UNIT-4
Knowledge Representation

Knowledge → Knowledge is the information about a domain that can be used to solve problems in that domain. OR It is the sum of what is currently known. It consists of information that has been integrated, categorised, applied, experienced, revised etc. It is not just a data it also consists of facts, ideas, beleaves, association rules, relationships etc.

Importance of Knowledge: (In AI)

i) Knowledge is important in problem solving by computer.

ii) It is possible for an agents or systems to act accurately on some input only when it has the knowledge or experience about the input.

Issue in knowledge Representations:

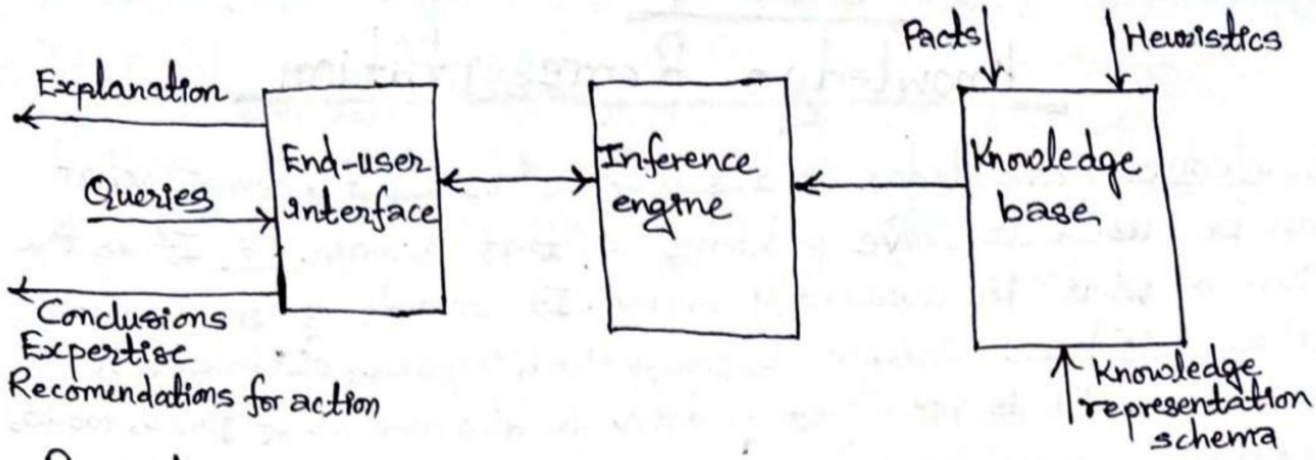
The process of converting informal knowledge into computer understandable form is known as knowledge representation. To solve any problem by a computer, first it must be represented in computer understandable form. Some of the issues regarding knowledge representation can be formulated as questions as follows:-

→ How can the problem be represented?
→ What specific knowledge about the world is required?
→ How can an agent acquire the knowledge from experts or from experience?
→ What distinctions in the world are needed to solve the problem?
→ How can the knowledge be debugged, maintained and improved?

Knowledge Representation System and its Properties:-

Knowledge Representation system is a formalized structure and sets of operations that contains the descriptions, relationships and procedures provided in an AI system. The process considers how to represent the knowledge in the system such that it can be used to solve problems. It basically consists of two components:

1) Knowledge base
2) Inference Methods.



Properties:-

i) Representatn adequacy:- Knowledge Representation system has the ability to represent all kinds of required knowledge.

ii) Inferential adequacy:- It is the ability to manipulate the knowledge represented to produce new knowledge corresponding from original knowledge.

iii) Inferential efficiency:- It is the ability to direct the inferential mechanisms ~~into~~ the most strong appropriate guides.

iv) Acquisitional efficiency:- It is the ability to acquire new knowledge using automatic methods wherever possible rather than reliance on human intervention.

⊛. Types of Knowledge Representation Systems:-

1) Semantic Nets:- A sementic nets or network allows us to store knowledge in the form of a graphic (in the form of graphs) with nodes and arcs representing objects and their relationships. It could represent physical objects or concepts or even situations.

This is simple and easy to implement and understand. It is more natural than logical representation. It allows us to categorize objects in various forms and link those objects.

Components of a Semantic Network:

i) Lexical part

nodes → denoting objects
links → denoting relation between objects
labels → denote particular objects and relations.

ii) structural part: the links and nodes from directed graphs. The labels are placed on the links and nodes.

iii) <u>Semantic part</u> → Meanings are associated with the link and node lables.

iv) <u>Procedural part</u> →

constructors → allow creation of new links and nodes.
destructors → deletion of links and nodes
writers → allow creation and alternation of labels.
readers → can extract answers to questions.

Types of Semantic Network:

i) <u>Definitional networks</u> → These are designed to show relation between a concept type and a newly defined subtype. It supports the rule of inheritance for copying properties defined for a supertype to all of it's subtypes.

ii) <u>Assertional networks</u> → These are designed to assert positions. Unlike definitional networks, the information in an assertional network is assumed to be contingently true, unless it is explicitly marked with a modal operator.

iii) <u>Implicational networks</u> → These use implication as the primary relation for connecting nodes. They may be used to represent pattern of beliefs, causality or inferences.

iv) <u>Learning networks</u> → These are used to build or extend their representations by acquiring knowledge from examples.

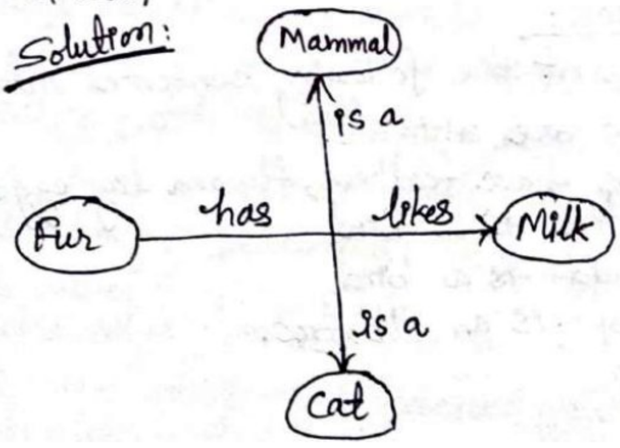
v) <u>Hybrid networks</u> → Combines two or more of the previous techniques, either in a ~~sig~~ single network or seperate, but closely interacting networks.

Examples of Semantic Network: (only for understanding)

i) <u>General Example:</u> Represent following statements in figure.

→ cat is a mammal.
→ cat likes milk
→ cat has fur
→ tommy is a cat.

<u>Solution:</u>



ii) AND/OR tree Example:-

AND/OR tree → It is an assignment of "True" or "False" to each of the leaves of tree.

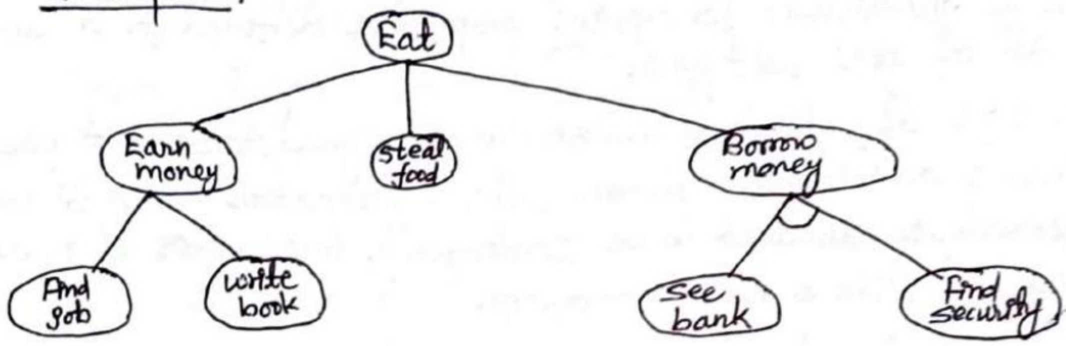
→ An OR node is true if at least one of it's children is true.

→ An AND node is true if all of it's children are true.

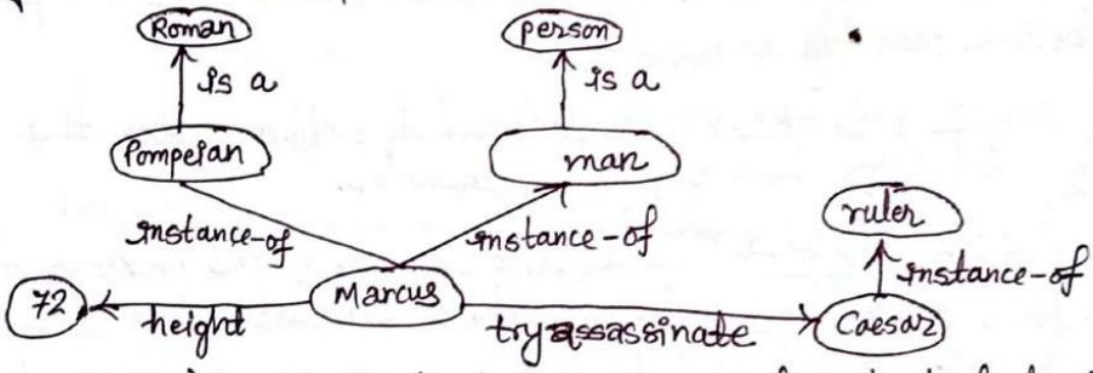
Nodes:

representation of AND node. representation of OR node.

Example:



iii) Binary relations example:



(Marcus, Caesar) or constants (72) → represent individual objects

(Pompeian, Roman, man, person, ruler) → represent classes of individuals

instance-of → represents element of a class.

is a → represents subclass of a class.

tryassassinate → represents tried to assassinate

⊛. Solved Examples:-

Example 1: Represent the following sentences into a semantic network.

→ Birds are animals.

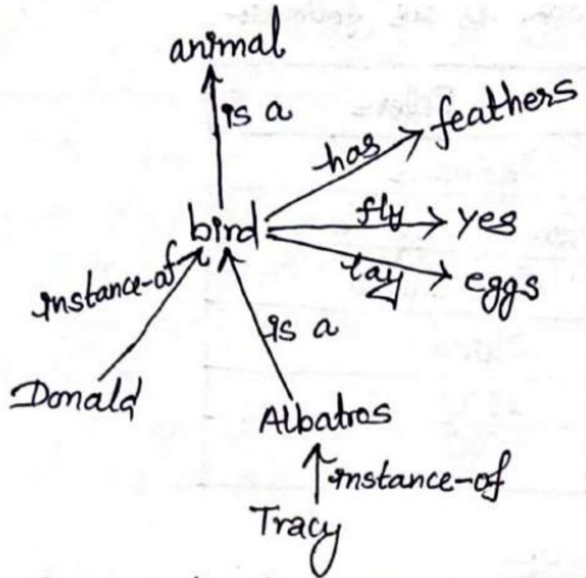
→ Birds have feathers, fly and lay eggs.

→ Albatros is a bird.

→ Donald is a bird.

→ Tracy is an albatros.

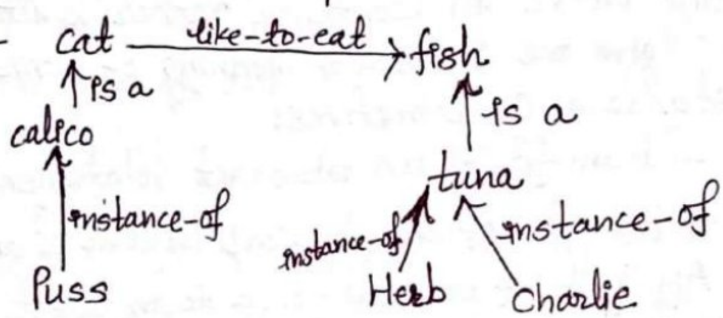
Solution:



Example 2: Represent the following sentences into a semantic network:

→ Puss is a calico.
→ Herb is a tuna
→ Charlie is a tuna
→ All tunas are fishes.
→ All calicos are cats.
→ All cats like to eat all kinds of fishes.

Solution:



2) Frame based Knowledge Representations:-

Frame is a type of schema used in many AI applications including vision and natural language processing. Frames provide a convenient structure for representing objects that are typical to a sterotypical situations like visual scenes, structure of complex physical objects etc.

Frame systems usually have collection of frames connected to eachother. Each frame has a name and slots. Slots are the properties of the entity that has the name, and they have values or pointer to other frames. When the slots of a frame are all filled, the frame is said to be instantiated, it now represents a specific entity of the type defined by the unfilled frame. Empty frames are sometimes called object prototypes.

Slots having particular value may be:

→ a default value.
→ an inherited value from a higher frame.
→ a specific value which might represent an exception.

Example:- A frame for a book is as follows:-

Slots	Fillers
publisher	Thomson
title	Expert Systems
author	Giarranto
edition	Third
year	1998
pages	600

3) Conceptual Dependencies:-

CD theory was developed to represent the meaning of Natural Language sentences. It helps in drawing inferences and and is independent of the language. CD representation of a sentence is not built using words in sentence, rather built using conceptual primitives which give the intended meaning of words. Following are some of the standard CD primitives:

i) ATRANS → transfer of an abstract relationship (i.e, give).
ii) PTRANS → transfer of the physical location of an object (e.g, go).
iii) PROPEL → Application of physical force to an object (e.g, push).
iv) MOVE → Movement of a body part by it's owner (e.g, kick).
v) GRASP → Grasping of an object by an action (e.g. throw).
vi) INGEST → Ingesting of an object by an animal (e.g. eat).
vii) EXPEL → Expulsion of something from the body of an animal (e.g, cry)
viii) MTRANS → Transfer of mental information (e.g, tell).
ix) SPEAK → Producing of sounds (e.g. say).
x) ATTEND → Focusing of a sense organ toward a stimulus (e.g, listen).

There are four conceptual categories:

ACT → Actions {one of the CD primitives}
PP → Objects {picture producers}
AA → Modifiers of actions {action aiders}
PA → Modifiers of PP's {picture aiders}

Example 1:

I gave a book to the man. CD representation is as follows:

$$I \overset{P}{\Longleftrightarrow} ATRANS \overset{O}{\longleftarrow} book \overset{R}{\longleftarrow} \begin{cases} \rightarrow man\ (to) \\ \leftarrow I\ (from) \end{cases}$$

where, → Arrows indicate directions of dependency.
→ Double arrow indicates two way link between actor and action.

O → for the object case relation.
R → for the recipient case relation.
P → for past tense.
D → Destination.

Rule 1: PP ⇔ ACT
⇒ It describes the relationship between an actor and the event he or she causes.

Example 2: John ran.

CD representation is: John $\overset{P}{\Leftrightarrow}$ PTRANS

Rule 2: ACT ← PP
⇒ It describes the relationship between ACT and a PP(object) of ACT.

Example 3: John pushed the bike.

CD representation is: John ⇔ PROPEL ← bike

Rule 3: PP ↔ PP
⇒ It describes the relationship between two PP's, one of which belongs to the set defined by the other.

Example 4: John is doctor.

CD representation is: John ↔ doctor

Rule 4: PP ← PP
⇒ It describes the relationship between two PP's, one of which provides a particular kind of information about the other.

Example 5: John's dog.

CD representation is: dog $\xleftarrow{\text{poss-by}}$ John

Rule 5: PP ⇔ PA
⇒ It describes the relationship between a PP and a PA that is asserted to describe it.

Example 6: John is fat.

CD representation is: John ⇔ weight (>80).

Rule 6: PP ← PA.
⇒ It describes the relationship between a PP and an attribute that already has been predicated of it.

Problems with CD representation:-

i) It is difficult to construct original sentence from it's corresponding CD representation.
ii) Rules are to be carefully designed for each primitive action in order to obtain semantically correct interpretation.
iii) The CD representation becomes complex requiring lot of storage for many simple actions.
iv) It is difficult to find correct primitive in the given context.

4) Scripts (Script Structure):

The scripts are useful in describing certain sterotyped situations such as going to theater. It consists of set of slots containing default values along with some information about the type of values similar to frames. The values of the slots in scripts must be ordered and have more specialized roles.

Script Components:

i) Entry Conditions → Must be satisfied before events in the script can occur.
ii) Results → Conditions that will be true after events in script occur.
iii) Props → Slots representing objects involved in the events.
iv) Roles → Persons involved in the events.
v) Track → Specific variation on more general pattern in the script. Different tracks may share many components of the same script but not all.
vi) Scenes → The sequence of events that occur.

5) Rule Based Knowledge Representation System (RBS):-

A rule-based system (or production system) is a Knowledge Based System (KBS) in which the language is stored as rules; the rules come from human experts in a particular domain. It represents knowledge in the form of condition-action pairs called production rules.

→ If the condition C is satisfied then the action A is appropriate.

Examples:

→ If it is raining then open the umbrella.
→ If Cesar is a man then Cesar is a person.

In RBS the knowledge is seperated from AI reasoning processes, which means that new RBSs are easy to create. The syntax of rules is

IF <premise> THEN <action>

Components of RBS:

i) Working Memory → Small memory into which only appropriate rules are copied.

ii) Rule base → To facilitate efficient access to the antecedent.

iii) Interpreter → Processing engine.

Example:- Production system for string sorting.

Problem: Sorting a string composed of letters a, b & c.

Short ~~Fem~~ Term Memory: cbaca

Production Set:

1. ba → ab
2. ca → ac.
3. cb → bc.

Interpreter: Choose one rule according to some strategy.

Solution:

Iteration #	Memory	Conflict Set	Rule fixed.
0	cbaca	1,2,3	1
1	cabca	2	2
2	acbca	2,3	2
3	acbac	1,3	1
4	acabc	2	2
5	aacbc	3	3
6	aabcc	∅	halt

6) Logic Based Knowledge Representations:-

@ Propositional Logic: (PL)

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

Example:

→ There are 7 days in a week (True Proposition)

→ 3+3=7 (False Proposition).

Note: Prepositions are either true or false but not both.

Syntax of proportional logic:

The syntax of proportional logic is defined by the allowable sentence (atomic & composite). The sentence which is indivisible is called atomic sentence which consist of single proportion (that can be either true or false). Combination of two or more than two atomic sentences by using logical connectives (like AND, OR, NOT) is called complex or composite sentence.

Now proportional logic can be defined as; read as not S

If S is sentence → $\neg S$ is sentence (negative).

If S_1 & S_2 are sentences → $S_1 \wedge S_2$ is sentence (conjunction)

$S_1 \vee S_2$ is sentence (disjunction)

$S_1 \to S_2$ is sentence (implication).

$S_1 \leftrightarrow S_2$ is sentence (double implication).

Semantic of proportional logic:

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \to Q$	$P \leftrightarrow Q$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

Formal grammar for proportional logic:

Sentence → Atomic sentence / Complex sentence.

Atomic sentence → T / F / Symbol

Symbol → P / Q / R / ...

Complex sentence → $\neg$Sentense / $S \wedge S$ / $S \vee S$ / $S \to S$ / $S \leftrightarrow S$

Logical Equivalence:

Two sentences α & β are logically equivalent ($\alpha = \beta$) iff they are true in same set of models.

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$ commutativity of $\wedge$

$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$ commutativity of $\vee$

$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha (\beta \wedge \gamma))$ associativity of $\wedge$

$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha (\beta \vee \gamma))$ associativity of $\vee$

$\neg(\neg \alpha) \equiv \alpha$ double-negation elimination.

$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$ contraposition.

$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$ implication elimination

$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$ biconditional elimination.

$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$ de Morgan

$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$ de Morgan

$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$ distributivity of $\wedge$ over $\vee$.

$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$ distributivity of $\vee$ over $\wedge$.

Validity in propositional logic:

A sentence is valid if it is true in all models. Valid sentences are also known as tautologies. Every valid sentence is logically equivalent to True.

e.g. $A \vee \neg A$

Q. Prove how $A \vee \neg A$ is valid? Use truth table.

Solution:

A	$\neg A$	$A \vee \neg A$
T	F	T
F	T	T

} valid proved.

Satisfiability in propositional logic:

→ A sentence is satisfiable if it is true in some model.

e.g. $A \vee B$

→ A sentence is unsatisfiable if it is true in no model.

e.g. $A \wedge \neg A$.

Inference in propositional logic:

To infer the knowledge from KB mostly we use modus ponens.

i.e, $\dfrac{\alpha \Rightarrow \beta, \alpha}{\beta}$ & Elimination $\dfrac{\alpha \wedge \beta}{\alpha}$.

The set of entailed sentence can only increase ~~added~~ the information added to the KB, this process is called monotocity.

Inference using resolution:

1> Unit resolution rule:

$$\frac{l_1 \vee \dots \vee l_{k1} \quad m}{l_1 \dots \vee l_{i-1} \vee l_{i+1} \dots \vee l_k}$$

where, l_i is literal.

2) Generalized resolution rule:

$$\frac{l_1 \vee \ldots l_{k_1} \quad m_1 \vee_1 \ldots \vee m_n}{l_1 \vee \ldots \vee l_{j-1} \vee l_{j+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n}$$

Resolution by using CNF:

A sentence that is expressed as conjunction of disjunctions of literals is said to be in conjunctive normal form (CNF). Process for converting KB in CNF is as follows:

1) Eliminate biconditional (i.e, $\Leftrightarrow$)
   e.g, $(B \leftrightarrow C) \equiv (B \rightarrow C) \wedge (C \rightarrow B)$
2) Eliminate implication (i.e, $\rightarrow$)
   e.g, $(B \rightarrow C) \equiv (\neg B \vee C)$
3) Move '$\neg$' inwards
   e.g, $\neg(A \wedge C) \equiv (\neg A \vee \neg C)$
4) Apply distributivity law ($\wedge$ over $\vee$) and flatten;
   e.g, $(A \vee (B \wedge C)) \equiv (A \vee B) \wedge (A \vee C)$
   e.g, $((A \vee B) \vee C) \equiv (A \vee B \vee C)$

Q1. Convert Following KB into CNF.

i) $B \leftrightarrow (A \vee C)$

Solution:

$B \leftrightarrow (A \vee C)$

$\equiv (B \rightarrow (A \vee C) \wedge (A \vee C) \rightarrow B))$

$\equiv (\neg B \vee (A \vee C) \wedge (\neg(A \vee C) \vee B))$

$\equiv (\neg B \vee A \vee C) \wedge [(\neg A \vee B) \wedge (\neg C \vee B)]$

$\equiv (\neg B \vee A \vee C) \wedge (\neg A \vee B) \wedge (\neg C \vee B)$

ii) $(A \rightarrow B) \vee (C \rightarrow B)$

$\equiv (\neg A \vee B) \vee (\neg C \vee B)$

iii) $(A \vee B) \rightarrow C$

$\equiv \neg(A \vee B) \vee C$

$\equiv (\neg A \wedge \neg B) \vee C$

$\equiv (\neg A \vee C) \wedge (\neg B \vee C)$

Resolution Algorithm:

Process (Algorithm)

i) Convert the given KB into CNF
ii) Add negation of the sentence to be entailed.
iii) Repeat the resolution rule.
iv) If there comes empty clause then
   → sentence is entailed to KB
   else
   → sentence is not entailed.

Q1. Consider the knowledge base $KB = (B \Leftrightarrow (A \vee C) \wedge \neg B)$.
Prove that $\neg A$ can be inferred from above KB by using resolution.

Solution:

Step1: Converting KB into CNF

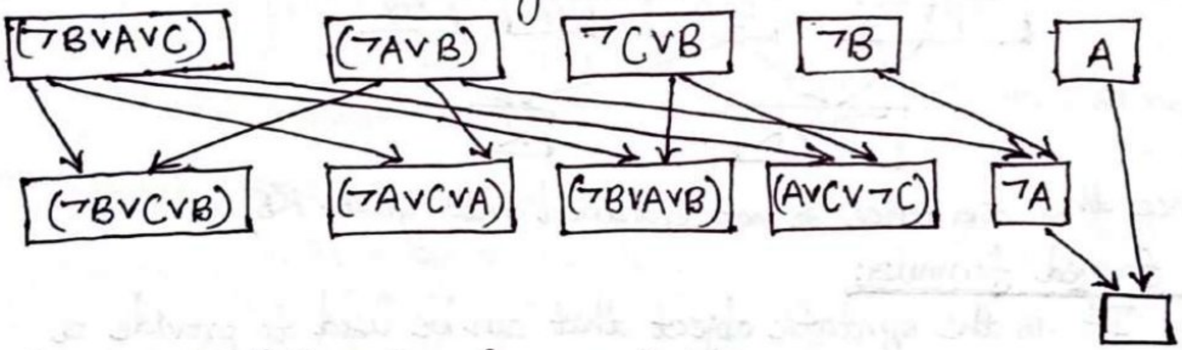
$$B \Leftrightarrow (A \vee C) \wedge \neg B$$
$$\equiv B \Rightarrow (A \vee C) \wedge (\neg(A \vee C) \Rightarrow B) \wedge \neg B$$
$$\equiv (\neg B \vee A \vee C) \wedge (\neg(A \vee C) \vee B) \wedge \neg B$$
$$\equiv (\neg B \vee A \vee C) \wedge ((\neg A \wedge \neg C) \vee B) \wedge \neg B$$
$$\equiv (\neg B \vee A \vee C) \wedge (\neg A \vee B) \wedge (\neg C \vee B) \wedge \neg B$$

(Note that $\Leftrightarrow$ is same as $\leftrightarrow$ and $\Rightarrow$ is same as $\rightarrow$ we can use any notation.)

Step2: Add negation of sentence to be inferred from KB into KB.
Now KB contains following sentences all in CNF.

$$\equiv (\neg B \vee A \vee C)$$
$$\equiv (\neg A \vee B)$$
$$\equiv (\neg C \vee B)$$
$$\equiv \neg B$$
$$\equiv A$$

Step3: Now use Resolution algorithm:



Q2. Given $KB = \{(G \vee H) \rightarrow (\neg J \wedge \neg K), G\}$. Show that the given KB entail $\neg J$ or not?

Solution:

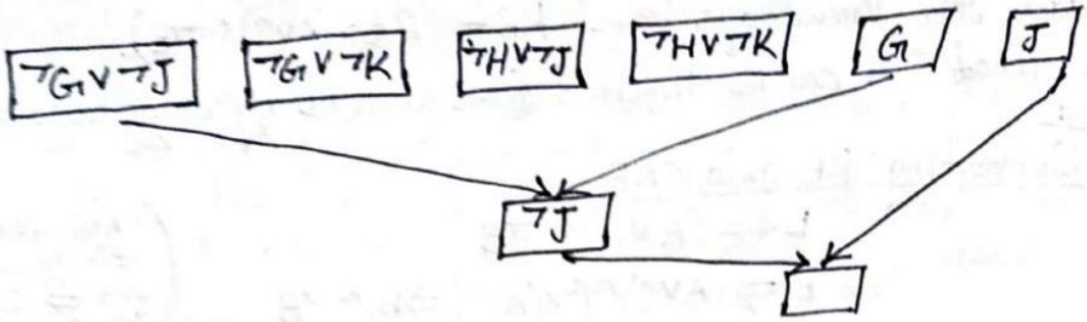
Step1:

$$(G \vee H) \rightarrow (\neg J \wedge \neg K)$$
$$\equiv (\neg(G \vee H) \vee (\neg J \wedge \neg K)$$
$$\equiv (\neg G \wedge \neg H) \vee (\neg J \wedge \neg K)$$
$$\equiv (\neg G \vee \neg J) \wedge (\neg G \vee \neg K) \wedge (\neg H \vee \neg J) \wedge (\neg H \vee \neg K)$$

Step2:

$$\neg(\neg J) \equiv J$$

Step3:



Hence the sentence is entailed into given KB.

Q3. $KB = \{P \to \neg Q, \neg Q \to R\}$. Show that the sentence $\neg P \to Q$ is entailed from given CNF.

Solution:

Step1:

$P \to \neg Q$ $\neg Q \to R$ $\neg P \to Q$

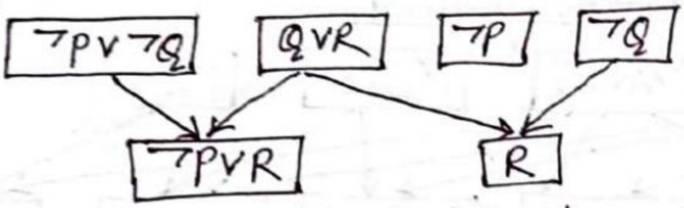
$\equiv \neg P \vee \neg Q$ $\equiv Q \vee R$ $\equiv P \vee Q$

Step2:

$P \vee Q$

$\equiv \neg P \wedge \neg Q$

Step3:



Hence the sentence is not entailed into given KB.

⊛ Well formed formula:

It is the syntatic object that can be used to provide a semantic meaning. A formal language can be considered to be identical to the inference iff they are well formed formula (wff). The wff are inductive defined as follows:

↳ each propositional variable is on its own a wff.

↳ if ϕ is a wff, then $\neg\phi$ is a wff.

↳ if ϕ & ψ are wff, & is binary connective, then (ϕ, ψ) is wff.

The wff for predicate logic is defined as:

↳ $\neg\phi$ is a wff when ϕ is a wff.

↳ $(\phi \wedge \psi)$ & $(\phi \vee \psi)$ are wff when ϕ & ψ are wff.

Backward Chaining and Forward Chaining:

The completeness of resolution makes a very good inference model. But in many practical situation full power of resolution is not required. Real world KB often contains only horn clauses. A horn clause is disjunction of literal with at most one positive literal.

e.g.

$\neg A \vee B \vee \neg C$ (horn clause)

$A \vee B \vee C$ (not horn clause).

Properties of horn clause:

→ Can be written as implication.

→ Inference through forward and backward chaining.

A) Forward Chaining:

Forward chaining is the logical process of inferring unknown truths from known data and moving forward using determined conditions and rules to find a solution. It is also known as data driven reasoning.

Steps:

→ Given knowledge base of true facts.

→ Apply all rules that match facts in knowledge base.

→ Add conclusions to the knowledge base.

→ Repeat until goal is reached, OR repeat until no new facts added.

Example 1: Suppose we have three rules: (not imp)

R1: If A and B then D

R2: If B then C

R3: If C and D then E

⇒ If facts A and B are present, we infer D from R1 and infer C from R2. With D and C inferred, we now infer E from R3.

Q1. $P \rightarrow Q$

$L \wedge H \rightarrow P$

$B \wedge L \rightarrow H$

$A \wedge P \rightarrow L$

$A \wedge B \rightarrow L$

A

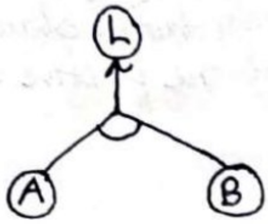
B

Show that Q can be inferred from given KB.

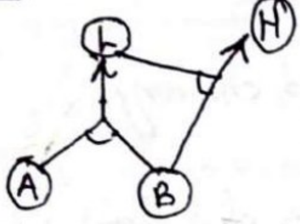
Solution:

Step1:

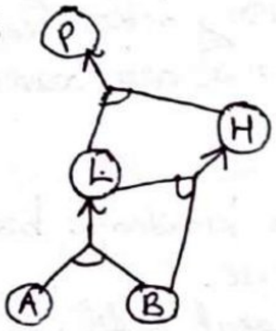
Step2:



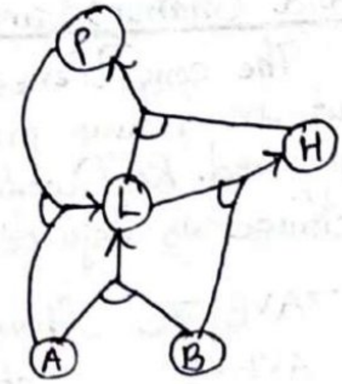
Step3:



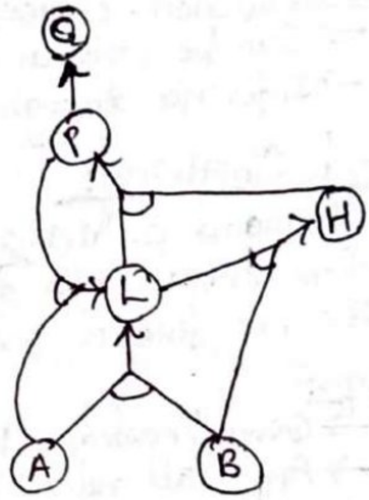
Step4:



Step5:



Step6:



B> Backward Chaining:-

Idea:

→ Work backward from Query Q to prove backward chaining.
→ Check if q is known already or prove by backward chaining.
→ Avoid loops: Check if new sub-goal is already on the goal state.

Q. $P \rightarrow Q$
$L \wedge H \rightarrow P$
$B \wedge L \rightarrow H$
$A \wedge P \rightarrow L$
$A \wedge B \rightarrow L$
A
B

Show that Q can be inferred from given KB by using backward chaining.

Solution:

We know that $P \rightarrow Q$, now we will try to prove P is true.
Again, $L \wedge H \rightarrow P$, try to prove L and H are true.

$B \wedge L \rightarrow H$ } try to prove. B, L, A and P are true.
$A \wedge P \rightarrow L$

Now, A & B are already known since. $A \wedge B \rightarrow L$, so L is known.
Since $B \wedge L \rightarrow H$, now H is also known.
Again, $L \wedge H \rightarrow P$, now P is also known/true.
Hence, Q can be inferred from given KB.

b) Predicate Logic:- FOPL (First-Order Predicate Logic):

FOPL is the symbolized reasoning in which each sentence or statement is broken down into a subset of predicate, where predicate is considered as function & relationship of subject. It is also a bi-valued function i.e, $P: X \rightarrow \{True \text{ or } False\}$ is called predicate on value x. A sentence in FOPL is written in the form of Px or $P(x)$, where P is predicate and x is subject.

e.g; man (Ram)
brother (Ram, Hari)
married (Puran, Sita)
brother of (Ram) = Hari

FOPL Syntax:

Sentence → Atomic sentence / Quantifier / ... sentence / ¬sentence
Atomic sentence → Predicate (term) / Term → Term
Term → function (term) / Term → Term
Connective → $\wedge / \vee / \neg / \rightarrow / \leftrightarrow$
Quantifier → $\forall, \exists$
Constant → A, B, C, X_1, X_2, Ram, Hari, ...
Variable → x_1, x_2, counter position, ...
Predicate → brother, has colour, ...
function → father of, sqrt, cosine, ...

FOPL Semantic:

An interpretation is required to give semantics FOPL. An interpretation is a non-empty set of objects. An interpretation provides:
→ Constant symbol to an object in the domain
→ Function symbols.
→ Predicate symbols.
then we can define universal & existential quantifiers.

Object/Subject: could be specified or unspecified.

Scanned with CamScanner

Normally predicates start with capital letters.

eg:

Loves (Ram, Sita)

Loves (x, y).

Sunny (Friday)

hot (x) ⟷ Hot (day)

Quantifiers (Quantification):

Quantifiers allow us to express properties of collection of objects instead of enumerating objects by name. There are two types of quantifiers. They are:

1) Universal quantifier (for all i.e, $\forall$):

$\forall$ <variable> <sentence>

e.g. Everyone at Texas is smart.

$\Rightarrow \forall x \; AT(x, Texax) \rightarrow Smart(x)$

It is logically equivalent to the conjunction of instantiations of P.

i.e, $AT(Arjun, Texas) \rightarrow Smart(Arjun) \wedge AT(Karishma, Texas)$.

It is also equivalent to a set of implications over all objects.

Note: '$\rightarrow$' is the main connective of $\forall$.

Common mistake at $\forall$

$\forall x \; AT(x, Texas) \wedge Smart(x)$

mistake

2) Existential Quantification (There exists i.e, $\exists$):

$\exists x$ <variable> <sentence>

e.g. Someone at Texas is smart.

$\Rightarrow \exists x \; AT(Texas) \wedge Smart(x)$.

It is logically equivalent to the disjunction of instantiations of P.

i.e, $AT(Arjun, Texas) \wedge Smart(Arjun) \vee AT(Karishma, Texas) \wedge Smart(Karishma)$.

Common mistake to $\exists$

$\exists x \; AT(x, Texas) \rightarrow Smart(x)$

mistake.

Representing Knowledge in FOPL:

The objects from real world are represented by the constants (A, B, C, Ram, ...). For instance, the symbol "Tom" may represent a certain individual Tom.

→ Properties of object may be represented by predicates.
e.g, danger(Tom)

→ Predicate also represents relation between objects.
e.g, friends (Hari, Ram)

→ Predicate also represent in the form of Boolean constant.
e.g, sister (Arjun, Sristhi) → F

Properties of quantifiers:-

i) $\forall x \; \forall y \equiv \forall y \; \forall x$

ii) $\exists x \; \exists y \equiv \exists y \; \exists x$

iii) $\exists x \; \forall y \neq \forall y \; \exists x$

e.g

$\exists x \; \forall y$ loves(x,y)
$\forall y \; \exists x$ loves(x,y)] Not logically equivalent.

Inference in FOPL:

FOPL inference can be done by converting the KB into the predicate logic and using propositional logic inference. The major task in inference of FOPL is to deal with quantifiers so instantiation is required. There are two types of instantiation. They are:

1) Universal Instantiation (UI):

Substitute ground item (term without variable) for the variable.

e.g. $\forall x$ King$(x) \wedge$ Greedy$(x) \rightarrow$ Evil(x)
King (John), Greedy (John), Brother (Richard, John), Father (Arjun, John).

Now its UI is:

King (John) ∧ Greedy (John) → Evil (John)
King (Richard) ∧ Greedy (Richard) → Evil (Richard)
King (Arjun) ∧ Greedy (Arjun) → Evil (Arjun)
King (father of (John))
King (brother of (John)).

2) Existential Instantiation (EI):

For any sentence & variable V in that, introduce a constant that is not in the KB (called skolen constant) and substitute that constant on V.

e.g., $\exists x$ Crown$(x) \wedge$ Onhead$(x, John)$

After EI:

Crown$(c1) \wedge$ Onhead$(c1, John)$

where c1 is skolem constant.

Resolution of FOPL:

→ Same as proposition logic (PL)
→ Allows extended syntax such as quantifiers.
→ Define causual form for FOPL i.e, CNF
→ Eliminate quantifiers from causual form.
→ Adopt resolution procedure to cope with variables. i.e, unification.

Rule for converting into CNF:

1) Eliminate bi-implication & implication similarly as propositional logic.

2). Move negation inwards by De-Morgan's law.

$\neg \forall x\ P \rightarrow \exists x\ \neg P$

& $\exists x\ P \rightarrow \forall x\ \neg P$

3) Eliminate double negation.

$\neg(\forall x\ \neg P) \rightarrow \exists x\ P$

$\neg(\exists x\ P) \rightarrow \forall x\ P$

4) Remove bound variables if necessary. OR operation

$\forall x\ P(x) \vee \exists Q(x) \rightarrow \forall x\ [P(x) \vee \exists y\ Q(y)]$

5) Use equivalence to move quantifier to the left.

e.g. $\forall x\ P(x) \wedge Q \rightarrow \forall x\ (P(x) \wedge Q)$

↳ where x does not belong to Q.

6) Skolemize:

$\exists x\ P(x) \rightarrow P(C1)$

$\forall x\ \exists y\ P(x,y) \rightarrow \forall x\ P(x, c1)$

7) The formula now has only universal quantifier & drop universal quantifier.

8) Use distribution law to get CNF.

Q. Convert following KB into CNF. (FOPL)

$\forall x[\forall y\, P(x,y) \rightarrow \neg\forall x\,(Q(x,y) \rightarrow R(x,y))]$

$\equiv \forall x\,[\neg(\forall y\, \neg P(x,y) \vee \neg\forall y\,(Q(x) \rightarrow \neg R(y))]$

$\equiv \forall x\,[\exists y\, P(x,y) \vee \neg\forall y\,(\neg Q(x) \vee \neg R(y))]$

$\equiv \forall x\,[\exists y\, P(x,y) \vee \exists y\,(Q(x) \wedge R(y))]$

$\equiv \forall x\,[\exists y\, P(x,y) \vee \exists z\,(Q(x) \wedge R(z))]$

$\equiv \forall x\,[\exists y\, \exists z\,(P(x,y) \vee (Q(x) \wedge R(z))]$

$\equiv \forall x\,[P(x,g(x)) \vee (Q(x) \wedge R(f(x))]$

$\equiv \quad P(x,g(x)) \vee (Q(x) \wedge R(f(x)))$

$\equiv (P(x,g(x)) \vee Q(x)) \wedge (P(x,g(x)) \vee R(f(x)))$

Q. Convert the following sentences in FOPL.

1\> Both TIC & TCHIT are in Network.
$\Rightarrow$ In (TIC, Texas) $\wedge$ In (TCHIT, Texas)

2\> Both CSIT & BCA are TU affiliate.
$\Rightarrow$ Affiliation (CSIT, TU) $\wedge$ Affiliation (BCA, TU).

3\> Manoj wrote "the girl I love."
$\Rightarrow$ Writes (Manoj) $\rightarrow$ loves (girl).

4\> Either Manoj or Harish wrote "the girl I like".
$\Rightarrow$ Writes (Manoj) $\vee$ Writes (Harish) $\rightarrow$ likes (girl).

5\> Puskal owns a copy of AI.
$\Rightarrow$ Owns (Puskal, Copy) $\rightarrow$ AI (copy).

6\> Every song that Baibhav sings on Sainik was written by Baibhav.
$\Rightarrow \forall x$ Sing (Baibhav, x) $\wedge$ Sainik (x) $\rightarrow$ Writes (Baibhav).

7\> Ram is a man.
$\Rightarrow$ Man (Ram).

8\> America brough Alaska from Russia
$\Rightarrow$ bought (who, what, from)
bought (America, Alaska, Russia)

9\> Jym collects everything
$\Rightarrow \forall x$ Collects (Jym, x).

10\> Somebody collects something. $\Rightarrow \exists x,y$ collects (x,y).

11) No stinky shoes are allowed.
$\Rightarrow \exists x\ shoes(x) \wedge stinky(x) \wedge \neg allowed(x)$.

12) Good people always have good friends.
$\Rightarrow \forall x\ Person(x) \wedge Good(x) \rightarrow \exists y\ friend(x,y)$.

13) You can fool some of the people all the time, and all of people some of the time but you cannot fool all the people all the time.

$\Rightarrow [\exists x \forall t (person(x) \wedge time(t) \rightarrow canfool(x,t))] \wedge [\forall x \exists t (person(x) \wedge time(t) \rightarrow canfool(x,t)] \wedge [\forall x \forall t (person(x) \wedge time(t) \rightarrow \neg canfool(x,t))]$

Unification and Lifting:

Atomic sentences → The most basic sentences in FOPL formed from predicate symbol followed by a parenthesis with sequence of terms.

e.g. Chinky is a cat
$\Rightarrow$ Cat(Chinky).

Complex sentence → Combination of atomic sentences using connectives.

Before unification and lifting we must have the idea of general modus ponens (GMP) in FOPL. i.e, For atomic sentence P_i, P_i' & q, where there is substitution θ such that
SUBSET (θ, P_i) = SUBSET (θ, P_i') for all i.

$$\frac{P_1', P_2', \ldots, P_n', (P_1 \wedge P_2 \wedge \ldots \wedge P_n \rightarrow q)}{\text{where } P_i'\theta = P_i\theta \text{ for all } i.}$$

i.e, SUBSET (θ, q)

$\therefore$ n atomic sentences + one implication applying SUBSET (θ, q) yeilds the conclusion.

e.g. $P_1' =$ King (John) $\quad P_2' =$ Greedy (y)

$P_1 =$ King (x) $\quad P_2 =$ Greedy (x)

$\theta = \{x/John, y/John\}$ $\quad q =$ evil(x)

SUBSET (θ, q) is Evil(John).

⇒ The process of converting knowledge base (KB) into logically equivalent FOBL is called lifting.

Unification: Process we use to find substitution that makes different logical expression looks identical.

i.e, Unify $(\alpha, \beta) = \theta$ if $\alpha\theta = \theta\beta$.

e.g, Who does Ramesh know?

Unify (knows(Ramesh, x)) → Knows (Ramesh, Sita)

$\theta = \{x/Sita\}$

Inference using Resolution:

i) Convert all sentences in CNF.
ii) Negate conclusion & convert into CNF.
iii) Add negated conclusion to premise clause.
iv) Repeat until contradiction or no progress is made.

Question:
i) If something is intelligent, it has common sense.
ii) Deep blues do not have common sense.
Prove that Deep blue is not intelligent.

Solution:
1) $\forall x$ Intelligent(x) → Hascommonsense(x)
2) ¬Has common sense(Deep blue)
3) ¬Intelligent (Deep blue).

Step1: Converting into CNF:

$\equiv \forall x$ [¬Intelligent(x) ∨ Hascommonsense(x)]

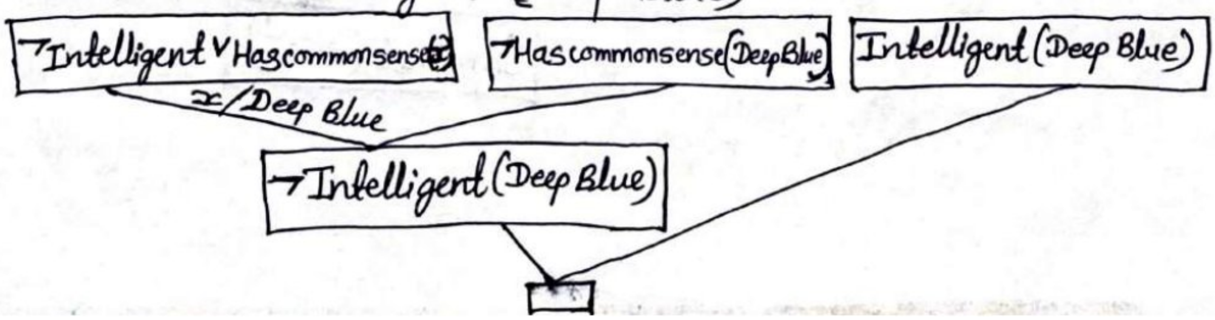
$\equiv$ ¬Intelligent(x) ∨ Hascommonsense(x)

Step2: Adding Negation to the conclusion.

¬(¬Intelligent (Deep Blue))

$\equiv$ Intelligent (Deep Blue)

Step3:



Scanned with CamScanner

Forward and Backward Chaining in FOPL:

Q. related to forward & backward chaining

Q. The law says that it is crime for an American to sell weapons to hostile nation. The country Nono, an enemy of America, has some missiles and all of its missiles were sold by Colonel West who is American. Now prove that Colonel West is criminal.

any small village of a country

Solution:

Representing in FOPL:

↳ It is a crime for American to sell weapons to hostile nation.

$American(x) \wedge Weapon(y) \wedge sell(x,y,z) \wedge Hostile(z) \rightarrow Criminal(x)$.

↳ The country Nono, an enemy of America, has some missiles.

$\exists x \; Owns(Nono, x) \wedge Missile(x)$.

FI: Replace x by M1.

$Owns(Nono, M1) \wedge Missile(M1)$.

↳ All missiles are sold by Colonol West.

$Missile(x) \wedge Owns(Nono, x) \rightarrow sells(West, x, Nono)$.

Now, we must know that missiles are weapons.

$Missile(x) \rightarrow Weapon(x)$.

Again, enemy of America counts a hostile.

$enemy(x, America) \rightarrow Hostile(x)$.

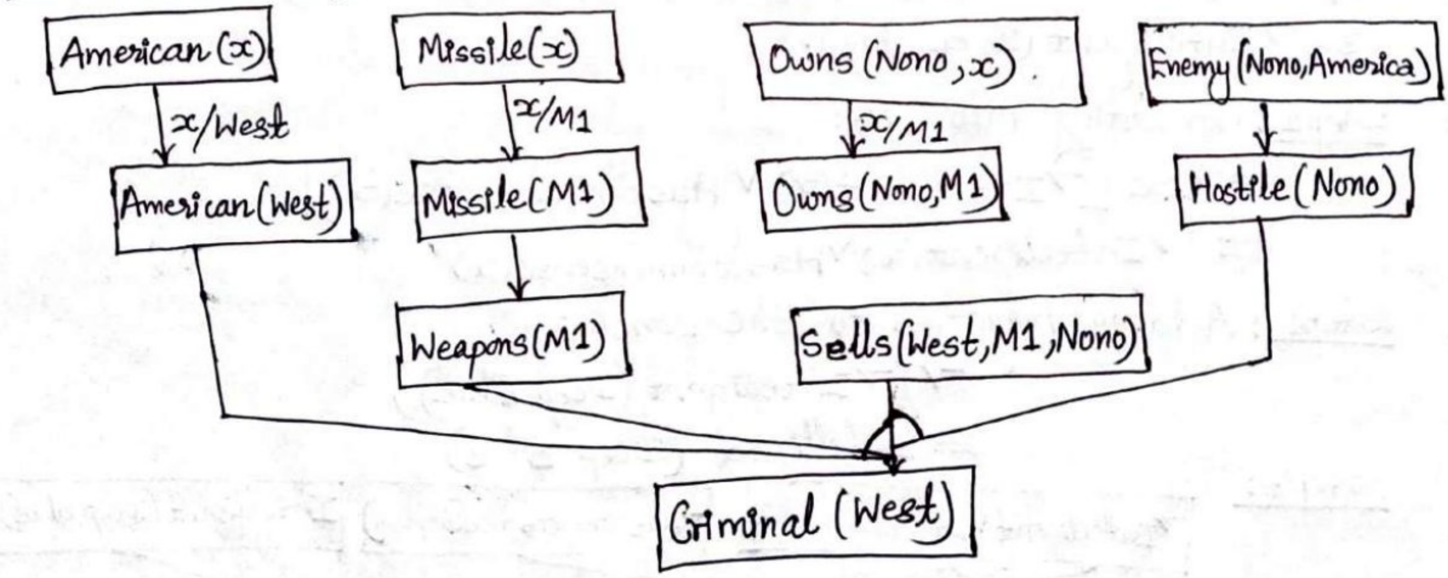
Also, Colonel West is an American.

$American(West)$

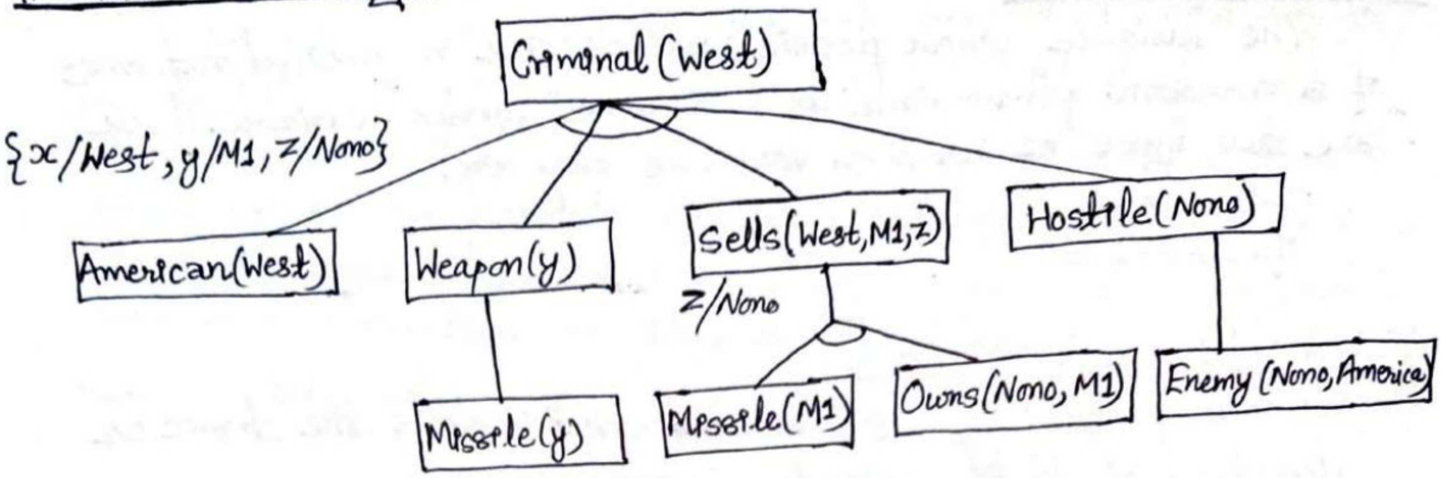
Again, the country Nono is a enemy of America.

$enemy(Nono, America)$.

Forward Chaining:



Backward Chaining:



Handling Uncertain Knowledge:-

→ Agents need to handle uncertainity whether due to partial observability, non-determinism or combination of both. An agent may never know for certain what state it in or where it will end up after sequence of actions.

→ Although agent already have the knowledge about environment but due to some external natural factor agent will not be able to achieve actual mathematical interpretation goal, so it must take decision by analysing external factors and that decision is called rational decision.
e.g. Automated taxi

→ Let us take another example of uncertain knowledge in simple rule.

Toothache → Cavity

The rule itself is wrong in real scenario. Not all patients with ~~toothache~~ have cavities, some of them have gum disease or any other problems. Now the rule becomes:

Toothache → Cavity V Gum problem V ...

→ Unfortunately in order to make rule true we have to add almost unlimited list of possible problems.

→ So to move from one state to another state any fixed rule is applied, this case is considered as condition of uncertainity.

→ The only decision of uncertainity will be generated from probability theory.

Random Variables:

The variable whose possible values are numerical outcomes of a random phenomena is known as random variable. There are two types of random variables they are:

i) Discrete.
ii) Continious.

Probability and it's types:

→ Simply probability is a number that reflects the chance or likelihood of that particular event occurance.

→ ~~It refers to~~ The degree of belief in proposition in the absence of any other information is called unconditional or prior probability.

Example:

$$P(A) = \frac{\text{total number of way that event A occur}}{\text{total number of possible outcomes.}}$$

$$P(\text{cavity}) = \frac{\text{cavity}}{\text{Toothache}}$$

→ The probability of the event occuring with some relationship to one or more other events is called conditional or posterior probability.

Example:

representing age interval like young, adult, teenage

$$P(\text{cavity} / \text{Teen}) = \frac{\text{Cavity} \wedge \text{Teen}}{\text{Toothache}}$$

→ Decision theory = Probability theory + Utility theory.

Inference using Full Joint Distribution:

The process of generating knowledge by using joint probability is called an inference using full joint probability distribution. It is the statistical measure that calculate the likelihood of two events occuring together & at the same point of time.

Example:- Let we have the following set of domains:

Age = {Child, Teen, Young, Adult} & Cavity {T, F}

Now, the probability of P(Age, Cavity) is represented in 4X2 matrix as:

Age	Child	Teen	Young	Adult
Cavity=T	0.144	0.02	0.016	0.02
Cavity=F	0.576	0.08	0.064	0.08

⊛ Baye's Rule and it's uses:

Baye's theorem is a way to apply conditional probability for prediction. Conditional probability is the probability of an event happening, given that it has some relationship to one or more other events. For example, the probability of getting a parking space is connected to the time of day we park, where we park and what conventions are going on at any time.

Mathematically, Bayes theorem is defined as:

$$P(A/B) = \frac{(B/A)\ P(A)}{P(B)}$$

Uses:

→ It is widely used to provide probabilistic prediction in AI.
→ In finance, Baye's theorem can be used to rate the risk of lending money to potential borrowers.
→ It allows us to update predicted probabilities of an event by incorporating new information.

Example: A doctor knows that the disease meningitis causes the patient to have a stiff neck 50% of the time. The doctor also knows that the probability that a patient has meningitis is 1/50,000 and the probability that any patient has a stiff neck is 1/20. Find the probability that a patient with a stiff neck has meningitis.

Solution:

Let m be the patient having Meningistis and s be the patient having stiff neck.

Given,

$$P(s/m) = 50\% = 0.5$$
$$P(m) = 1/50{,}000$$
$$P(s) = 1/20$$

Now,

$$P(m/s) = \frac{P(s/m).P(m)}{P(s)} = \frac{0.5 \times \frac{1}{50{,}000}}{1/20} = 0.0002$$

⊛. Bayesian Networks:

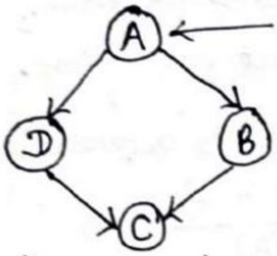
Bayesian network is a type of probabilistic graphical model that uses Bayesian inference for probabilistic inference. It is a directed cyclic graph which consist of:

i) a set of random variables represented as nodes of network.

ii) Set of directed links connecting pair of nodes. If there is an arrow from node x to y then x is called parent of y.

iii) Each x_i has a conditional probability distribution $(P(x_i)/Parents(x_i))$ that quantifies the effect of the parents on that node.

Example:

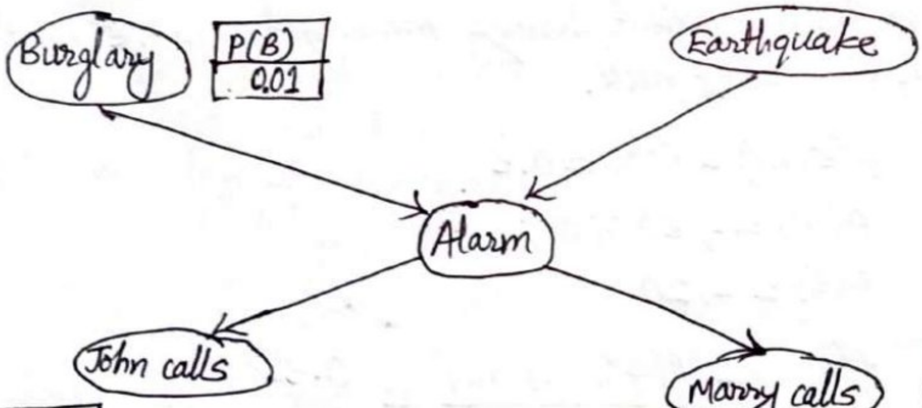


A is the parent of D & B

The conditional probability distribution is calculated to find the effect of A in B.

→ Bays network is used to find the relationship of dependent variable on uncertain condition.

Example: Let we have a new burglar alarm installed at home. It is reliable at detecting a burglary but also responds on the occasion of minor earthquakes. We also have two neighbours Johny and Mary who have promised us to call when we are at our work if they hear the alarm. John nearly calls when he hears the alarm but sometime confuses the telephone ringing with the alarm & call them too. Mary on the other hand likes rather loud music & often misses the alarm. Altogether given the evidence of who has or has not called the probability on Bayesian network is:



P(B)
0.01

P(E)
0.002

B	E	P(A)
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

A	P(J)
T	0.90
F	0.05

A	P(M)
T	0.70
F	0.01

Here, the probability of alarm depends on the burglary & earthquake but the calls of John & Mary only depends on the alarm.

⊛. Inference with Bayesian (Belief) Network :-

First we simply evaluate the joint probability of a particular assignment of values for each variable (or a subset) in the network. For this, we already have a factorized form of the joint distribution, so we simply evaluate the product using the provided conditional probabilities.

Example: What is the probability that the alarm has sounded, but neither burglary nor an earthquake has occured, and both John and Marry call? continue Q. of previous e.g

Solution:-

from previous example

Given,

$P(B) = 0.01$

$P(\neg B) = 1 - 0.01 = 0.99$

$P(E) = 0.002$

$P(\neg E) = 1 - 0.002 = 0.998$

$P(A/\neg B, \neg E) = 0.001$

$P(J/A) = 0.90$

$P(M/A) = 0.70$

Therefore the probability of above condition is given by;

$$P(\neg B) \times P(\neg E) \times P(A/\neg B, \neg E) \times P(J/A) \times P(M/A)$$

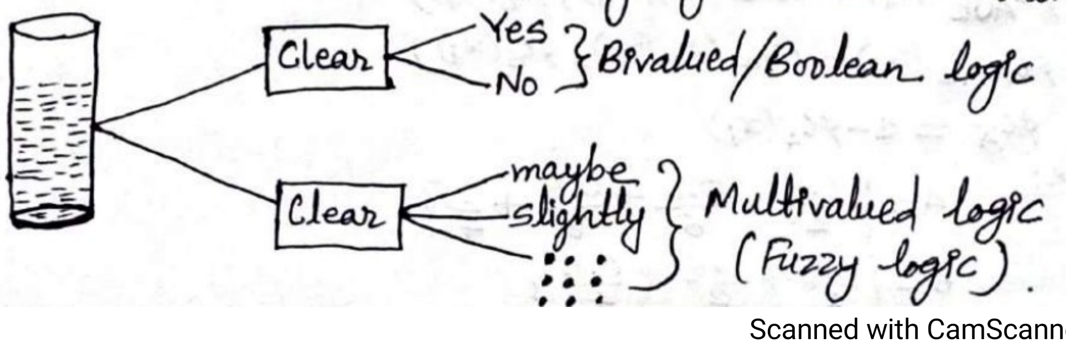
$$= 0.99 \times 0.998 \times 0.01 \times 0.9 \times 0.7$$

$$= 0.00062$$

⊛ Fuzzy Logic :-

The word fuzzy refers to things which are not clear. Any event, process, or function that is changing continiously cannot always be defined as either true or false, which means that we need to define such activities in a Fuzzy manner.

In fuzzy systems, the values are indicated by a number in the range from 0 to 1. Here 1.0 represents absolute truth and 0.0 represents absolute fa'eness. The number which indicates the value in fuzzy systems is called truth value.



⊛. Fuzzy sets:

i.e, classical set

A set is an unordered collection of different elements. If the order of the elements is changed or any element of a set is repeated, it does not make any changes in the set.

Fuzzy sets can be considered as an extension and gross over simplification of classical sets. Classical set contains elements that satisfy precise properties of membership while fuzzy set contains elements that satisfy imprecise properties of membership. A set X in which each element y has a grade of membership $\mu_x(y)$ in the range 0 to 1, i.e, set membership may be partial.

For example:- If cold is a fuzzy set, exact tempreature values might be mapped to the fuzzy set as follows:

15 degrees → 0.2 (slightly cold).

10 degrees → 0.5 (quite cold)

0 degrees → 1 (totally cold).

Membership:

Membership of each element in fuzzy set is mapped as $\mu(x)=[0,1]$.

If A is a fuzzy set over domain $X=\{a,b,c\}$ then A is defined as;

$$A=\left\{\frac{\mu_A(a)}{a}+\frac{\mu_A(b)}{b}+\frac{\mu_A(c)}{c}\right\}.$$

e.g $\text{Even}=\left\{\frac{0.9}{1}+\frac{1}{2}+\frac{0.98}{3}\right\}$ or

$=\left\{\frac{0.9}{1},\frac{1}{2},\frac{0.98}{3}\right\}$ or

$=\{(1,0.9),(2,1),(3,0.98)\}$

Fuzzy Set Operations:-

i) Union

ii) Intersection

iii) Complement

Suppose A & B over X then;

$\mu_{A\cup B}(x_i)=\text{Max}(\mu_A(x_i),\mu_B(x_i))$

$\mu_{A\cap B}(x_i)=\text{Min}(\mu_A(x_i),\mu_B(x_i))$

$\mu_{\bar{A}}=1-\mu_A(x_i)$.

Example: $A=\left\{\frac{0.5}{1}+\frac{0.2}{6}+\frac{0.9}{8}+\frac{1}{4}\right\}$

$B=\left\{\frac{0.9}{1}+\frac{0.6}{2}+\frac{0.5}{4}\right\}$

$$A \cup B = \left\{ \frac{MAX(0.5, 0.9)}{1} + \frac{MAX(0, 0.6)}{2} + \frac{MAX(1, 0.5)}{4} + \frac{MAX(0.2, 0)}{6} + \frac{MAX(0.9, 0)}{8} \right\}$$
$$= \left\{ \frac{0.9}{1} + \frac{0.6}{2} + \frac{1}{4} + \frac{0.2}{6} + \frac{0.9}{8} \right\}$$

$$A \cap B = \left\{ \frac{MIN(0.5, 0.9)}{1} + \frac{MIN(0, 0.6)}{2} + \frac{MIN(1, 0.5)}{4} + \frac{MIN(0.2, 0)}{6} + \frac{MIN(0.9, 0)}{8} \right\}$$
$$= \left\{ \frac{0.5}{1} + \frac{0}{2} + \frac{0.5}{4} + \frac{0}{6} + \frac{0}{8} \right\}$$

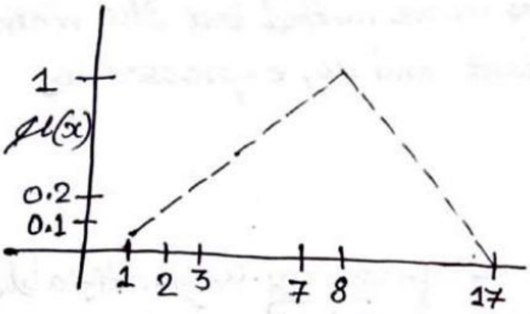
$$\mu_{\overline{A}} = \left\{ \frac{1-0.5}{1} + \frac{1-0.2}{6} + \frac{1-0.9}{8} + \frac{1-1}{4} \right\}$$
$$= \left\{ \frac{0.5}{1} + \frac{0.8}{6} + \frac{0.1}{8} + \frac{0}{4} \right\}$$

each value को लागि for e.g. $\frac{1-0.9}{1}$ for first + ...

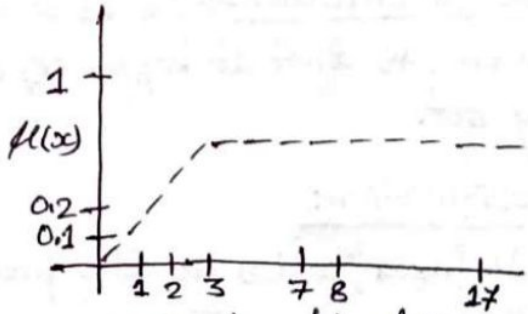
Q. Verify $\overline{A \cup B} = \overline{A} \cap \overline{B}$ for above example.

Here,
- → माथिको जस्तै $B = \mu_B$ calculate गर्ने.
- → त्यसपछि माथिको $A \cup B$ बाट $(1 - A \cup B)$ गरि $\overline{A \cup B}$ निकाल्ने (यो LHS भयो Q. को).
- → Finally RHS calculate गर्ने $\overline{A} \cap \overline{B}$, $\overline{A}$ र $\overline{B}$ first calculate गरि (यो RHS).
- → LHS र RHS equal आउछ।

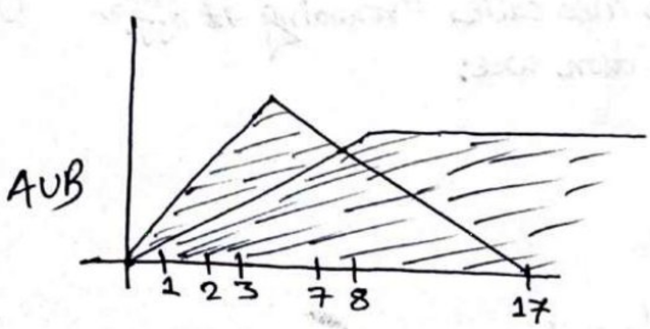
⊛ Membership graph of Fuzzy set.

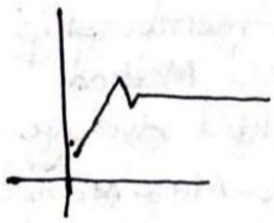


Membership of A

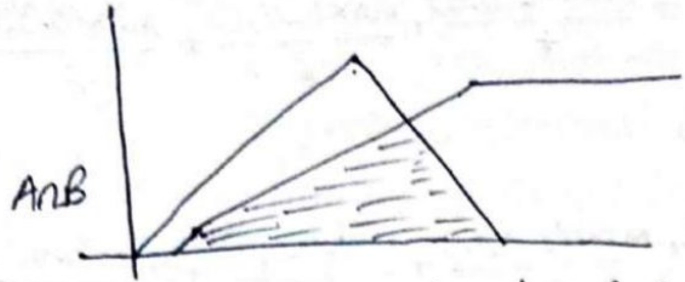


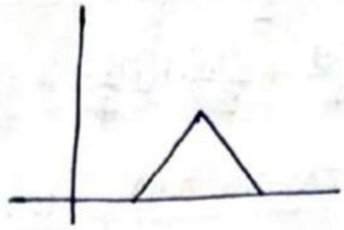
Membership of B





Membership of AUB





Membership of A∩B

⊛. Fuzzification:

Fuzzification is the process ~~in which~~ where membership functions are applied, and the degree of membership is determined. There are two important methods of fuzzification.

i) Support fuzzification → In this method, the fuzzified set can be expressed with the help of following relation:

$$A^{\sim} = \mu 1\ Q(x1) + \mu 2\ Q(x2) + \ldots + \mu n\ Q(xn).$$

where, $Q(x_i)$ → kernel of fuzzification.

This method is implemented by keeping μ_i constant and x_i being transformed to a fuzzy set $Q(x_i)$.

ii) Grade fuzzification → It is similar to above method but the main difference is that it keeps x_i constant and μ_i expressed as fuzzy set.

⊛. Defuzzification:

Defuzzification is the process of producing a quantifiable result in fuzzy logic. It interprets the membership degrees in the fuzzy sets into a specific action or real-value. Mathematically the process of Defuzzification is also called "rounding it off."

The different methods of Defuzzification are:

i) Max-membership method

ii) Centroid Method.

iii) Weighted Average Method.

iv) Mean-Max Membership.

⊛. Applications of fuzzy logic systems:

→ Used in aerospace field for altitude control of spacecraft and satellite.

→ Used in automotive system for speed control, traffic control.

→ Used in Natural language processing.

→ Used in modern control systems such as expert systems.

→ Used in Neural Networks.

⊛ Advantages of Fuzzy logic system:

→ This system can work with any type of inputs whether it is imprecise, distorted or noisy input information.

→ The construction of Fuzzy logic Systems is easy and understandable.

→ The algorithms can be described with little data, so little memory is required.

→ Fuzzy logic comes with mathematical concepts of set theory and the reasoning of that is quite simple.

⊛. Disadvantages of Fuzzy logic systems:

→ It leads to ambiguity and there is no systematic approach to solve a given problem.

→ Proof of its characteristics is difficult or impossible in most cases.

→ Accuracy is compromised, since it works on precise as well as imprecise data.