Unit-4: Knowledge Representation

- Ankit Pangeni

DATE

Definition and Importance of Knowledge.

Knowledge is what we know as facts. The facts are declarative sentences that are either true or false. In case of problem solving computer, Knowledge is the information about a particular domain that can be used to solve problems in that domain.
It is not just a data, it also consists of facts, ideas, believes, association rules, relationships, etc.

Importance.

- Important in problem solving. It requires full knowledge about the problem domain, and this knowledge must be represented in the computer understandable form.

- Knowledge plays an important role in demonstrating intelligence behaviour in agents or systems

Note: Process of converting informal knowledge into computer understandable form is known as knowledge representation scheme. It specifies the form of knowledge. A knowledge base is the representation of all the knowledge that is stored by an agent

$$
\begin{tikzcd}
\text{problem} \arrow[r, "\text{Solve}"] \arrow[d, "\text{represent}"'] & \text{solution} \\
\text{representation} \arrow[r, "\text{compute}"] & \text{output} \arrow[u, "\text{interpret}"']
\end{tikzcd}
$$

informal
formal.

Issues in Knowledge Representation.

Some of the issue regarding knowledge representation can be formulated as follows:-

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

In Summary, selection of important attributes to form the bank of information about an object or word, and the relationship between the attributes of an object and what level should the knowledge be represented, i.e. Granularity are the issues in knowledge representation.

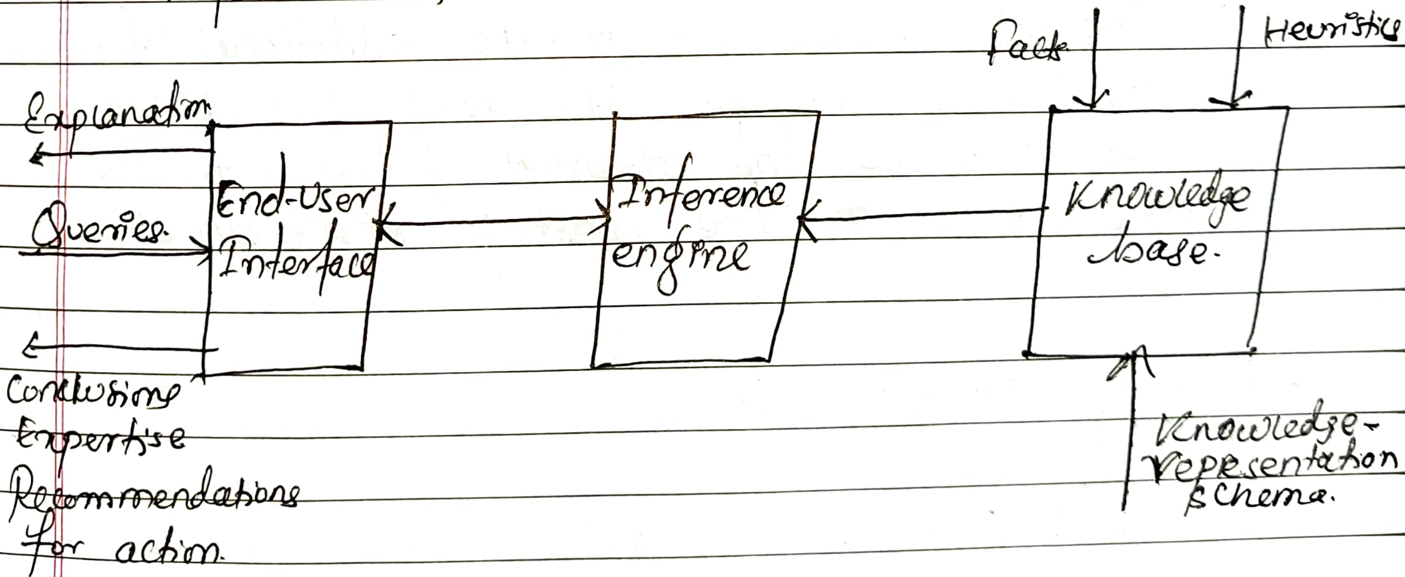
Knowledge representation Systems and their properties

Knowledge representation system is a formalized structure and set of operations that contains the descriptions, relationships, and procedures provided in an AI system. The process considers how to represent the knowledge such that it can be used to solve problems.

It basically Consists of two components.

1) Knowledge base
→ It is the representation of knowledge that is stored by an agent.

2) Inference method.
→ Method to find the conclusion based on facts, information, and evidence.



A good knowledge representation system should consist of following properties:

DATE

i) Representational adequacy

The scheme is rich enough to express the knowledge needed to solve the problem. i.e. The ability to represent all of the kinds of knowledge that are needed in a certain domain

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

iii) Inferential efficiency:-

The ability to incorporate additional information into the knowledge structure which can be used to focus the attention of the inference mechanisms in the most promising direction

iv) Acquisitional efficiency.

The ability to acquire new knowledge easily using automatic methods wherever possible rather than reliance on human intervention.

Types of knowledge Representation System

I. Semantic Nets.

It is a directed or undirected graph consisting of vertices, which represent concepts, and edges, which represent Semantic relations between concepts.
It is a method of knowledge representation that represents semantic relations between concepts in the form of a network. The resulting knowledge base is also known as knowledge map

It is a kind of directed graph where a node may represent a fact description such as: Physical object, Concept or event.
And an arc (or link) represents relationships between nodes. Some standard relationship types are:

'Is a' : represents class/inheritance relationships.
'Has a' : represents part-subpart i.e. property relationship

(Components of a semantic network:)

* Lexical part
  - nodes:- denoting Objects
  - links: denoting relationship between objects
  - labels:- denoting particular objects & relations

* Structural part : The links & nodes from directed graph ~~are~~ the lables are placed on the links and nodes

DATE

* Semantic part :- meanings are associated with the link & node labels.

* Procedural part :-
constructors → allow creation of new links & nodes
destructors → deletion of links and nodes.
writers → allow creation & alteration of labels
readers → can extract answers to questions

(Types of Semantic Network):

* Definitional networks : 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 its subtypes.

* Assertional networks : 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.

* Implicational networks : These use implication as the primary relation for connecting nodes. They may be used to represent pattern of beliefs, casuality or inferences

* learning networks : These are used to build or extend their representations by acquiring knowledge from examples

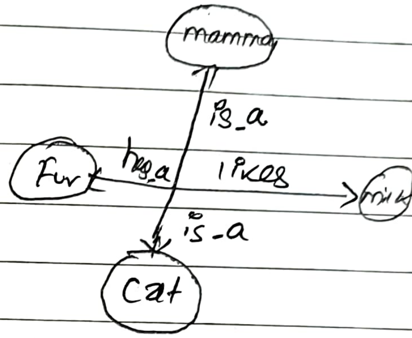
* Hybrid networks:

Combines two or more of the previous techniques, either in a single network or separate, but closely interacting networks.

Examples of Semantic networ.

Ex: 1
- Cat is a mammal
- Cat likes milk
- Cat has fur
- tommy is a cat

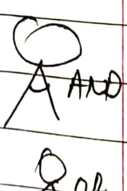
→



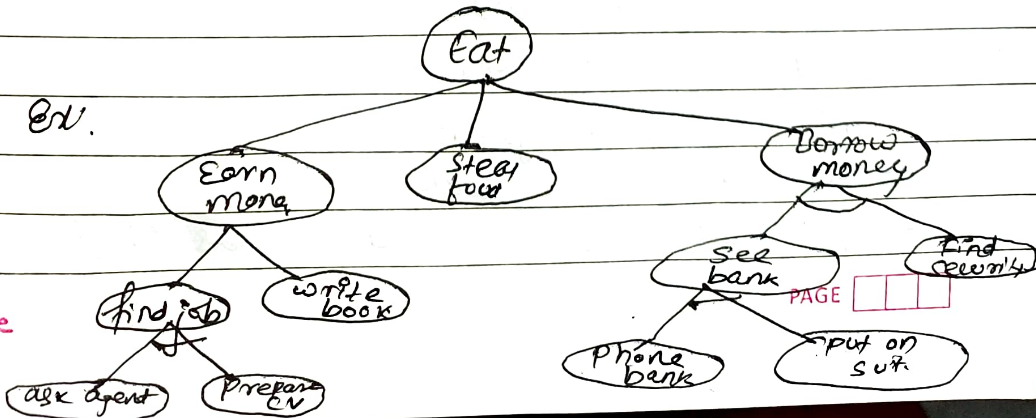
AND/OR Tree.

→ An AND/OR tree is a tree whose internal nodes are labelled either "AND" or "OR". A valuation of an AND/OR tree is an assignment of "TRUE" or "FALSE" to each of the leaves.

→ Given a tree T & a valuation over the leaves of T, the values of the internal nodes & of T are defined recursively in the obvious way:-
- An OR node is TRUE if at least one of its children is TRUE.
- An AND node is TRUE if all of its children are TRUE



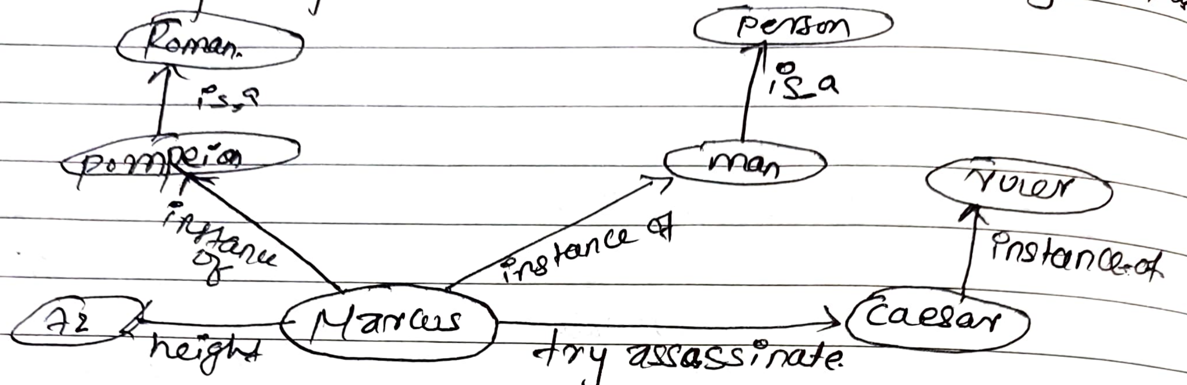
Ex.



DATE

Binary relations

An example of semantic network with binary relation



→ (Marcus, caesar) or constants(72) → represents ~~classes~~ of individuals objects

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

→ instance of → represents element of a class

→ tryassasinate → represents tried to assassinate

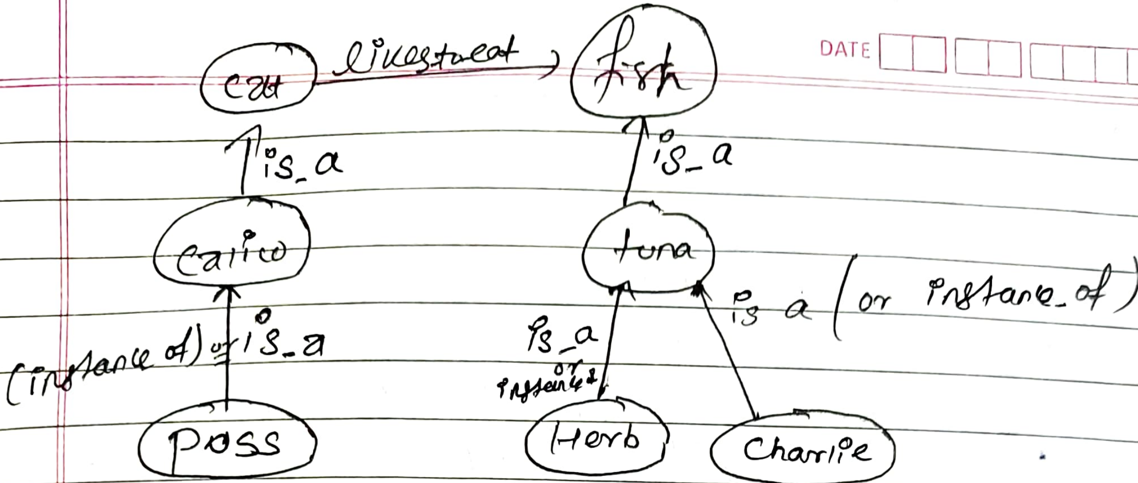
→ is a → represents sub class of a class.

ex: 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

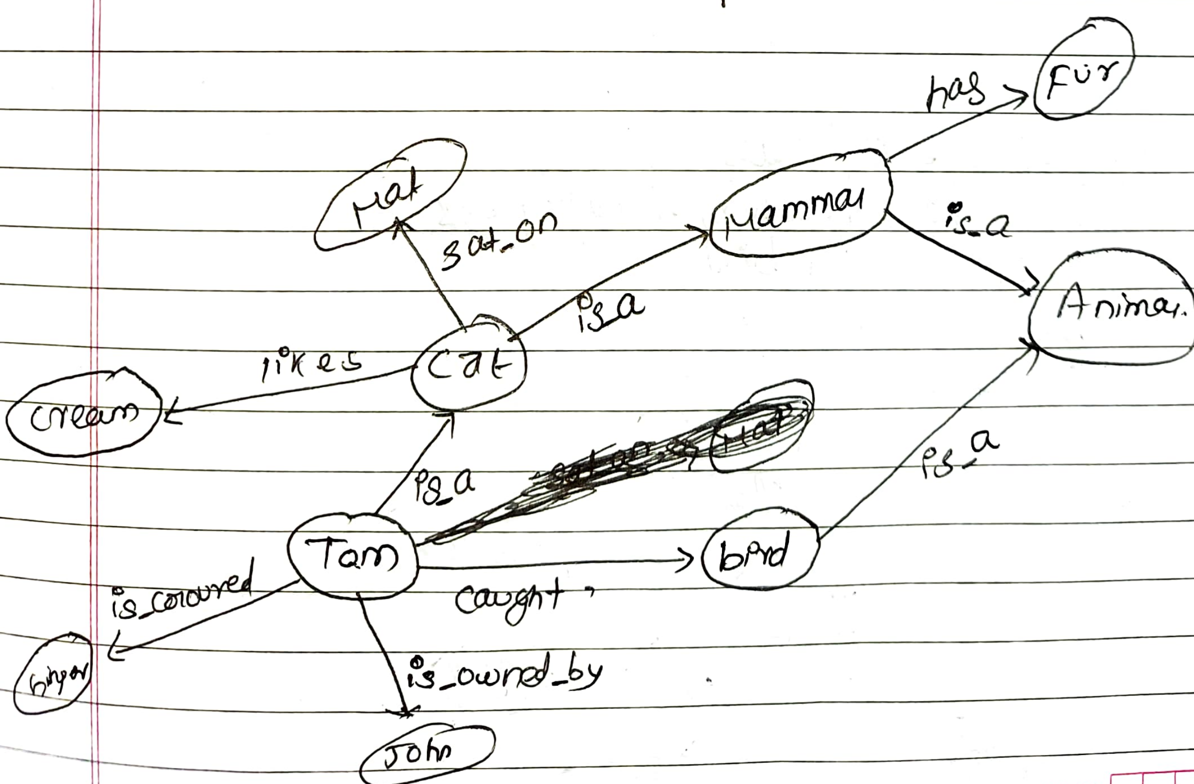
classmate

PAGE



Ex: Represent the following sentences into a semantic network

→ Tom is a cat	→ A cat is a mammal
→ Tom caught a bird	→ A bird is an animal
→ Tom is owned by John	→ All mammals are animals
→ Tom is ginger in color	→ Mammals have fur
→ Cats like cream	
→ The cat sat on the mat	

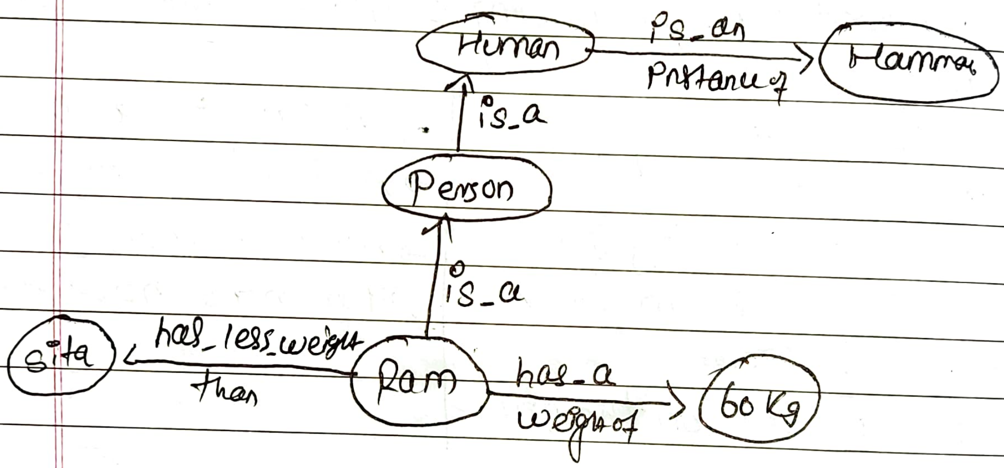


Note: no need to give circles.

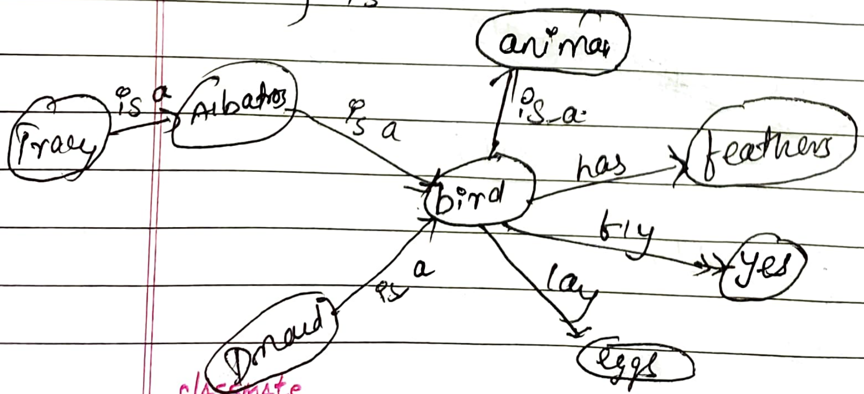
DATE

Ex: Construct a Semantic network

→ Ram is a person
→ Persons are humans
→ All humans have nose
→ Humans are instances of mammals
→ Ram has a weight of 60 kg
→ weight of Ram is less than weight of sita



Ex. → Birds are animals
→ Birds have feathers, fly & lay eggs
→ Albatros is a bird
→ Donald is a bird
→ Tracy is an albatros



2. Frame based knowledge Representations :-

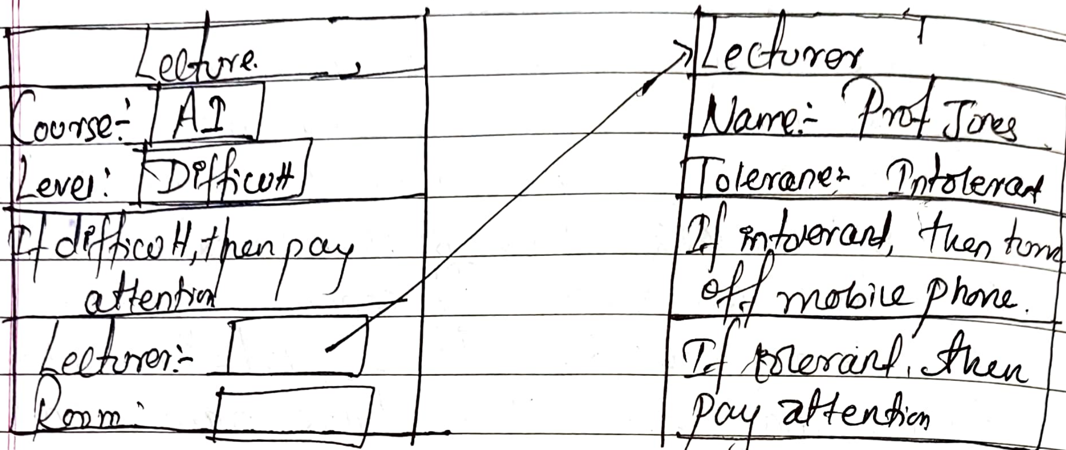
A frame is a record like structure which consists of a collection of attributes and its values to describe an entity in the world. Frames are the AI data structure which divide knowledge into substructures by representing stereotypes situations. It consists of slots and slot values collection. These slots may be any types and sizes. Slots have names and values which are called facets.

The idea of frame hierarchies is very similar to the idea of class hierarchies found in OOP. Frames are an application of object-oriented approach to knowledge-based systems. Frame is a type of technology which is widely used in various applications including natural language processing and machine visions.

Ex: Let's take an example of a frame for a book

Slots	Filters
Title	Artificial Intelligence.
Genre	Computer science.
Author	Peter Norvig
Edition	3rd Edition
Year	1996
Page	1152.

An example of two frames: Lecture and Lecturer connected with a link:



3. Conceptual Dependencies.

CD theory was developed to represent the meaning of Natural language sentences. It helps in drawing inferences 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.

- ATRANS → transfer of an abstract relationship. (i.e. give)
- PTRANS → transfer of the physical location of an obj. (eg. go)
- PROPEL → Application of physical force to an obj. (eg. push)
- MOVE → Movement of a body part by its owner (eg. kick)
- GRASP → Grasping of an obj. by an action (eg. Throw)
- INGEST → Ingesting of an obj by an animal (eg eat)
- EXPEL → Expulsion of something from the body of an animal (eg cry)
- SPEAK → Producing sounds (eg. say)
- ATTEND → Focusing of a sense organ toward a stimulus (eg. listen)
- MTRANS → Transfer of mental information (eg. tell)

- MBUILD: Building new information out of old (eg. decide)
- CONC - Conceptualize information (eg. think)

DATE

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

I ⇔ ATRANS ← book ← [→ man (to) ; ← I (from)]

Note: This example is same as saying: I gave the man a book, the man got a book from me, or the book was given to man by me etc.

* CD Notation:

- Arrows indicate direction of dependency.
- Double arrow indicates two way link between actor and action.
  - O - for the object case relation
  - R - for the receipt case relation.
  - P - for past tense
  - D - destination

Ex: John Ran

CD rep: John $\overset{P}{\Leftrightarrow}$ PTRANS.

~~Rule 2: ACT ← PP~~

Ex: John pushed the bike.

John $\overset{P}{\Leftrightarrow}$ PROPEL $\overset{O}{\leftarrow}$ bike

Ex: John is a doctor.

John ↔ doctor

Ex: John's dog.

dog ← John (poss-by) or dog ⇑ poss-by John

Note: "is" comes then use ↔ or ⇔

Ex: John is fat

John ⇔ weight (> 80)

Ex: Smart John.

John ← smart

Ex: John took the book from Mary.

John $\overset{P}{\Leftrightarrow}$ ATRANS $\overset{O}{\leftarrow}$ book $\overset{R}{\leftarrow}$ [→ John ; ← Mary]

Ex: A nice boy

boy ↑ nice

DATE

* Problems with CD representation

- It is difficult to construct original sentence from its corresponding CD representation
- Rules are to be carefully designed for each primitive action in order to obtain semantically correct interpretation.
- It becomes complex requiring lot of storage for many simple actions.

* Some more examples :-

Ex: John ate ice cream with a spoon

$$\text{John} \overset{p}{\Longleftrightarrow} \text{INGEST} \xleftarrow{\ o\ } \text{icecream} \xleftarrow{\text{with}} \text{spoon}$$

Ex: Jenny cried

$$\text{Jenny} \overset{p}{\Longleftrightarrow} \text{EXPEL} \xleftarrow{o} \text{tears}$$

eyes, poss by Jenny

Ex: Mike went to India

$$\text{Mike} \overset{p}{\Longleftrightarrow} \text{PTRANS} \xleftarrow{d} \begin{cases} \to \text{India} \\ \leftarrow \ ? \end{cases}$$

Ex: Tree grows

$$\text{Tree} \Longleftarrow \begin{cases} \to \text{size} > C \\ \leftarrow \text{size} = C \end{cases}$$

Ex: Bill shot Bob

{x}: Bill shot Bob

$\Uparrow$

{y}: Bob's health is poor

(upper arrow means 1st up event happen then down even happen.)

Ex: While going home I saw a snake (down arrow represents parallel events)

I am going home

$\Downarrow$

I saw a snake

✱ Some examples of Conceptual Dependency

1. Jenny cried.

=> Jenny $\overset{p}{\Leftrightarrow}$ EXPEL $\overset{o}{\leftarrow}$ tears $\overset{d}{\leftarrow}$ (→ ? ; ← eyes)

eyes $\Uparrow$ poss-by Jenny

2. Mike went to India.

Mike $\overset{p}{\Leftrightarrow}$ PTRANS $\overset{d}{\leftarrow}$ (→ India ; ← ?)

3. Mary read a novel.

Mary $\overset{p}{\Leftrightarrow}$ MTRANS $\overset{o}{\leftarrow}$ info $\overset{d}{\leftarrow}$ (→ CP(Mary) ; ← Novel.)

Ti (instrumental)

4. John is the tallest

John height (+10)

5. John is short

John height (< average)

6. Frank is dead.

Franz health (-10)

7. Dave is sad

Dave mental state (-10)

DATE

8. Since drugs can kill, I stopped.

o r

one ⇔ INGEST ← drugs ← → one's (body)

← mouth

↑ c

one → health → = -10

← > -10

↑ c/

I ⇔ INGEST ← drugs → body

← mouth.

Note:
c/ => negation.

9. King Ram marry Sita, the daughter of king Janak

sita

King Ram ⇔

DATE

4. Scripts

Scripts are used for knowledge representation. It is a structure that prescribes a set of circumstances which could be expected to follow from one another.
It is considered to consist of a no. of slots or frames but with more specialized roles.
They are useful in describing certain sterotyped situations such as going to theater.

Components of script:

(1) Roles :- Persons involved in an event. Ex: student
(2) Props :- Objects involved in an event (ex: pen, copy)
(3) Entry conditions : Conditions that need to be satisfied before event occur in script. (Ex :- Student should have id card)
(4) Results :- Conditions that will be true after event in script occurs. Ex :- Students returning filled answersheet after getting exam question
(5) Track :- Variations on the script. Ex: Exam centre
(6) Scenes : Sequence of event that occurs
Ex :- Scenes in an exam of student ex :-

Exam hall entry
↓
getting allocated seat
↓
getting paper
↓
writing answer
↓
submitting answer sheet

Advantages:
→ Event prediction is possible. Ex: If std. doesn't have card. Then they can't give exam

→ Single coherent interpretation may be build up of collection of observation

Disadvantage

→ Less general than frames
→ May not be suitable to represent all kind of Knowledge.

5) Rule Based Knowledge Representation System (RBS)

A rule-base system (or production system) is a knowledge Based System (KBS) in which the knowledge is stored as rules; an expert System is a RBSs in which the rules come from human experts in a particular domain.
The underlying idea of production systems is to represent knowledge in the form of condition-action pairs called production rules.
• If the condition C is satisfied then the Action A is appropriate

Ex: If it is raining then open umbrella.

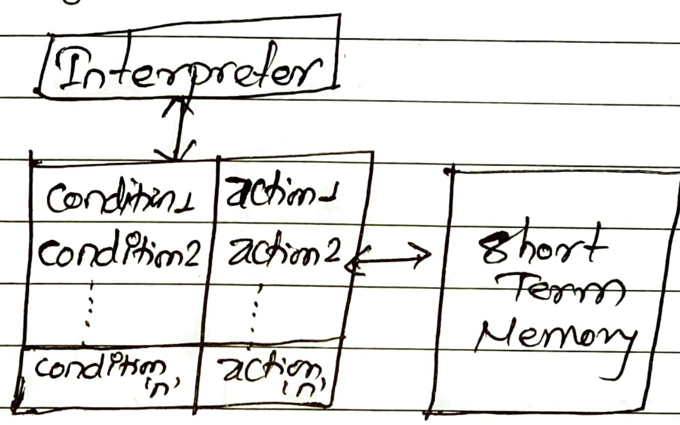
In a RBS, the knowledge is separated from the AI reasoning processes, which means that new RBSs are easy to create. An RBS can be fast, flexible & easy to expand.
Syntax of rule:-

IF <premises> THEN <action>

DATE

Components of RBs.

- Working Memory → Small memory into which only appropriate rules are copied.
- Rule base → the rules themselves, possibly stored specially to facilitate efficient access to the antecedent
- Interpreter → Process engine which carries out reasoning on the rules & derives an answer.



The interpretor cycle:- (Retrieval, Refinement & Execution)

→ Execution of a production system can be defined as the series of cycles:

- Match-memory contain matched against condition of production rules, this produces a subset of production called conflict set
- Conflict resolution - one of the production in the conflict set is then selected
- Apply the rule.

DATE

6) Logic Based Knowledge Representation:-

a) Propositional Logic.

A proposition is a declarative sentence that is either true or false but not both. Propositional logic is the simplest form of logic where all the statements are made by propositions. It is a technique for knowledge representation in logical and mathematical form.

Ex:→ Kathmandu is the capital of Nepal. (True propos

→ 2+5=8 (False proposition)

→ Close the door. (not a proposition)

* Knowledge base:- (KB) (Logic based)

It is a set of sentences represented in a knowledge representation language and represents assertions about the world. KB agents combine general knowledge with current percepts to infer hidden aspects of current state prior to setting action

Some abilities of knowledge based agents are:

- Represent states & actions.
- Incorporate new concepts
- Update internal representation of the world.
- Deduce hidden properties of the world.
- Deduce appropriate actions

DATE

Logic :- Logic are formal languages for representing information such that conclusions can be drawn. It has syntax & semantics

- Syntax defines the sentences in the language.
- Semantics define the meaning of sentences, that defines truth of a sentence in a world

Entailment : It means that one thing follows from another.

$KB \models \alpha$.

Knowledge base KB entails sentence α if and only if α is true in all worlds where KB is true.

Ex: $x+y=4$ entails $4=x+y$.

It is basically a relationship between sentences that is based on semantics.

Models: Logicians typically think in terms of models, which are formally structured worlds with respect to which truth can be evaluated.

m is a model of a sentence α if α is true in m

$M(\alpha)$ is the set of all models of α.

Logical Inference:- It is the process of deriving logical conclusions from premises known or assumed to be true. It is the act of reasoning from factual knowledge or evidence.

DATE

* Propositional Logic: Syntax:

The syntax of propositional 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.

We use symbols like P_1, P_2 to represent sentences. Propositional logic is defined as:-

If S is a sentence, $\neg S$ is a sentence (negation)

If S_1 and S_2 are sentences,

$S_1 \land S_2$ is sentence (conjunction)

$S_1 \lor S_2$ is sentence (disjunction)

$S_1 \rightarrow S_2$ is sentence (implies)

$S_1 \leftrightarrow S_2$ is sentence (double implie)

* Propositional Logic: Semantics.

Each model specifies true/false for each proposition symbol. Rules for evaluating truth with respect to a model:-

P	Q	$\neg P$	$P \land Q$	$P \lor Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
F	F	T	F	F	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
T	T	F	T	T	T	T

 Note: $P \leftrightarrow q$ is true if $p \rightarrow q$ is true & $q \rightarrow p$ is true

* Logical equivalence:

Two sentences α and β are logically equivalent ($\alpha \equiv \beta$) iff true they are true in same set of models:-

or, if $\alpha \models \beta$ and $\beta \models \alpha$.

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

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

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

$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$ 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 (Tautology)

A sentence is valid if it is true in all models.
ex: True, $A \vee \neg A$, $A \rightarrow A$, $(A \wedge (A \rightarrow B)) \rightarrow B$

Every valid sentence is logically equivalent to true.

* Satisfiability

A sentence is satisfiable if it is true in some model. Eg. $A \vee B$.

It is unsatisfiable if it is true in no model.
Eg. $A \wedge \neg A$

Validity and satisfiability are related concepts.
- α is valid iff $\neg\alpha$ is unsatisfiable.
- α is satisfiable iff $\neg\alpha$ is not valid.

Satisfiability is connected to inference via the following.
- $KB \models \alpha$ iff $(KB \wedge \neg\alpha)$ is unsatisfiable

* Inference rules in PL.

1. Modus Ponens:

It states that if P and $P \to Q$ is true, then we can infer that Q will be true.

$$\frac{P,\ P \to Q}{\therefore Q}$$

Ex: If I am sleepy then I go to bed $(P \to Q)$
I am sleepy. (P)
$\Rightarrow$ Then conclusion: I go to bed. (Q).

Proof: use truth table

2. Modus Tollens.

It states that if $P \to Q$ is true, and $\neg Q$ is true, then $\neg P$ will also be true.

$$\frac{P \to Q,\ \neg Q}{\text{then} \quad \neg P}$$

Ex: If I am sleepy I go to bed $(P \to Q)$
I do not go to bed $(\neg Q)$
Conclusion: I am not sleepy $(\neg P)$

3. Hypothetical Syllogism.

It states that if $P \to R$ is true whenever $P \to Q$ is true, and $Q \to R$ is true.

$$\frac{P \to Q, Q \to R}{\therefore P \to R}$$

4. Disjunctive Syllogism.

It states that if $P \vee Q$ is true, and $\neg P$ is true, then Q is true.

$$\frac{P \vee Q, \neg P}{\therefore Q}$$

Ex: Today is sunday or monday, $(P \vee Q)$
Today is not sunday $(\neg P)$
conclusion; Today is monday (Q)

5. Addition.

It states that if P is true, then $P \vee Q$ will be true

$$\frac{P}{P \vee Q}$$

Note: $\frac{A \wedge B}{\neg A}$ (elimination)

6. Simplification:

It states that if $P \wedge Q$ is true, then Q or P will also be true:

$$\frac{P \wedge Q}{Q} \text{ or } \frac{P \wedge Q}{P}$$

Note: $\frac{p \quad q}{p \wedge q}$ (conjunction)

7. Resolution: (Generalized)

It states that if $P \vee Q$ and $\neg P \wedge R$ is true then $Q \vee R$ will also be true.

$$\frac{P \vee Q, \neg P \wedge R}{Q \vee R} \quad \Bigg| \quad \frac{P \vee Q, \neg P \vee R}{Q \vee R}$$

DATE

- Resolution uses CNF (Conjunctive Normal Form)
- The resolution rule is sound.
- Resolution is complete in the sense that it can always be used to either confirm or refuse a sentence. (it can't be used to enumerate true sentences).

* Conversion to CNF / Resolution by using CNF

A sentence that is expressed as a conjunction of disjunctions of literals is said to be in conjunctive normal form (CNF)

A sentence in CNF that contains only k literals per clause is said to be in k-CNF.

Rules to Convert PL to CNF :-

① Remove bicondition $\Leftrightarrow$

$$A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$$

② Remove implication $\rightarrow$

$$A \rightarrow B \equiv \neg A \vee B$$

③ Move negation inwards using De Morgan's rule

$$\neg(A \vee B) = \neg A \wedge \neg B$$

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

$$\neg(\neg A) \equiv A$$

④ Apply distributive Law

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

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

Example: Convert following KB into CNF

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

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

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

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

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

Ex: $(A \to B) \vee (C \to B)$

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

Ex: $(A \vee B) \to C$

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

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

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

* Resolution Algorithm

process (algorithm)

i) Convert given KB to 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

Ex: Consider the Knowledge Base (KB)

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

Prove $\neg A$ can be inferred from above KB by resolution

1st Convert the KB into CNF

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

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

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

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

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

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

Now, add negation of sentence to be inferred from KB: A

Now, KB contains following sentences all in CNF

Premises: $(\neg B \vee A \vee C)$

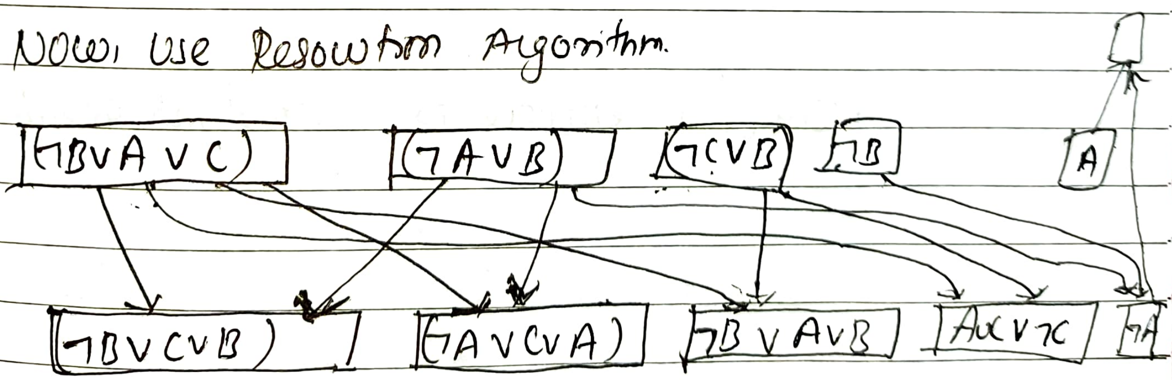
$(\neg A \vee B)$

$(\neg C \vee B)$

$\neg B$

A

Now, use Resolution Algorithm.



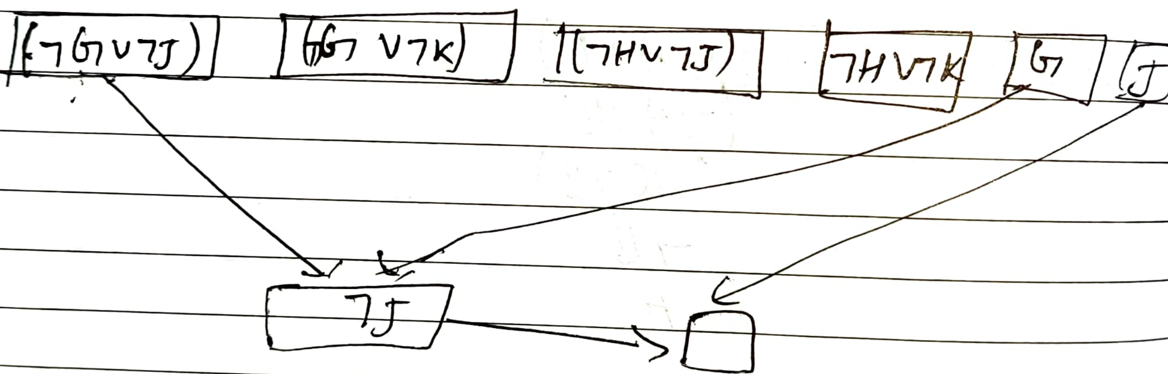
Ex: Given $KB = \{(G \vee H) \rightarrow (\neg J \wedge \neg K), G\}$. Show that the given KB entail $\neg J$ or not

⇒ 1st Convert KB into CNF

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

Add negation: $\neg(\neg J) \equiv J$

now,



Hence, the sentence is entitled into given KB

Ex: Given $KB = \{P \rightarrow \neg Q, \neg Q \rightarrow R\}$. Show that the Sentence $\neg P \rightarrow Q$ is entailed

⇒ Step-1:

$P \rightarrow \neg Q$ | $\neg Q \rightarrow R$ | $\neg P \rightarrow Q$

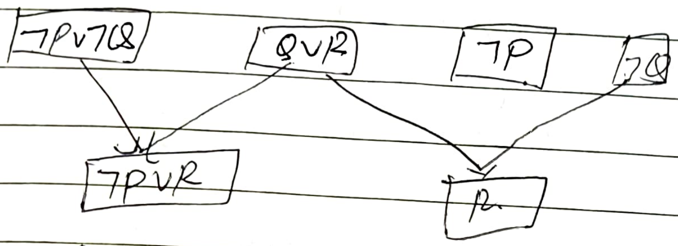
$\equiv \neg P \vee \neg Q$ | $\equiv Q \vee R$ | $\equiv P \vee Q$

Step-2:-

$P \vee Q$

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

Step-3:-



Hence, the sentence is not entailed into given KB

Well formed formula:

It is the syntactic 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). Any expression that obeys the syntactic rules of propositional logic is wff.

The syntax of propositional logic is easy to learn. It has only three rules:

1) Any capital letter by itself is a WFF
2) Any WFF can be prefixed with "~". (The result will be WFF too)
3) Any two WFFs can be put together with ".", "v", "⊃", or "≡" between them, enclosing the result in parenthesis. (This will be a WFF too)

Ex: A, $\sim A$, $(\sim A \cdot B)$, $\sim((\sim A \cdot B) \supset C)$

ex of non WFF: $A\sim$, (A)

DATE

* Forward Chaining and backward Chaining

FC and BC are the two most important strategies in the field of AI and lie in the expert system domain of AI. They are used by the inference engine in making the deduction.

(Inference Engine): It is the component of intelligent system in AI, which applies logical rules to the KB to infer new information from known facts. It commonly proceeds in two modes: forward chaining & backward chaining

(Horn clause):- A clause which is a disjunction of literals with at most one positive literal is known as horn clause. All definite clauses are horn clause.

(Definite clause):- A clause which is a disjunction of literals with exactly one positive literal.
Ex: $(\neg p \vee \neg q \vee k)$. It has only one +ve literal.

It is equivalent to $p \wedge q \rightarrow k$

(i) Forward Chaining

It is a method of reasoning in AI in which inference rules are applied to existing data to extract additional data until an endpoint (goal) is achieved.

Here, the inference engine starts by evaluating existing facts, derivations & conditions before deducing new information.

An endpoint (goal) is achieved through the manipulation of existing knowledge in KB.

- It is a ~~top~~ down-up approach
- Also called data driven reasoning as we reach to goal using available data
- ~~Modus pones inference rule is used~~

Ex:

$$\frac{\begin{array}{c} A \\ A \rightarrow B \end{array}}{B}$$

He is running.
If he is running, he sweats.
He is sweating

Steps:

→ Given knowledge base of true fact.
→ Apply all rules that match facts in KB.
→ Add conclusions to KB
→ Repeat until goal is reached or no new fact is added

Ex: Consider the following facts:-

If it is sunny & warm day, you will enjoy
If it is raining, you'll get wet
It is sunny
It is warm
It is raining

Goal: you will enjoy.

Soln

P: It is sunny

Q: It is warm

R: You'll enjoy

S: It is raining

T: You'll get wet

Premises: $P \wedge Q \rightarrow R$

$S \rightarrow T$

P

Q

S

Goal: R

Negation: $\neg R$

Now, convert into CNF

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

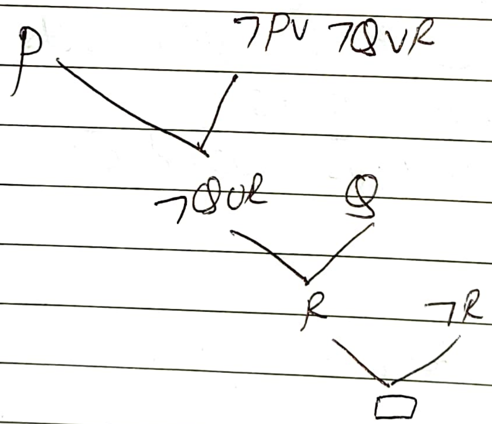
$\neg S \vee T$

P

Q

S

Tree is:



Hence, You'll enjoy.

Qn:

$P \rightarrow Q$

$L \wedge H \rightarrow P$

$B \wedge L \rightarrow H$

$A \wedge P \rightarrow L$

$A \wedge B \rightarrow L$

A

B

Show that Q can be inferred from given KB

b) Backward Chaining.

It is a concept in AI that involves backtracking from endpoint or goal to steps that lead to the endpoint.

It starts from the goal & moves backward to comprehend the steps that were taken to attain this goal.

- It uses up-down approach
- It is based on modus pones inference rule.
- The goal is broken into sub-goal or sub-goals to prove the facts true
- Also called goal-driven approach, as list of goals decides which rules are selected & used.

Ex: $\dfrac{\begin{matrix} B \\ A \to B \end{matrix}}{A}$

Tom is sweating
If a person is running, he will sweat
Tom is running

Note:
Avoid loops: Check if new subgoal is already on the goal stack.
Avoid repeated work: Check if new subgoal
1. has already been proved true, or
2. has already failed.

DATE

PAGE

DATE

Ex: In above KB

$P \rightarrow Q$

$L \wedge M \rightarrow P$

$B \wedge L \rightarrow M$

$A \wedge P \rightarrow L$

$A \wedge B \rightarrow L$

A

B

Prove that Q can be inferred from above KB.

=) We know, $P \rightarrow Q$, so we try to prove P

$L \wedge M \rightarrow P$

Try to prove L and M

$B \wedge L \rightarrow M$

$A \wedge P \rightarrow L$

Try to prove, B, L, A, P.

A and B is already known, Since $A \wedge B \rightarrow L$,
L is also known

Since, $B \wedge L \rightarrow M$, M is also known

Since, $L \wedge M \rightarrow P$, P is known, hence the proof

b) Predicate Logic : FOPL (First-order Predicate logic)

FOPL is a symbolized reasoning in which each sentence or statement is broken down into a subset and predicate, where predicate is considered as a function and relationship of subject.

A sentence in FOPL is written in the form of Px or P(x), where P is predicate and x is subject

Atomic sentence
Ex: man(Ram) => Ram is a man.
brother(Ram, Hari) => Ram is brother of Hari
married(Puran, Sita) => Puran is married to Sita
brotherOf(Ram) = Hari => Ram is brother of Hari

Atomic Sentence: It is formed from a predicate symbol followed by a parenthesized list of terms

Complex Sentence: They are made from atomic sentences using connective.
Ex: Father(Jack, John) ∧ Mother(Jill, John) ∧ Sister(Jane, John)
¬ Sister(John, Jane)
Older(Father-of(John), 30)

* FOPL Syntax:

Sentence → Atomic sentence / Quantifier / ... Sentence / ¬sentence
Atomic Sentence → Predicate(term) / Term → term
Term → function (term) / Term → term
Connective → ∧ / ∨ / ¬ / → / ↔

classmate

Quantifier → ∀, ∃
Constant → A, B, C, x_1, x_2, Ram, Hari....
Variable → x_1, x_2, counter position, ..
Predicate → brother, has color, ..
Function → father of sqrt, cosine, ...

* FOPL Semantics:
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 domain
→ function symbols
→ predicate symbols
Then we can define universal & existential quantifier.

Object/Subject could be specified or unspecified

Eg: loves(Ram, Sita)
Loves(x, y)
hot(x) → hot(day)

* Representing knowledge in FOPL

- Objects from the real world are represented by constant symbols (a, b, c). Ex: "Ram" may represent any individual named "Ram".
- Predicate of objects may be represented by predicates applied to those objects.
Ex: male(Ram) represents the fact that Ram is male
- The value of predicate is one of the Boolean Constants
Father(Dashrath, Ram) = True (T)
Father(Ram, Dashrath) = F

DATE

- Besides constants, the arguments of the predicates can be functions, as well:-

Ex: weight (Raj) = 83

greaterthan (weight (Raj), 82)

* Quantifiers (Quantification)

A quantifier is a language element which generates quantification, & quantification specifies the quantity of specimen in the universe of ~~disclosure~~ discourse.

These are the symbols that permit to determine or identify the range and scope of the variable in the logic expression.

There are two types of quantifier:

1. Universal quantifier (for all, anyone, everyone, everything)

It is a symbol of logic representation, which specifies that the statement within its range is true for everything or every instance of a particular thing. It is represented by symbol $\forall$. and its syntax is $\forall$ <variables> <sentence>

Ex: All man play football.

→ Let's consider x refers to man so all x can be represented as Universe of Disclosure (UD) as:

x_1 plays football

x_2 plays football

----

x_n plays football.

$\therefore \forall x$ man$(x) \rightarrow$ play (x, football)

classmate PAGE

2. Existential quantifier (for some, at least one)

It is the type of quantifier, which express that the statement within its scope is true for at least one instance of something.
It is denoted by $\exists$,
If x is a variable, then existential quantifier will be $\exists x$ or $\exists(x)$
Here, we always use AND i.e. conjunction symbol "$\wedge$"

Ex: Some boys are intelligent.
$\Rightarrow$ $\exists x$ boys$(x) \wedge$ intelligent(x)

$$\left(\begin{array}{c} x_1 \\ \vee \\ x_2 \\ \vee \\ x_3 \\ \vee \\ x_n \end{array}\right)$$

* Properties of quantifier.

- $\forall x \forall y$ is similar to $\forall y \forall x$ (in universal quantifier)
$\Rightarrow$ "Everyone loves anything." is similar to anything is loved by everyone

- $\exists x \exists y$ is similar to $\exists y \exists x$ (in Existential
$\Rightarrow$ "There is a person who hates a girl." is similar to A girl is hated by a person

- $\exists x \forall y$ is not same as $\forall y \exists x$
$\Rightarrow$ "There is a person who loves everyone." is not similar to "Everyone is loved by at least one person". [or someone loves everyone $\neq$ everyone is loved by someone]

$$\forall y \exists x \text{ loves}(x,y) \neq \exists x \forall y \text{ loves}(x,y)$$

Not: $\forall x$ Taking $(x, AI) \wedge$ Smart(x)
$\Rightarrow$ It means Everyone is taking AI & everyone is smart (which is wrong)

- **Quantifier duality:-** each can be expressed using another

$\forall x$ likes$(x$, Icecream$) \equiv \neg \exists x \neg$ likes$(x$, Icecream$)$
$\exists x$ likes$(x$, Brocoli$) \equiv \neg \forall x \neg$ likes$(x$, Brocoli$)$

$\rightarrow$ Everyone loves icecream is similar to, there is no one who does not like icecream.

Note:

$\forall x \neg P \equiv \neg \exists x P$	$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$
$\neg \forall x P \equiv \exists x \neg P$	$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$
$\forall x P \equiv \neg \exists x \neg P$	$P \wedge Q \equiv \neg(\neg P \vee \neg Q)$
$\exists x P \equiv \neg \forall x \neg P$	$P \vee Q \equiv \neg(\neg P \wedge \neg Q)$

<u>Some practise questions</u> (convert sentences into FOL)

1. All birds fly.
$\Rightarrow$ $\forall x$ birds$(x) \rightarrow$ fly(x)

Q. Someone is happy
$\Rightarrow$ $\forall x$ happy(x)

2. Every man respects his parent.
$\Rightarrow$ $\forall x$ man$(x) \rightarrow$ respects$(x$, parent$)$

3. Some boys play cricket
$\Rightarrow$ $\exists x$ boys$(x) \wedge$ play$(x$, cricket$)$

4. Not all students like both Maths & Science
$\Rightarrow$ $\neg \forall(x)$ student$(x) \rightarrow$ like$(x$, Maths$) \wedge$ like$(x$, science$)$

DATE

5. Student are people who are enrolled in course.

⇒ $\forall x: student(x) \longrightarrow people(x) \land enrolled(x, course)$

6. father are male parent with child

⇒ $\forall x: father(x) \longrightarrow male(x) \land has\_child(x)$

7. One's mother is one's female parent

⇒ $\forall x \forall y\ mother(x, y) \leftrightarrow female(x) \land parent(x, y)$

x is a mother of y iff x is female & x is parent of y

8. Every person who buys a policy is smart.

⇒ $\forall x \forall y: Person(x) \land policy(y) \land buys(x, y) \longrightarrow smart(x).$

or $\forall x\ Person(x) \land buys(x, policy) \longrightarrow smart(x)$

9. No person buys expensive policy

⇒ $\forall x \forall y: person(x) \land policy(y) \land expensive(y) \longrightarrow \neg buys(x, y)$

or $\neg \forall x\ person(x) \longrightarrow buys(x, exp\_policy)$

10. All Children like Apples

⇒ $\forall x: Child(x) \rightarrow likes(x, Apples)$

11. Everyone likes apples unless they are allergic to it.

⇒ $\forall x: likes(x, apples) \lor allergic(x, apples)$

or

$\forall x\ \neg allergic(x, apples) \rightarrow likes(x, apples)$

11. Everyone likes sweets.

⇒ $\forall x: likes(x, sweet)$ or $\forall x\ p(x) \rightarrow likes(x, sweet)$

where, p = person

DATE

12. Everyone likes some kind of food

$\Rightarrow$ $\underline{\forall x \exists y}$ [food(y) $\wedge$ likes(x, y)]

13. There is a kind of food that everyone likes

$\Rightarrow$ $\underline{\exists y \forall x}$ [food(y) $\wedge$ likes(x, y)]

14. Someone likes all kind of food.

$\Rightarrow$ $\underline{\exists x \forall y}$ [food(y) $\wedge$ likes(x, y)]

15. Every food has someone who likes it

$\Rightarrow$ $\underline{\forall y \exists x}$ [food(y) $\wedge$ likes(y, x)]

16. Not everyone like Apples.

$\Rightarrow$ $\neg \forall x$ likes(x, Apples) or $\exists x \neg$likes(x, Apples)

17. No one likes apples.

$\Rightarrow$ $\neg \exists x$ likes(x, Apples) or $\forall x \neg$likes(x, Apples)

18. Brothers are sibling

$\Rightarrow$ $\forall x, y$ Brother(x, y) $\longrightarrow$ sibling(x, y)

It is same as saying, if x is a brother of y then x is a sibling of y.

19. Not everyone like sweets.

$\Rightarrow$ $\neg \forall x$ [p(x) $\longrightarrow$ likes(x, sweet)]

It can be written as: $\neg \forall x$ [$\neg$p(x) $\vee$ likes(x, sweet)]

$\neg \forall x \neg$ [p(x) $\wedge$ $\neg$likes(x, sweet)]

$\exists x$ [p(x) $\wedge$ $\neg$likes(x, sweets)]

We used $\boxed{\exists x \cong \neg \forall x \neg}$

classmate

PAGE

DATE

* Free and bound variable.

A variable is said to be free in a formula if it occurs outside the scope of the quantifier
Ex: $\forall x \exists(y) [p(x,y,z)]$, where z is free variable

A variable is said to be bound in a formula if it occurs within the scope of the quantifier
Ex: $\forall x [A(x) B(y)]$, here x & y are bound variables

* Inference in FOPL :/ Inference in FOL (FOL to FOPL)

FOPL inference can be done by converting the knowledge base to predicate logic & using propositional inference. The major task in inference of FOPL is to deal with quantifier so instantiation is required
There are two types of instantiation

1. Universal Instantiation (UI):-

Also called as universal elimination or UI is a valid inference rule. It can be applied multiple times to add new sentences. The new KB is logically equivalent to previous KB. It states that we can infer any sentence p(c), by substituting a ground term c from $\forall x\, p(x)$ for any object in the universe of discourse.

Note: C is a constant within domain x

It can be represented as $\dfrac{\forall x\, p(x)}{p(c)}$

Ex: If "Every person like ice-cream" => $\forall x\, P(x)$ so we can infer that "John likes ice-cream" => $P(C)$

Ex: "All kings who are greedy are evil."
=> $\forall x$ king(x) $\wedge$ greedy(x) $\rightarrow$ evil(x).
we can infer

- King (John) $\wedge$ Greedy (John) $\rightarrow$ Evil (John)
- King (Father (John)) $\wedge$ Greedy (Father (John)) $\rightarrow$ Evil (Father (John))
- King (brother_of (John))

2. Existential Instantiation (EI)

Also called Existential elimination, which is a valid inference rule in first-order logic. It can be applied only once to replace the existential sentence. The new KB is not logically equivalent to old KB, but it will be satisfiable if old KB was satisfiable.

This rule states that one can infer $P(C)$ from the formula given in the form of $\exists x\, P(x)$ a new constant symbol c. The restriction with this rule is that c is used in the rule must be a new term for which $P(C)$ is true.

It can be represented as: $\dfrac{\exists x\, P(x)}{P(C)}$

Someone goes to school
=> $\exists x\, P(x) \wedge$ goes (school, x) {P = Person}

It can be:

$P(C) \wedge$ goes (school, C)

Note: Existential instantiation is a special case of skolemization. Here, C is skolem constant.

DATE

* Unification and lifting:

→ Unification is the process we use to find substitutions that make different logical expressions look identical. Algorithm: UNIFY(p, q) = θ where SUBST(θ, p) = SUBST(θ, q), θ is our unifier value.

→ Lifting means making only those substitutions that are required to allow particular inferences to proceed.

Lets take two literals as input and make them identical using substitution. Let A and B be two atomic sentences and σ be a unifier such that Aσ = Bσ, then it can be expressed as UNIFY (A, B)

Ex: Find the Most General Unifier (MGU) for Unify{ King(X), King(John)}

→ Let, A = King(x), B = King(John)

Substitution θ = {John / x} is a unifier for these atoms & applying these substitution then both expressions will be identical.

o Unification is a key component of all first-order inference algorithms

o It returns fail if the expressions dont match with each other

Another Example of unification is Generalized Modus Pones rule

classmate

PAGE

DATE

* Generalized Modus ponens rule:-

For the inference process in FOL, we have a single inference rule which is called Generalized modus ponens. It is the lifted version of modus ponens.

It can be summarized as, "P implies Q and P is asserted to be true, therefore Q must be true.

According to Modus ponens, for atomic sentences p_i, p_i', q. Where there is a substitution θ such that $(\theta, p_i') =$ SUBST(θ, p_i), it can be represented as:

$$\frac{p_1', p_2', \text{----}, p_n' \ (p_1 \land p_2 \land \text{----} \land p_n \Rightarrow q)}{\text{SUBST}(\theta, q)}$$

* Resolution of ~~CNF~~ FOPL:-

- → Same as propositional logic.
- → Allows extended syntax such as quantifier
- → Define casual from casual form
- → Eliminate quantifiers from causual form
- → Adopt resolution procedure to cope with variables i.e. unification

Note: You can do in any order after step-1

DATE

Steps to convert FOL into CNF

- Eliminate implication $\left\{ \begin{array}{l} \alpha \to \beta \equiv \neg\alpha \vee \beta \\ \alpha \leftrightarrow \beta \equiv (\alpha \to \beta) \vee (\beta \to \alpha) \end{array} \right.$

- Standardize variable.

$\exists(x)$ smile(x)	$\exists(x)$ Smile(x)
$\exists x$ graduating(x)	$\exists(y)$ graduating(y)
$\forall x$ happy(x)	$\forall(z)$ happy(z)

- Move negation inwards

$$\neg(\forall(x)\, p(x)) = \exists x\, \neg p(x)$$
$$\neg(\exists x\, p(x)) \equiv \forall x\, \neg p(x)$$
$$\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$$
$$\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$$
$$\neg(\neg\alpha) \equiv \alpha$$

- Skolemization (Remove existential quantifier)

$\exists(x)$ Smile(x)

$\exists(y)$ graduating(y)

After Skolemization

Smile(A)

Graduating(B)

- Drop universal quantifier

$\forall(x)$ Smile(x)

$\forall(y)$ graduating(y)

After dropping

Smile(x)

graduating(y)

Some examples to convert sentences to FOPL

1. Bill is a student

⇒ Student(Bill)

2. Bill takes either analysis or geometry

⇒ Takes(Bill, Analysis) $\vee$ Takes(Bill, geometry)

3. Bill takes analysis or geometry but not both at same time.

⇒ Takes(Bill, Analysis) $\longleftrightarrow \neg$ Takes(Bill, Geometry)

4. Some students love Bill

⇒ $\exists x$ student(x) $\wedge$ love(x, Bill)

5. Jym collects everything

⇒ $\forall x$ collects(Jym, x)

6. Somebody collects something

⇒ $\exists x, y$ Collects(x, y)

7. No stinky shoes are allowed

⇒ $\exists x$ shoes(x) $\wedge$ stinky(x) $\wedge$ $\neg$ allowed(x).

DATE

8. Good people always have good friends

→ $\forall x$ person$(x) \wedge$ Good$(x) \rightarrow \exists y$ friend(x,y)

Inference using resolution

Q. How resolution algo. is used in FOPL to infer conclusion.

→ Steps:

1. Conversion of facts into FOL
2. Convert FOL to CNF – [Explain this also]
3. Negate the statement need to prove
4. Draw resolution graph tree (Unification)

Ex: • All people who are graduating are happy
• All happy people smile.
• Someone is graduating

Use resolution to infer that someone is smiling

⇒ Steps: Convert into FOL

$\forall x$ Graduating$(x) \rightarrow$ happy(x)

$\forall x$ happy$(x) \rightarrow$ smile(x)

$\exists x$ Graduating(x)

Goal: $\exists x$ ~~smiling~~ smile(x)

Negation: $\neg \exists x$ smile(x)

classmate

DATE

step 2: Convert to CNF

- Eliminate Implication.

$\forall x [\neg Graduating(x) \vee Happy(x)]$
$\forall x [\neg Happy(x) \vee Smile(x)]$
$\exists x \; Graduating(x)$
$\neg \exists x \; Smile(x)$

- Standardize variable.

$\forall x [\neg Graduating(x) \vee Happy(x)]$
$\forall y [\neg Happy(y) \vee smile(y)]$
$\exists z \; Graduating(z)$
$\neg \exists w \; Smile(w)$

- Negation Inwards.

$\forall x [\neg Graduation(x) \vee Happy(x)]$
$\forall y [\neg Happy(y) \vee Smile(y)]$
$\exists z \; Graduating(z)$
$\forall w \; \neg Smile(w)$

- Skolemization.

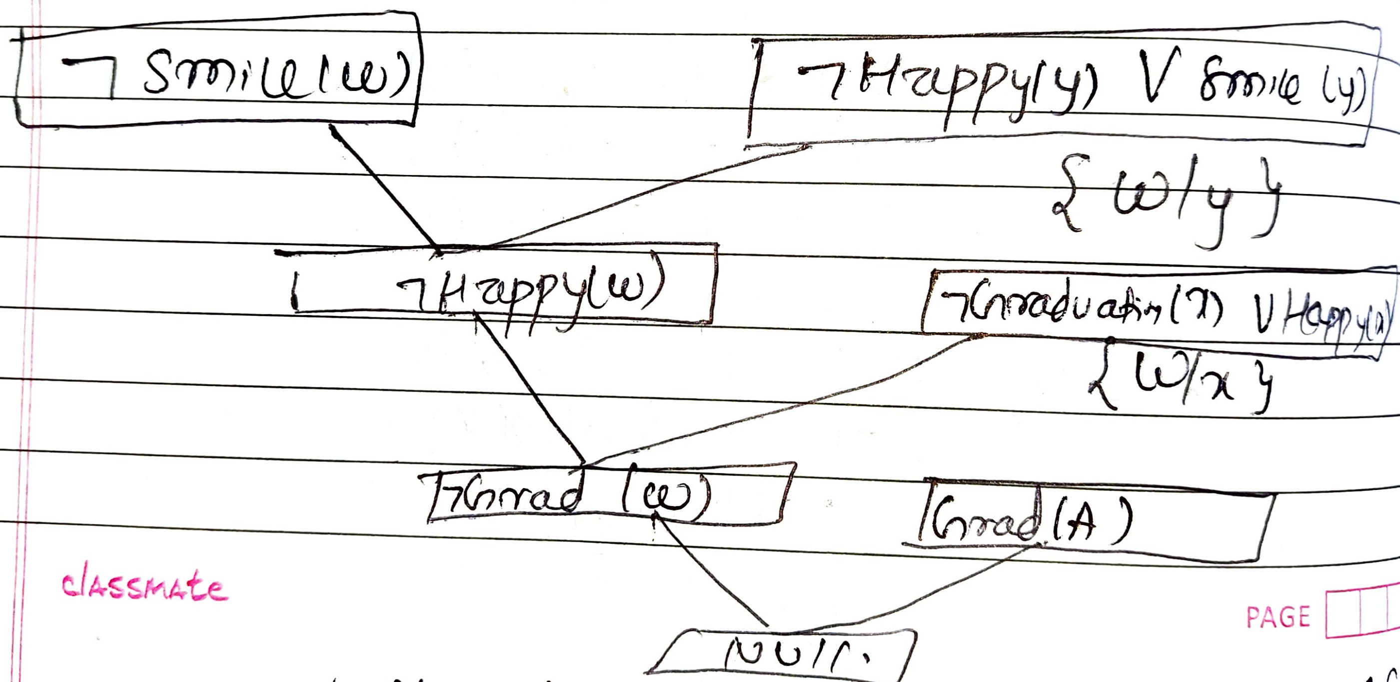
$\forall x [\neg Graduation(x) \vee Happy(x)]$
$\forall y [\neg Happy(y) \vee Smile(y)]$
$Graduating(A)$
$\forall w \; \neg smile(w)$

- Drop universal Quantificas

$\neg Graduation(x) \vee Happy(x)$
$\neg Happy(y) \vee Smile(y)$
$Graduating(A)$
$\neg Smile(w)$

Step-3: Draw reson tree:-

Note: Start with negation of goal



Hence, by ~~conduction~~, Contradiction, someone is smil

DATE

Ex: John likes all kind of food
Apple and vegetables are food
Anything anyone eats and not killed is food
Anil eats peanuts and is still alive
Harry eats everything that anil eats

Goal: John likes peanuts.

⇒ Step-1: Convert into FOL.

$\forall x \; food(x) \rightarrow likes(John, x)$

$food(apple) \wedge food(vegetables)$

$\forall x \forall y \; eats(x, y) \wedge \neg killed(x) \rightarrow food(y)$

$eats(Anil, peanuts) \wedge alive(Anil)$

$\forall x : eats(Anil, x) \rightarrow eats(Harry, x)$

$\left.\begin{array}{l} \forall x : \neg killed(x) \rightarrow alive(x) \\ \forall x : alive(x) \rightarrow \neg killed(x) \end{array}\right\}$ added predicates

$likes(John, peanuts)$

Step-2: Convert into CNF

- Eliminate implication

$\forall x \; \neg food(x) \vee likes(John, x)$

$food(apple) \wedge food(vegetable)$

$\forall x \forall y \; \neg eats(x, y) \vee killed(x) \vee food(y)$

$eats(Anil, Peanuts) \wedge alive(Anil)$

$\forall x : \neg eats(Anil, x) \vee eats(Harry, x)$

$\forall x : killed(x) \vee alive(x)$

$\forall x : \neg alive(x) \vee \neg killed(x)$

Note: you can convert to CNF / use steps in any order

- Move negation Inwards

$\forall x\ \neg food(x) \vee likes(John, x)$
$food(apple) \wedge food(vegetables)$
$\forall x \forall y\ \neg eats(x, y) \vee killed(x) \vee food(y)$
$eats(Anil, peanuts) \wedge alive(Anil)$
$\forall x\ \neg eats(Anil, x) \vee eats(Harry, x)$
$\forall x\ \neg killed(x) \vee alive(x)$
$\forall x\ \neg alive(x) \vee \neg killed(x)$
$likes(John, peanuts)$

- Eliminate existential quantification (Skolemization) not necessary here, since there are none

- Drop all universal quantifier & write '$\wedge$' separately

$\neg food(x) \vee likes(John, x)$
$food(apple), food(vegetables)$
$\neg eats(x, y) \vee killed(x) \vee food(y)$
$eats(Anil, peanuts)$
$alive(Anil)$
$\neg eats(Anil, x) \vee eats(Harry, x)$
$\neg killed(x) \vee alive(x)$
$\neg alive(x) \vee \neg killed(x)$
$likes(John, peanuts)$

- Standardize.

$\neg food(x) \vee likes(John, x)$
$food(apple)$
$food(vegetables)$
$\neg eats(y, z) \vee killed(y) \vee food(z)$
$eats(Anil, peanuts)$
$alive(Anil)$
$\neg eats(Anil, p) \vee eats(Harry, p)$
$\neg killed(q) \vee alive(q)$
$\neg alive(r) \vee \neg killed(r)$
$likes(John, peanuts)$

classmate

PAGE

Note: Write '∧' independently:

- food(apple)
- food(vegetable)

eats(Anil, peanuts)
alive(Anil)

Step 3: Negate the goal

$\neg$ likes(John, Peanuts)

Step-4: Draw resolution graph

$\neg$ likes(John, peanuts)

$\neg$ food(x) V likes(John, x)

{peanuts/x}

$\neg$ food(peanuts)

$\neg$ eats(y, z) V killed(y) V food(z)

{peanuts/z}

$\neg$ eats(y, peanuts) V killed(y)

eats(Anil, peanuts)

{Anil/y}

killed(Anil)

$\neg$ alive(r) V $\neg$ killed(r)

{Anil/r}

$\neg$ alive(Anil)

alive(Anil)

NULL

Hence, proved.

By contradiction and

Ex: Anyone passing his/her history exam and winning lottery is happy.
Anyone who studies or is lucky can pass all his exams
John didn't study but he is lucky
Anyone who is lucky wins the lottery

Prove: John is happy.

⇒ Step 1: Convert into FOL

$\forall x \; pass(x, history) \land wins(x, lottery) \rightarrow happy(x)$
$\forall x \forall y \; studies(x) \lor lucky(x) \longrightarrow pass(x, y)$
$\neg study(John) \land lucky(John)$
$\forall x \; lucky(x) \longrightarrow wins(x, lottery)$

Goal: happy(John)
Negation: $\neg happy(John)$

Step-2: Convert FOL to CNF

- Eliminate implication & negation & existential

* $\forall x \neg pass(x, history) \lor \neg wins(x, lottery) \lor happy(x)$

$\forall x \forall y \; [\neg study(x) \land \neg lucky(x)] \lor pass(x, y)$
$\Rightarrow \forall x \forall y \; (\neg study(x) \lor pass(x, y)) \land (\neg lucky(x) \lor pass(x, y))$
we have two CNF:
* $\Rightarrow \forall x \forall y \; (\neg study(x) \lor pass(x, y))$
* $\Rightarrow \forall x \forall y \; (\neg lucky(x) \lor pass(x, y))$

* $\neg study(John)$
* $lucky(John)$
* $\forall x \; \neg lucky(x) \lor wins(x, lottery)$
Goal: * happy(John) Negation: $\neg happy(John)$

DATE

- Drop all universal quantifier

$\neg pass(x, history) \vee \neg wins(x, lottery) \vee happy(x)$

$\neg study(x) \vee pass(x, y)$

$\neg lucky(x) \vee pass(x, y)$

$\neg study(John)$

$lucky(John)$, $\neg lucky(x) \vee wins(x, lottery)$

no need to write goal. [Goal: $happy(John)$
negation. $\neg happy(John)$] only write negation

- Standardize.

$\neg pass(x, history) \vee \neg wins(x, lottery) \vee happy(x)$

$\neg study(y) \vee pass(y, z)$

$\neg lucky(p) \vee pass(p, q)$

$\neg study(John)$

$lucky(John)$

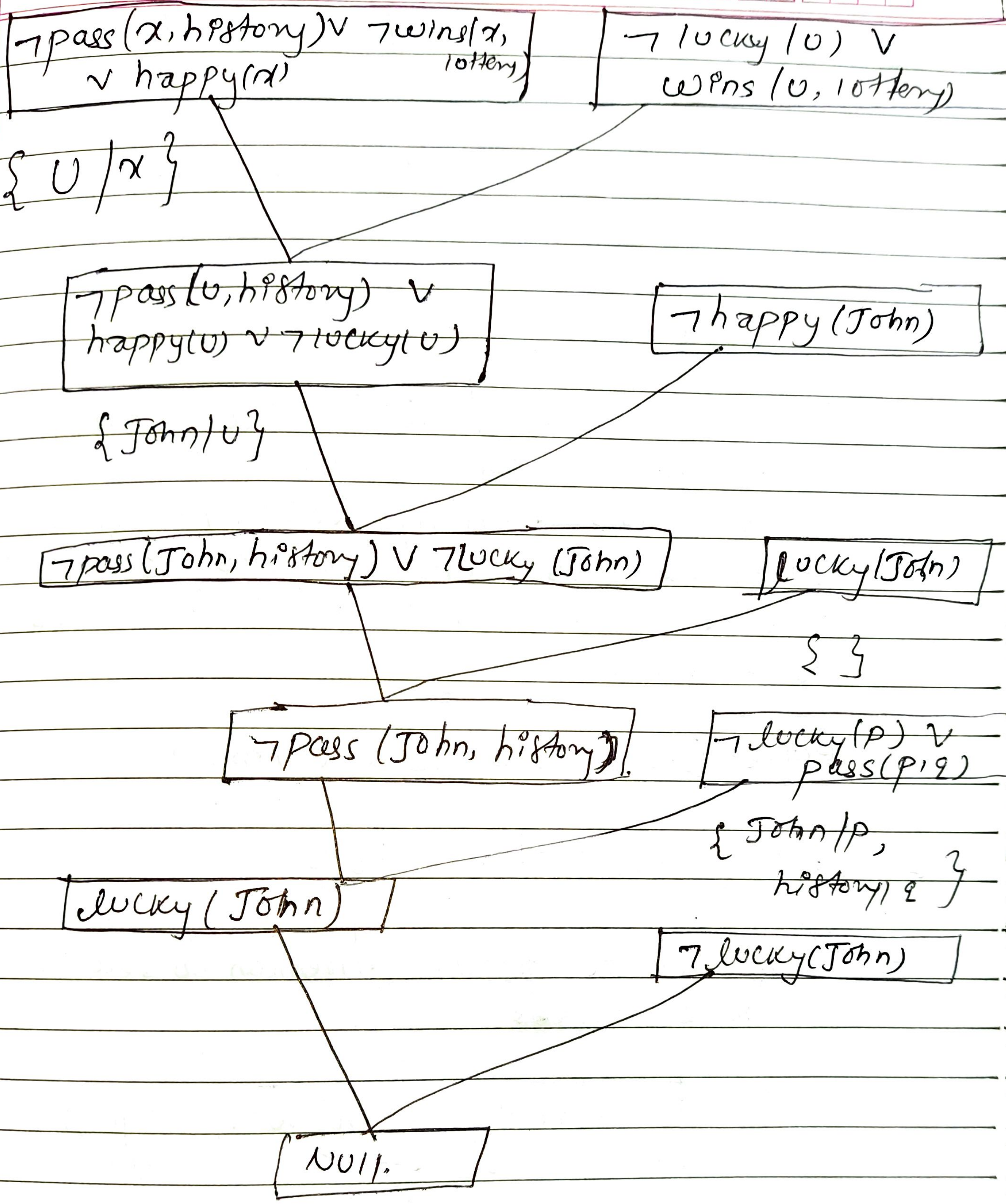
$\neg lucky(u) \vee wins(u, lottery)$

[Goal: $happy(John)$
Negation: $\neg happy(John)$] ~~no need to write goal~~ in every step. Just write its negation.

It is now in CNF

Step3: Draw resolution tree.

Note: you can start with negation of goal too. The results will be same.



Hence, by contradiction This proves that John is happy.

fig: Resolution tree

DATE

* Forward chaining and Backward Chaining in Predicate Logic (PL) or (FOPL)

Language Same as FC & BC in propositional Co...

a) Forward Chaining

Ex: The law says that it is a crime for an American to sell weapons to hostile nations. The country NONO, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.

Prove:- Colonel West is a criminal

⇒ Let's convert it into FOPL

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

~~Vx Vy V~~ $\boxed{American(x) \land Weapon(y) \land Sells(x,y,z) \land Hostile(z) \rightarrow Criminal(x)}$

↳ The Country NONO, an enemy of America, has some missiles.

* Country NONO has some missile

$\exists x \; Owns(Nono, x) \land Missile(x)$

Using Existential instantiation

$\boxed{Owns(NONO, M_1) \text{ and } Missile(M_1)}$

classmate

This can both be written independently
Owns(NONO, M_1)
Missile(M_1)

PAGE

DATE

* Country Nono, an enemy of America.

$\boxed{Enemy(Nono, America)}$

* All of its missiles were sold to it by Colonel West

$\forall x\ \boxed{Missile(x) \wedge Owns(Nono, x) \rightarrow Sells(West, x, Nono)}$

* Missiles are weapons.

$\boxed{Missile(x) \rightarrow Weapon(x)}$

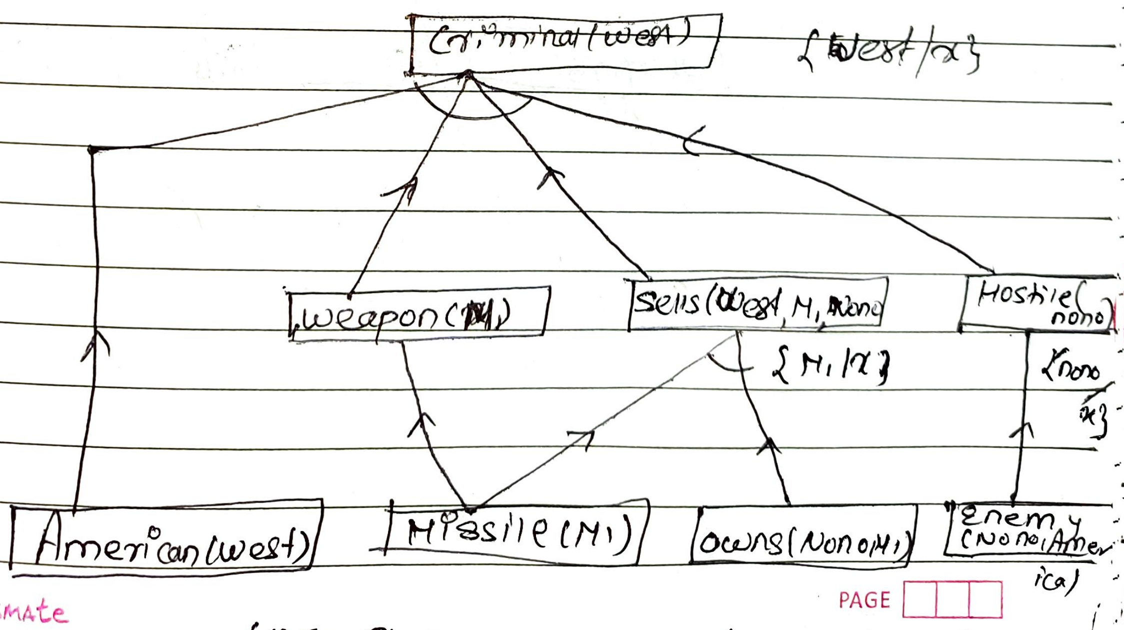
* An enemy of America counts as "hostile"

$\boxed{Enemy(x, America) \rightarrow Hostile(x)}$

↳ West, who is American

$\boxed{American(West)}$

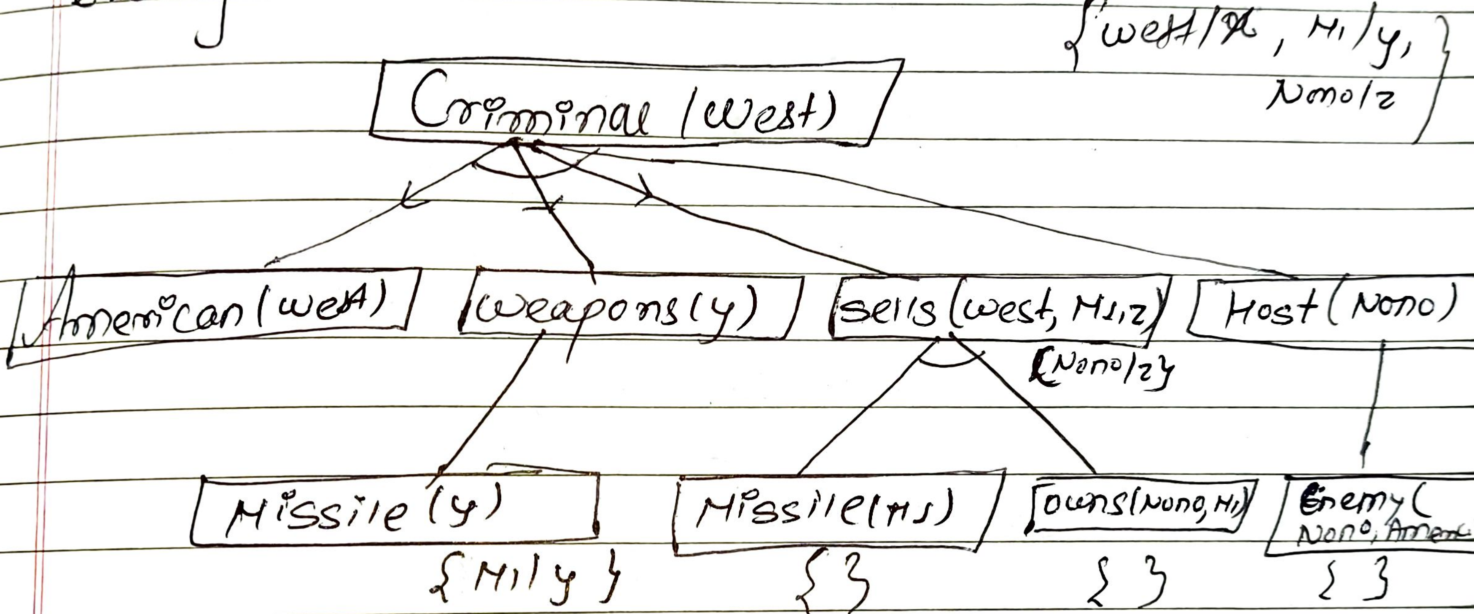
Now, Let's draw tree.



(Note: It is not necessary to give arrow) but just for understanding)

b) Backward Chaining.

Let's use backward Chaining in the same example.



DATE

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 stat it is 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 & that decision is called rational decision. Ex: automated taxi.

Let us take another example of uncertain knowledge:-

Toothache → Cavity.

The rule itself is wrong in real scenario. Not all patients with toothpaste 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 we have to add almost unlimited list of possible problems.

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

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

DATE

Random variables

The variables whose values are the outcomes of the random experiments are called random variables. There are two types of random variables.

- Discrete : They are the random variables, whose range is a countable set. A countable set can be either a finite set or a countably infinite set. Ex: X is a discrete variable & its range is a finite set ({0,1,2})

- Continuous :- They have range in the forms of some interval, bounded or unbounded, of the real line. Ex: Y is the random variable that is equal to the height of different people in a given population set.

Probability & its types

A probability is a number that reflects the chance or likelihood that a particular event will occur. It ranges from 0 to 1 and it can be also expressed as percentages ranging from 0% to 100%.

Properties :

- Probability 0 indicates event is not occuring (i.e. there is no chance of event occuring) & 1 indicate event will occur. So, it ranges from 0 to 1.
- Probability of event A is generally written as P(A).
- If P(A) > P(B) then A has higher chance of occuring than B
- If P(A) = P(B), then A & B are equally likely to occur

There are two types of probability:-

1. Prior Probability

It states that, the probability of event A is the number of ways A can occur divided by the total numbers of possible outcomes.

$$P(A) = \frac{m}{n} = \frac{\text{no. of ways event } A \text{ can occur}}{\text{total no. of possible outcomes}}$$

2. Posterior or Conditional probability

It is the probability of one event occuring with some relationship to one or more other events.

Ex. $P(A/B) = \frac{P(A \cap B)}{P(B)}$, $P(B/A) = \frac{P(A \cap B)}{P(A)}$

Where $P(A|B)$ is the probability of A occuring given the probability of B.

* Inference using Full Joint Distribution

The process of generating knowledge by using joint probability is called inference using full joint distribution. Joint probability is a statistical measure that calculates the likelihood of two events occuring together & at the same point in time. It is the probability of event Y occuring at the same time that event X occurs.

DATE

Ex: 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

Q. In a group of 100 sports car buyers, 40 bought alarm systems, 30 purchased bucket seats, and 20 purchased an alarm system and bucket seats. If a car buyer chosen at random bought an alarm system, what is the probability they also bought bucket seats?

⇒ Let, alarm systems = A
bucket seats = B

Then, $P(A) = \frac{40}{100} = 0.4$

$P(B) = \frac{30}{100} = 0.3$

$P(A \cap B) = \frac{20}{100} = 0.2$

Now, probability that the buyer chose/bought bucket seats given that they bought alarm system,

$P(B/A) = \frac{P(A \cap B)}{P(A)} = \frac{0.2}{0.4} = 0.5$ i.e 50%

DATE

* Baye's rule and its uses (Baye's Theorem)

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. ex: your probability of getting a parking space is connected to the time of the day you park, and what conventions are going on at any time. Baye's theorem is slightly more detailed (nuanced)

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

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

Uses of Baye's rule.

- Widely used to provide probabilistic prediction in AI
- Used in weather forecasting
- In finance, it can be used to rate the risk of lending money to potential borrower
- Used to calculate the next step of the robot when the already executed step is given

DATE

Some examples based on baye's theorem

Q. A doctor is aware that disease meninglis causes a patient to have a stiff neck, and it occurs 80% of the time. He is also aware of some more facts.

- The known probability that a patient has meningitis disease is $1/30,000$
- The known probability that a patient has a stiff neck is 2%.

What is the probability that a patient has diseases ~~megn~~ meningitis with a stiff neck?

⇒ Let, S = proposition that patient has stiff neck

B = Proposition that patient has megntis.

Given,

$$p(M) = 1/30,000$$

$$p(S) = 2\% = 0.02$$

$$p(S/M) = 80\% = 0.8.$$

∴ Probability that patient has disease meningitis with a stiff neck is

$$P(M/S) = \frac{p(M) \cdot p(S/M)}{p(S)} = \frac{\frac{1}{30,000} \times 0.8}{0.02}$$

$$= 0.001333$$

$$\simeq \frac{1}{750}$$

Hence, we can assume that 1 patient out of 750 patient with stiff neck has meningitis disease.

DATE

Q. From a standard deck of playing cards, a single card is drawn. The probability that the card is king is $4/52$. then calculate posterior probability $P(\text{King}/\text{Face})$, which mean the drawn face is a king card

⇒ Given,

$p(\text{King}) = 4/52 = 1/13$

$p(\text{face}) = \frac{12}{52} = \frac{3}{13}$

$P(\text{face}/\text{king}) = 1$ (we know, probability that it is a face known that it is a king is 1 because all kings are faces)

Now, $P(\text{King}/\text{face}) = \frac{p(\text{King}) \times P(\text{face}/\text{King})}{p(\text{face})}$

$= \frac{1/13 \times 1}{3/13}$

$= 1/3$

Q. Given the following statistics, what is the probability that a women has cancer if she has a positive mammogram result?

DATE

- 1 % of women over 50 have breast cancer.
- 90 % of women who have breast cancer test positive on mammograms
- 8 % of women will have false positive

⇒ Let, C = event of breast cancer.
M = positive on mammogram
$\neg C$ = not having breast cancer
$\neg M$ = false positive on mammogram

Then,
$P(C) = 1\% = 0.01$
$P(\neg C) = 1 - 0.01 = 0.99$
$P(M/C) = 90\% = 0.9$

$P(M/\neg C) = 8\% = 0.08$

Now, $P(M) = P(M \cap C) + P(M \cap \bar{C})$

$\therefore P(M) = P\left(\frac{M}{C}\right) \cdot P(C) + P\left(\frac{M}{\neg C}\right) \cdot P(\neg C)$

$= (0.01 \times 0.9) + (0.08 \times 0.99)$
$= 0.0882$

because,
$P(A/B) = \frac{P(A \cap B)}{P(B)}$
$\therefore P(M/C) = \frac{P(M \cap C)}{P(C)}$
$P(M \cap C) = P(M/C) \cdot P(C)$

$\therefore$ Probability that a woman has cancer if she has a positive mammogram result is

$$P(C/M) = \frac{P(C) \cdot P(M/C)}{P(M)}$$

$$= \frac{0.01 \times 0.9}{0.0882}$$

$$= 0.10$$

Bayesian Networks

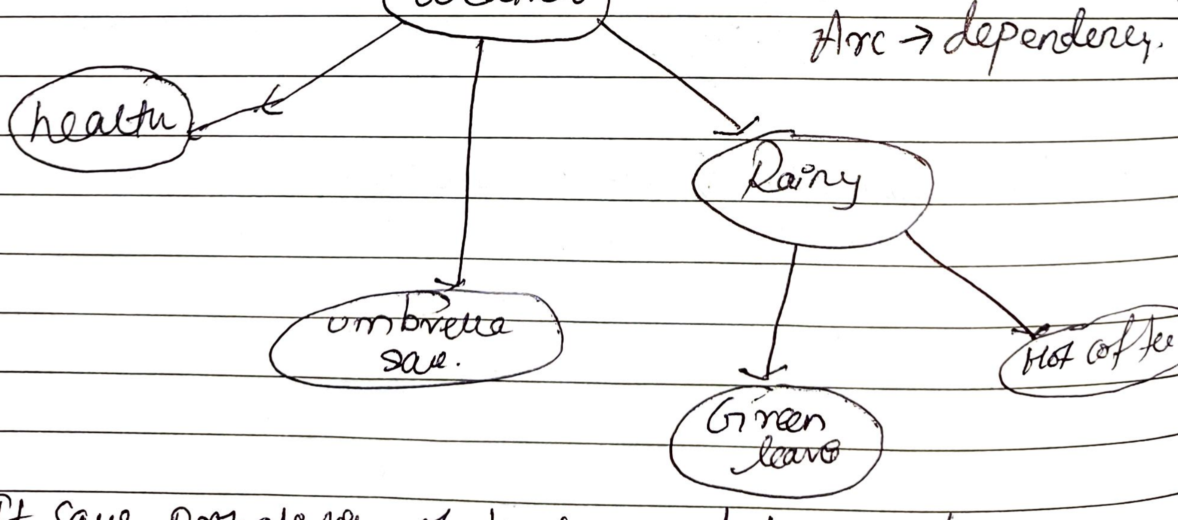
A Bayesian network (BN) is a probabilistic graphical model for representing knowledge about an uncertain domain where each node corresponds to a random variable and each edge represents the conditional probability for the corresponding random variable.

Due to dependencies and conditional probabilities a BN corresponds to a directed acyclic graph (DAG) where no loop or self connection is allowed. It is also called belief networks or probabilistic network or casual network or knowledge map.

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

- Where,
  Node → Random variable
  Arc → dependency.

Ex:

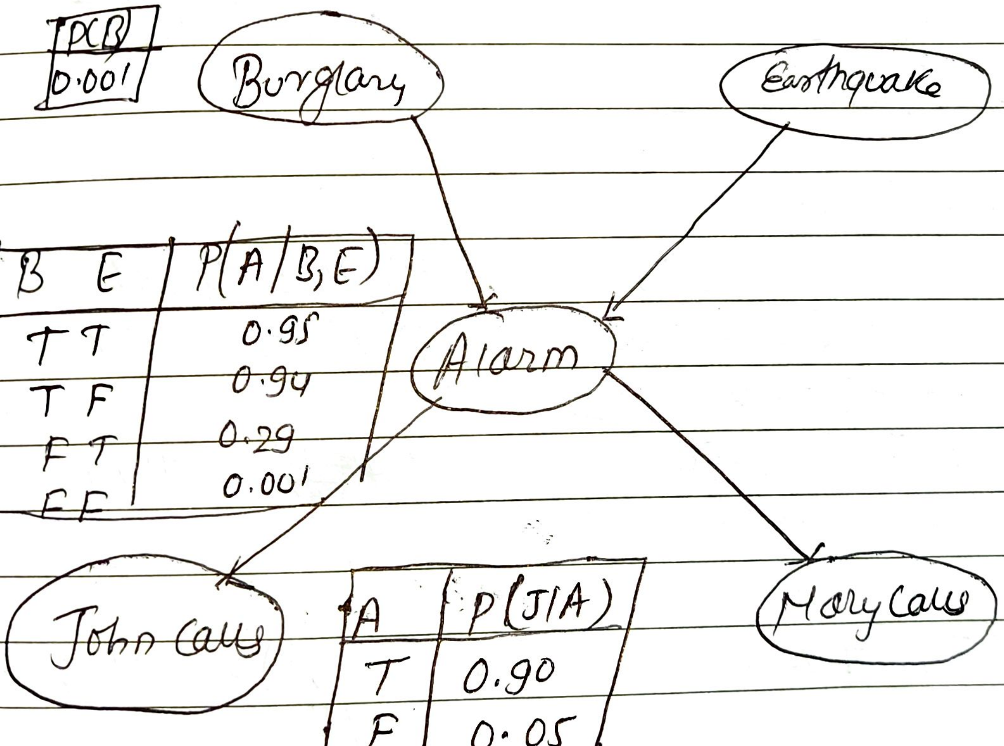


It says, probability of having good health depends on weather and so on..

DATE

Ex: You have a burglar alarm installed in your home. It is fairly reliable ~~that~~ at detecting a burglary, but also responds on ocassion to minor earthquakes. You also have two neighbors, John and Mary, who have promised to call you at work when they hear the alarm; John always calls but sometimes confuses the telephone ringing with the alarm and calls then, too. Mary, on the other hand, likes rather loud music and sometimes misses the alarm altogether.

. We would like to estimate the probability of a burglary with given evidence who has or has not call



P(B)
0.001

P(E)
0.002

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

A	P(J/A)
T	0.90
F	0.05

A	P(M/A)
T	0.70
F	0.01

DATE

* Inference with Bayesian (Belief) Network

Firstly 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 factorial form of the joint distribution. So, we simply evaluate the product using the provided conditional probabilities.

Continue from previous example →

Q. What is the probability that the alarm has sounded, but neither burglary nor an earthquake has occurred, and both John and Mary call?

⇒ From table,

$P(B) = 0.01$ ∴ $P(\neg B) = 0.999$

$P(E) = 0.002$ ∴ $P(\neg E) = 0.998$

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

$P(J/A) = 0.90$ ∴ ~~P(J~~ $P(J/\neg A) = 0.05$

$P(M/A) = 0.70$

∴ $P(J \wedge M \wedge A \wedge \neg B \wedge \neg E) = P(J/A) \cdot P(M/A) \cdot P(A/\neg B, \neg E) \cdot P(\neg B) \cdot P(\neg E)$

$= 0.90 \times 0.70 \times 0.001 \times 0.99 \times 0.998$

$= 0.00062$

DATE

Q. What is the probability that John call?

:) $P(J) = P(J/A) \cdot P(A) + P(J/\neg A) \cdot P(\neg A)$ [ie. $P(J) = P(J \cap A) + P(J \cap \bar{A})$]

(It is because we know, John calls when there is alarm ~~and~~ or when alarm is not ringing (ie. telephone call)

$$= P(J/A) \left\{ \begin{array}{l} P(A/B,E) * P(B,E) + \\ P(A/\neg B,E) * P(\neg B,E) + \\ P(A/B,\neg E) * P(B,\neg E) + \\ P(A/\neg B,\neg E) * P(\neg B,\neg E) \end{array} \right\} +$$

$$P(J/\neg A) \left\{ \begin{array}{l} P(\neg A/B,E) * P(B,E) + \\ P(\neg A/\neg B,E) * P(\neg B,E) + \\ P(\neg A/B,\neg E) * P(B,\neg E) + \\ P(\neg A/\neg B,\neg E) * P(\neg B,\neg E) \end{array} \right\}$$

Since B&E are independent

$$= 0.90 \left\{ (0.95 \times \underbrace{0.001 \times 0.002}) + (0.29 \times 0.999 \times 0.002) + (0.94 \times 0.001 \times 0.998) + (0.001 \times 0.999 \times 0.998) \right\}$$

$$+ 0.05 \left\{ ((1-0.95) \times 0.001 \times 0.002) + ((1-0.29) \times 0.999 \times 0.002) + ((1-0.94) \times 0.001 \times 0.998) + ((1-0.001) \times 0.999 \times 0.998) \right\}$$

$$= 0.90 \times 0.00252 + 0.05 \times 0.9974$$

$$= 0.0521$$

* Fuzzy logic

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

Take a look at the following diagram. It shows that 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 falseness. The number which indicates the value in fuzzy systems is called truth value.

Is Ankit Honest?
- True / Yes / 1
- False / No / 0

} Boolean logic (Crisp set)

Is Ankit Honest?
- Extremely Honest (1.0)
- Very honest (0.85)
- Sometimes honest (0.35)
- Extremely dishonest (0.0)

} Fuzzy Logic

* Fuzzy Sets

A classical 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 doesn't make any changes in the set

Fuzzy sets can be considered as an extension and gross over simplification of classical sets. Classical sets contain elements that satisfy precise properties 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, ie. set membership may be partial.

Ex: If cold is a fuzzy set, exact temperature 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 (Extremely cold)

* Membership in fuzzy set (degree of belongingness)

It provides a measure of degree of belongingness of an element to a fuzzy set.

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

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

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

Formal definition of fuzzy set:-

A fuzzy set F on a given universe of disclosure U is defined as a collection of ordered pairs $(X, \mu_F(X))$ where $X \in U$ and for all $X \in U$, $0 \le \mu_F(X) \le 1$

$$F = \{ (X, \mu_F(X)) \mid X \in U, 0 \le \mu_F(X) \le 1 \}$$

DATE

Example of fuzzy set:

Let $X = \{g_1, g_2, g_3, g_4, g_5\}$ be the reference set of students

Let A be the fuzzy set of "smart" students, where "smart" is fuzzy term.

$$A = \{(g_1, 0.4), (g_2, 0.5), (g_3, 1), (g_4, 0.9), (g_5, 0.1)\}$$

Here, A indicates that smartness of g_1 is 0.4 & so on.

* Fuzzy Set operations

- Union
- Intersection
- Complement

Suppose A & B over X then,

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

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

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

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

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

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

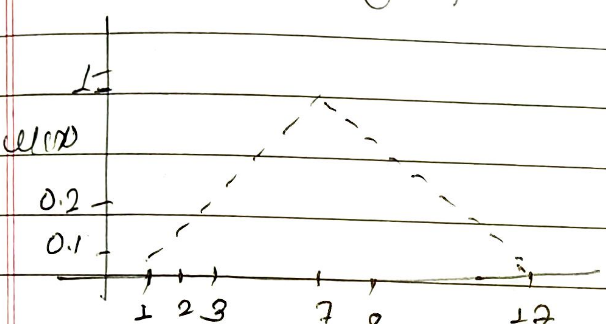
$= \left\{ \frac{0.9}{1} + \frac{0.6}{2} + \frac{1}{4} + \frac{0.2}{6} + \frac{0.9}{8} \right\}$

$$A \cap B = \left\{ \frac{0.5}{1} + \frac{0}{2} + \frac{0.5}{4} + \frac{0}{6} + \frac{0}{8} \right\}$$

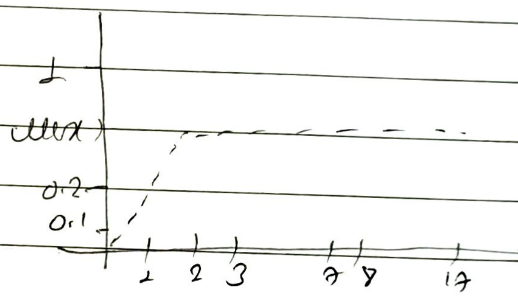
$$\mu_{\bar{A}} = \left\{ \frac{1-0.5}{1} + \frac{1-0.2}{6} + \frac{1-0.9}{8} + \frac{1-1}{4} \right\}$$

$$= \left\{ \frac{0.5}{1} + \frac{0.8}{6} + \frac{0.1}{8} + \frac{0}{4} \right\}$$

* Membership graph of fuzzy set

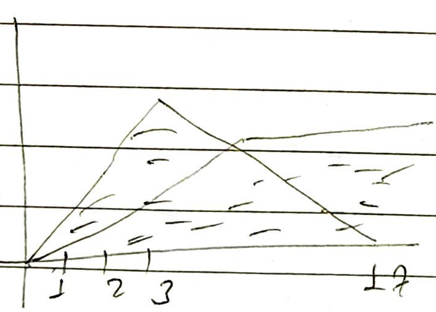


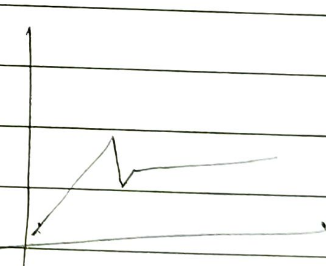
Membership of A



Membership of B

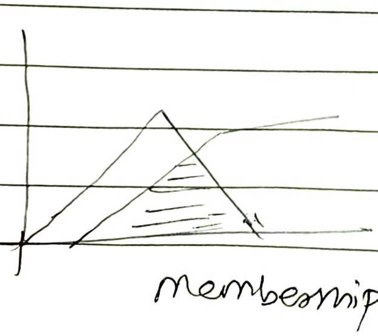
AUB

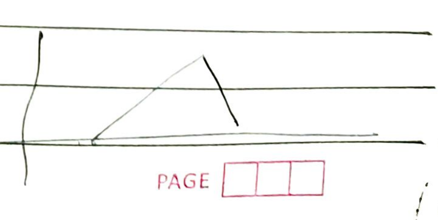




membership of AUB

AnB





* Fuzzification

It is the process of translating the measured numerical values into fuzzy linguistic values. In other words, it is where membership functions are applied, & the degree of membership is determined.

Two methods:

i) Support Fuzzification (S-fuzzification)

Here, the fuzzified set can be expressed with the help of the following relation.

$$\tilde{A} = \mu_1 Q(x_1) + \mu_2 Q(x_2) + \dots + \mu_n Q(x_n)$$

where, $Q(x_i) \rightarrow$ Kernel of fuzzification

This method is implemented by keeping μ_i constant & 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

It is the process of producing a quantifiable result in fuzzy logic. The fuzzy inference will output a fuzzy result, described in terms of membership of the fuzzy sets. Defuzzification interprets the membership degrees in the fuzzy sets into a specific action or real-value.

Different method

→ Max-Membership method

→ Centroid method

→ Weighted average method

→ Mean-Max Membership

DATE

Advantages of Fuzzy rule based / logic System

- Works with any types of inputs whether it is imprecise, distorted or noisy input information.
- Construction is easy & understandable
- Provides efficient solution to complex problems
- Algorithms can be described with little data, so little memory is required.
- Fuzzy logic comes with mathematical concepts of set theory & its reasoning is quite simple.

Disadvantages of Fuzzy logic Systems

- Suffers with ambiguity problem. Many researchers proposed different ways to solve a problem through fuzzy logic so there is no systematic approach.
- Proof of its Characteristics is difficult or impossible in most cases as we don't get mathematical description of our approach.
- Works on all types of inputs so accuracy is Compromised
  precise & imprecise.

Application of Fuzzy logic System

- Used in aerospace field for altitude control of spacecraft.
- Used in automotive system for speed & traffic control.
- Used for decision making support systems.
- Used in Chemical industry for PH controlling.
- Used in Natural language processing & Neural networks.

Questions asked from this Chapter

Q. Consider the following facts (2078 - 10 marks)

Every traffic chases some driver. Every driver who horns is smart. No traffic catches any smart driver. Any traffic who chases some driver but does not catch him is frustrated.

Now configure FOPL knowledge base for above statements. Use resolution algorithm to draw a conclusion that "If all drivers horn, then all traffics are frustrated"

Q. Construct semantic network for following. (2078 - 5 marks)

Ram is a person. Persons are humans. All humans have nose. Human are instances of mammals. Ram has a weight of 60 kg. Weight of Ram is less than the weight of Sita.

Q. How resolution algorithm is used in FOPL to draw conclusion? (2076 - 10 marks)

consider the facts:

→ "Anyone whom pugu loves is a star. Any hero who doesn't reherse do not act. Anmol is a hero. Any hero who doesn't work doesn't reherse. Anyone who doesn't act is not a star. Convert above to FOPL & use resolution to infer that "If Anmol doesn't work, then pugu does not love Anmol."

Q. What is script? How knowledge is represented in script? Illustrate its components with example. (2076 - 5 marks)

DATE

Q. Define frame. How knowledge is encoded in a frame? Justify with an example. (2076 - 5 marks)

Q. What do you mean by membership of an element in a fuzzy set? Given a domain of discourse $X = \{10, 20, 30, 40, 50, 60, 70\}$, construct a fuzzy set from X. Use your own assumptions for defining membership. (2076 - 5 marks)

Q. What is posterior probability? Consider a scenario that a patient have liver disease is 15% probability. A test says that 5% of patients are alcoholic. Among those patients diagnosed with liver disease, 7% are alcoholic. Now compute the chance of having liver disease, if the patient is alcoholic. (2078 - 5 marks)

Q. How facts in uncertain knowledge are represented? Configure a Bayesian network for following.

The probability of having rain is 60%. The chances of getting cold if it will rain is 80%. The probability of not having sunshine is 90%. The probability that it will be hot if it is sunshine is 0.67.

(2076 - old - 5 marks)

Q. How uncertain knowledge is represented. Given a following full joint probability distribution find the probability that a CD cover has length of 130 mm given width is 25 mm

y = width	x = length 129	130	131
25	0.12	0.42	0.06
16	0.08	0.28	0.04

(2076 - 5 marks)

classmate

PAGE

Q. What is Baye's theorem? Explain its application. (2075- 5 marks

Q. Why disjunctive normal form is required? Explain all steps with example. (2067- 5 marks) (2069- 5 marks)

Q. How do you convert to conjunctive normal form? Explain all the steps with example. (2072 - 5 marks

Q. Differentiate between inference & reasoning. Why probabilistic reasoning is important in AI? Explain with example. (2067 - 5 marks) (2073 - 5 marks)

Q. Describe briefly the approaches of knowledge representation with example. (2071 - 5 marks)

Q. Define knowledge representation system. How knowledge is represented using semantic networks? Illustrate with example. (2074 - 5 marks)

Q. Consider the following production system characterized by (2075 - 5 marks)

- Initial Short term: C5, C1, C3
- Production rules: C1 & C2 → C4

  C3 → C2

  C1 & C3 → C6

  C4 → C6

  C5 → C1

Show a possible sequence of two recognize- act cycle. Which will be the new content of the short - term memory after these two cycles?