate on Conjunctive normal form (CNF) or Clausal form.

Steps. ①. Convert the facts into first order logic.
②. Convert FOL statements to CNF.
③ Negate the statement that has to be proved.
④. Draw resolution graph

eg: ①. If it is sunny & ~~[illegible]~~ warm day you will enjoy.
②. If it is raining you will get wet.
③. It is a warm day
④. It is raining
⑤ It is sunny.

Prove → You will enjoy.

Step ① → ①. sunny ∧ warm → enjoy.
② raining → wet
③ warm
④ raining
⑤ Sunny

Step 2: Remove → symbol $a \to b \Rightarrow \sim a \vee b$.

① . $\sim(\text{Sunny} \wedge \text{warm}) \vee \text{Enjoy}$

② . $\sim \text{raining} \vee \text{wet}$

Step 3 $\sim \text{enjoy}$

Step 4 $\sim \cancel{\text{enjoy}}$ $\sim \text{sunny} \vee \sim \text{warm} \vee \cancel{\text{enjoy}}$

$\sim \text{sunny} \vee \sim \cancel{\text{warm}}$ $\cancel{\text{warm}}$

$\sim \cancel{\text{sunny}}$ $\cancel{\text{sunny}}$

$\{=\}$ Contradiction Proved.

$\{\phi\}$.

15. What are the different types of inferences?

Ans : Inferences are the process of drawing logical conclusions from premises or evidence. In the context of artificial intelligence and knowledge representation, there are several types of inferences that can be made, including:

1. Deductive inference: Deductive inference is the process of drawing logical conclusions from a set of premises using logical rules of inference. In deductive inference, the conclusion is guaranteed to be true if the premises are true.

2. Inductive inference: Inductive inference is the process of drawing general conclusions from specific observations or evidence. Inductive inference is based on probability and is not guaranteed to be true, but can be highly probable based on the available evidence.

3. Abductive inference: Abductive inference is the process of generating hypotheses or explanations to explain observed phenomena. Abductive inference is based on the principle of selecting the most likely explanation based on the available evidence.

4. Analogical inference: Analogical inference is the process of drawing conclusions by comparing similarities between two or more situations or objects. Analogical inference is often used to transfer knowledge from one domain to another.

5. Default inference: Default inference is the process of making assumptions or drawing conclusions based on default values or assumptions. Default inference is often used in

Uncertain Knowledge and Reasoning :-

Uncertain data → noisy data / missing data / unreliable data / inconsistent

Uncertain Knowledge ⇒ Lack of exact info. or knowledge that helps us to find conclusion or correct solⁿ

In such situation the agent does not guarantee a solⁿ but acts on its own assumption and probablities.

Uncertain situations can be dealt with using probability

(i) Probablity Logic

(ii) Fuzzy Logic ⇒ method of reasoning that resembles human reasoning.

(iii) Possibility Theory — mathematical theory for dealing with certain type of uncertainity → Alternative to probablity theory

(iv) Bayesian n/w ⇒ graphical model that represent uncertain knowledge using probablity distribution

(v) Dempster Shafer Theory.

⇓

→ able to deal with missing info.

→ generalization of Bayesian theory.

Reasoning :- S/w systems that generates conclusion from available knowledge using logical techniques such as deduction and induction.

12. How is knowledge represented using rules?

Ans : In artificial intelligence, knowledge can be represented using rules. A rule is a statement that asserts a relationship between two or more logical expressions. It has two parts: a premise (also known as antecedent) and a conclusion (also known as consequent). The premise specifies the conditions under which the rule is applicable, while the conclusion specifies the action or inference that can be drawn from the premise.

For example, consider the following rule:

If it is raining outside, then I will bring an umbrella.

In this rule, "it is raining outside" is the premise, and "I will bring an umbrella" is the conclusion. This rule can be used to make inferences about whether or not the speaker will bring an umbrella given a certain condition (i.e., whether it is raining outside).

In knowledge representation using rules, a set of rules is often used to represent a domain of knowledge. The rules can be organized into a knowledge base, which can be used to make inferences about the domain.

For example, consider a simple knowledge base about birds:

Rule 1: If a bird has wings and can fly, then it is a bird of prey. Rule 2: If a bird is a bird of prey and has sharp talons, then it is an eagle. Rule 3: If a bird is a bird of prey and has curved beak, then it is a hawk.

This knowledge base can be used to make inferences about whether or not a certain bird is an eagle or a hawk given certain conditions about its wings, ability to fly, sharp talons, and curved beak. By applying the rules in the knowledge base, we can make logical deductions that allow us to draw conclusions about the domain of knowledge.

13. How is knowledge represented using semantic nets?

Ans : Semantic nets, also known as concept maps or graphical knowledge representations, are a type of knowledge representation that uses nodes and links to represent knowledge in a graphical form. In a semantic net, nodes represent concepts or objects, and links represent relationships between them. Semantic nets are often used in artificial intelligence and cognitive science to represent knowledge in a human-readable form.

The following are the basic components of a semantic net:

1. Nodes: Nodes are graphical symbols that represent concepts or objects. For example, a node may represent a person, a place, or an event. Nodes are often labeled with a name or a symbol that represents the concept or object they represent.
2. Links: Links are graphical lines that connect nodes and represent relationships between them. Links can be labeled with a name or a symbol that represents the type of relationship between the nodes. For example, a link may represent a "part-of" relationship, an "is-a" relationship, or a "causes" relationship.
3. Attributes: Attributes are additional information associated with nodes and links. Attributes can be used to provide additional information about the concepts or objects represented by nodes or the relationships between them.

Semantic nets can be used to represent a wide range of knowledge domains, including scientific concepts, medical knowledge, and engineering principles. They can also be used to represent natural language text, allowing computers to understand the meaning of sentences and paragraphs.

For example, consider the following semantic net that represents knowledge about a dog:

[Dog] --(is-a)--> [Animal] [Dog] --(has-a)--> [Tail] [Dog] --(has-a)--> [Fur] [Dog] --(eats)--> [Food]

In this example, the node labeled "Dog" represents the concept of a dog, while the links represent the relationships between the dog and other concepts, such as animals, tails, fur, and food. This semantic net can be used to represent a wide range of knowledge about dogs, including their physical characteristics, their dietary habits, and their relationships with other animals.

14. How is knowledge represented using frames?

Ans : Frames are a type of knowledge representation that organizes knowledge into structured objects called "frames". A frame is a data structure that contains a set of slots that represent attributes or properties of an object, as well as values associated with those slots. Frames can be used to represent objects, concepts, or situations in a domain of knowledge.

Frames consist of the following components:

1. Frame name: The name of the frame, which identifies the type of object or concept being represented.
2. Slots: The slots represent the attributes or properties of the frame. Each slot has a name and a value associated with it. Slots can have default values or may be left empty until filled with specific values.
3. Inheritance: Frames can be organized into hierarchies, with child frames inheriting properties and attributes from their parent frames.

For example, consider the following frame that represents a person:

[Person]

- Name: (default = "")
- Age: (default = 0)
- Gender: (default = "")
- Address: (default = "")
- Phone: (default = "")

In this example, the frame name is "Person", and the slots represent the attributes or properties of a person, such as name, age, gender, address, and phone number. Each slot has a default value that can be overridden with a specific value for a specific person.

Frames can be used to represent a wide range of knowledge domains, including biology, medicine, engineering, and economics. They can be used to represent complex objects or concepts that have many attributes or properties, and they can be organized into hierarchies to represent the relationships between different types of objects or concepts. Frames can also be used in expert systems and other artificial intelligence applications to represent and reason about knowledge in a domain.

Forward chaining	Backward chaining
(I) starts from known facts and applies inference rules to extract more data until it reaches goal state	(I) starts from goal and works backward through inference rules to find required fact that support the goal.
(II) Bottom-up approach	Top-Down
(III) Also called data-driven inference as we reach the goal using available data.	goal-driven as we start from goal and divide into sub-goal to extract the facts.
(IV) Applies Breadth-first search strategy.	Applies depth-first search strategy
(V) Can generate infinite no. of possible conclusions.	finite
(VI) operates in forward direction	Backward
(VII) Aim for any conclusion	Aim for required data
(VIII) checks all available rules	checks for few required rules.

Bayes Theorem

Acc. to cond^n probability,

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

prob. of event A given that B has already occurred

$$P(B|A) = \frac{P(B \cap A)}{P(A)}$$

$P(A \cap B) = P(B|A) \cdot P(A)$

$$\boxed{P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}}$$

Bayes's Theorem

A: hypothesis; B: evidence/data

$P(A|B)$: Posterior Prob.

$P(B|A)$: Likelihood P. (prob. of evidence)

$P(A)$: Prior P.

$P(B)$: Marginal P.

Compute entity of Bayes Th.

$$P(\text{King}|\text{Face}) = \frac{P(\text{face}|\text{king}) \cdot P(\text{King})}{P(\text{face})}$$

$$= \frac{1 \cdot \frac{4}{52}}{\frac{12}{52}}$$

$$= \frac{1 \cdot 1/13}{3/13} = \frac{1}{3} \text{ Ans.}$$