6) Discuss various problem characteristics of AI.

Ans:- Before start modelling the search and trying to find solution for the problem, one must analysis it along several key characteristics intially. some of these are mentioned below.

Types of problem:-

There are 3 types of problems in real life: Ignorable, Recoverable and Irrecoverable

Ignorable:

These are the problems where we ignore the solution step for example, in proving a theorem, if some lemma is proved to prove a theorem, if some lemma is and later on we realize that it is not useful, then we can ignore this solution step

Recoverable:-

These are the problems where solution steps can be undone. for example, in water jug problem, if we have filled up the jug, we can empty it also

Irrecoverable:-

The problems where solution steps cannot be undone for example; any two players game such as chess, playing cards such problems can be solved by planning

Decomposition of a problem:-

Divide the problem into a set of independent smaller sub-problems, solve them and combine the solutions to get the final solution.

Consistency of knowledge Base used in solving problem:-

Make sure that knowledge could be in the form of solving problem is consistent. Inconsistent knowledge base will lead to wrong solutions. for example, if we have knowledge in the form of rules and facts as follows:

If it is humid, it will rain. If it is sunny, then it is daytime. It is sonny day.

Requirement of Solution:-

we should analyze the problem whether solution required is absolute or an relative. we can call solution to be absolute if we have to find exact solution. where as, it is relative. if we have reasonably good, approx solution

Role of knowledge:- It plays an Imp role in solving prob knowledge could be in any form of rules which help generating search space for finding the solution

1.10 STATE SPACE SEARCH (Introduction, general problem solving)

State space search and problem solving can be explained with a water jug problem.

Suppose we have a water jug problem:

You are given two jugs , a 4- gallon one and a 3- gallon one .Neither has any measuring markers on it . There is a pump that can be used to fill the jugs with water how can you get exact 2 gallons of water into the 4- gallon jug?

The state space for this problem can be described as of ordered pairs of integers (x, y) , such that x=0,1,2,3 or 4 and y=0,1,2 or 3 , represents the number of gallons of water in the 4- gallon jug , and y represents the quantity of water in the 3- gallon jug . The start state is (0,0) . The goal state is (2,n) for any value of n (since the problem does not specify how many gallons need to be in the 3- gallon jug).The operators to be used to solve the problem can be described as shown in Fig bellow. They are represented as rules whose left side are matched against the current state and whose right sides describe the new state that results from applying the rule.

1- (x,y) → (4,y) fill the 4- gallon jug
If x<4

2- (x,y) → (x,3) fill the 3-gallon jug
If x<3

3- (x,y) → (x-d,y) pour some water out of the 4- gallon jug
If x>0

4- (x,y) → (x-d,y) pour some water out of the 3- gallon jug
If y>0

5-(x,y) → (0,y) empty the 4- gallon jug on the ground
If x>0

6-(x,y) → (x,0) empty the 3- gallon jug on the ground
If y>0

7- (x,y) → (4,y-(4-x)) **pour water from the 3- gallon jug into the 4-gallon jug until the 4-galoon jug is full**
If x+y>=4 and y>0

8- (x,y) → (x-(3-y),3)) **pour water from the 4- gallon jug into the 3-gallon jug until the 3-gallon jug is full**
If x+y>=3 and x>0

9- (x,y) → (x+y,0) **pour all the water from the 3 -gallon jug into the 3-gallon jug**
If x+y<=4 and y>0

10- (x,y) → (0,x+y) **pour all the water from the 4 -gallon jug into the 3-gallon jug**
If x+y<=3 and x>0

11- (0,2) → (2,0) **pour the 2-gallon from the 3 -gallon jug into the 4-gallon jug**

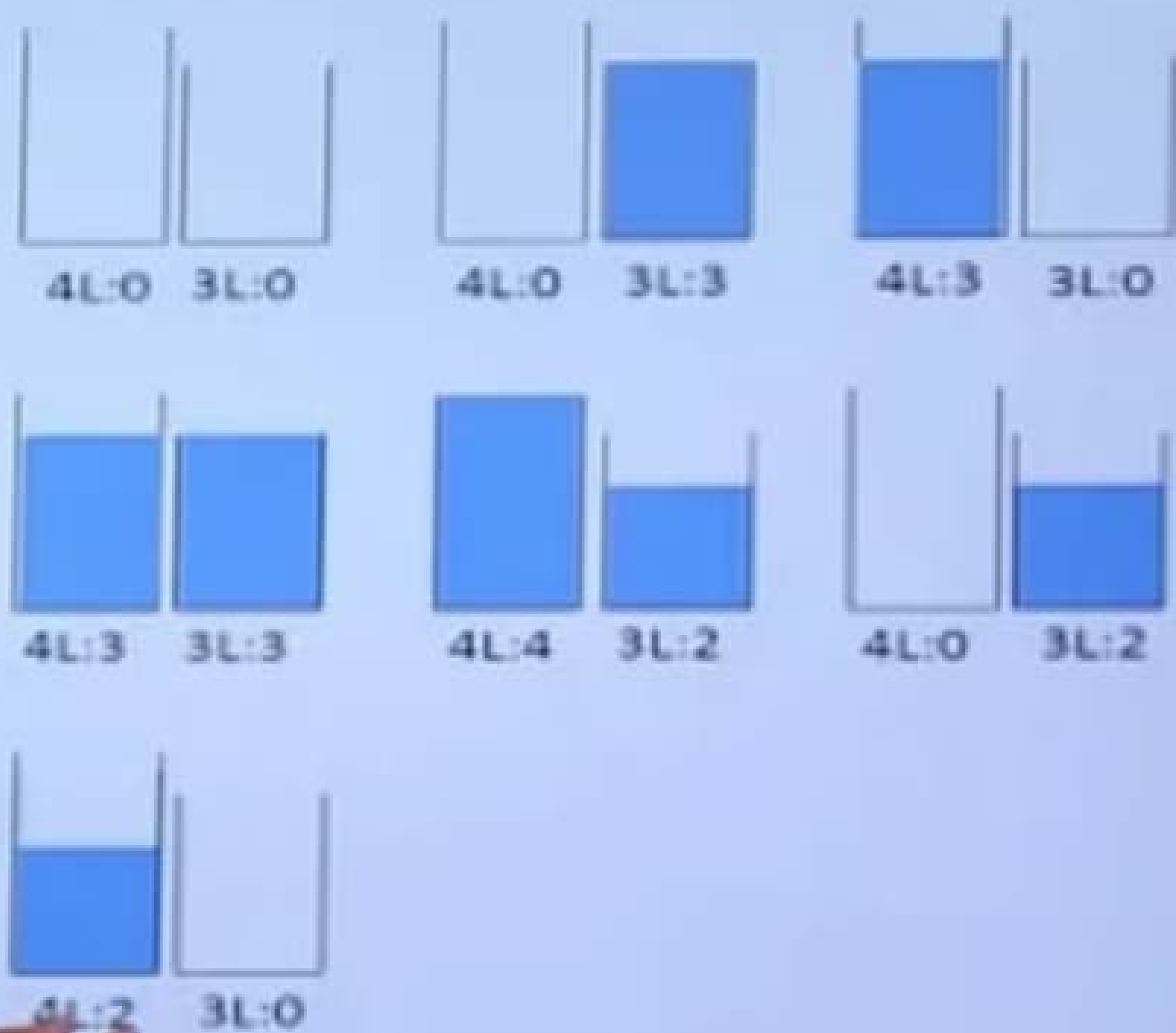
12- (2,y) → (0,x) **empty the 2 gallon in the 4 gallon on the ground**

Production for the water jug problem

Gallons in the 4- gallon Jug	Gallons in the 3- gallon	Rule Applied
0	0	
0	3	2
3	0	9
3	3	2
4	2	7
0	2	5 or 12
2	0	9 or 11

One solution to the water Jug problem.

One possible solution of the problem may be as follows –



demonstration...

1.8 CURRENT TRENDS IN AI

Here are 5 major AI trends .

1) The rise of AI-enabled chips

Unlike other software, AI heavily relies on specialized processors that complement the CPU. Even the fastest and most advanced CPU may not improve the speed of training an AI model. While inferencing, the model needs additional hardware to perform complex mathematical computations to speed up tasks such as object detection and facial recognition.

In 2019, chip manufacturers such as Intel, NVIDIA, AMD, ARM and Qualcomm will ship specialized chips that speed up the execution of AI-enabled applications. These chips will be optimized for specific use cases and scenarios related to computer vision, natural language processing and speech recognition. Next generation applications from the healthcare and automobile industries will rely on these chips for delivering intelligence to end-users.2019 will also be the year where hyperscale infrastructure companies like Amazon, Microsoft, Google, and Facebook will increase the investments in custom chips based on field programmable gate arrays (FPGA) and application specific integrated circuits (ASIC). These chips will be heavily optimized for running modern workloads based on AI and high-performance computing (HPC). Some of these chips will also assist next-generation databases to speed up query processing and predictive analytics.

2) Convergence of IoT and AI at the edge

In 2019, AI meets IoT at the edge computing layer. Most of the models trained in the public cloud will be deployed at the edge.

Industrial IoT is the top use case for artificial intelligence that can perform outlier detection, root cause analysis and predictive maintenance of the equipment.

Advanced ML models based on deep neural networks will be optimized to run at the edge. They will be capable of dealing with video frames, speech synthesis, time-series data and unstructured data generated by devices such as cameras, microphones, and other sensors.

IoT is all set to become the biggest driver of artificial intelligence in the enterprise. Edge devices will be equipped with the special AI chips based on FPGAs and ASICs.

3) Interoperability among neural networks becomes key

One of the critical challenges in developing neural network models lies in choosing the right framework. Data scientists and developers have to pick the right tool from a plethora of choices that include Caffe2, PyTorch, Apache MXNet, Microsoft Cognitive Toolkit, and TensorFlow. Once a model is trained and evaluated in a specific framework, it is tough to port the trained model to another framework.

The lack of interoperability among neural network toolkits is hampering the adoption of AI. To address this challenge, AWS, Facebook and Microsoft have collaborated to build Open Neural Network Exchange (ONNX), which makes it possible to reuse trained neural network models across multiple frameworks.

In 2019, ONNX will become an essential technology for the industry. From researchers to edge device manufacturers, all the key players of the ecosystem will rely on ONNX as the standard runtime for inferencing.

4) Automated machine learning will gain prominence

One trend that's going to change the face of ML-based solutions fundamentally is AutoML. It will empower business analysts and developers to evolve machine learning models that can address complex scenarios without going through the typical process of training ML models.

When dealing with an AutoML platform, business analysts stay focused on the business problem instead of getting lost in the process and workflow.

AutoML perfectly fits in between cognitive APIs and custom ML platforms. It delivers the right level of customization without forcing the developers to go through the elaborate workflow. Unlike cognitive APIs that are often considered as black boxes, AutoML exposes the same degree of flexibility but with custom data combined with portability.

5) AI will automate DevOps through AIOps

Modern applications and infrastructure are generating log data that is captured for indexing, searching, and analytics. The massive data sets obtained from the hardware, operating systems, server software and application software can be aggregated and correlated to find insights and patterns. When machine learning models are applied to these data sets, IT operations transform from being reactive to predictive.

1.3 INTELLIGENT SYSTEMS

An intelligent system is a machine with an embedded, Internet-connected computer that has the capacity to gather and analyze data and communicate with other systems. Other criteria for intelligent systems include the capacity to learn from experience, security, connectivity, the ability to adapt according to current data and the capacity for remote monitoring and management.

Intelligent Problem Solving

This section contains a very brief introduction to intelligent problem solving. A more detailed introduction is available in Moursund (2004).

The terms problem and intelligent problem solving are used throughout this document. We use these terms in a very broad sense, so that they include:

- posing, clarifying, and answering questions
- posing, clarifying, and solving problems
- posing, clarifying, and accomplishing tasks
- posing, clarifying, and making decisions
- using higher-order, critical, and wise thinking to do all of the above

Intelligent problem solving consists of moving from a given initial situation to a desired goal situation. That is, intelligent problem solving is the process of designing and carrying out a set of steps to reach a goal. Figure 1.1 graphically represents the concept of intelligent problem solving. Usually the term problem is used to refer to a situation where it is not immediately obvious how to reach the goal. The exact same situation can be a problem for one person and not a problem (perhaps just a simple activity or routine exercise) for another person.

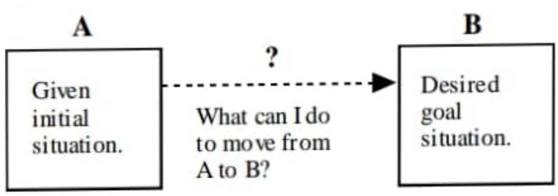


Fig 1.1 Problem Solving

Intelligent problem solving—how to achieve the final goal?

There is a substantial amount of research literature as well as many practitioner books on intelligent problem solving (Moursund, 2004). Here is a formal definition of the term problem. You—personally—have a formal, well-defined (clearly defined) problem if the following four conditions are satisfied:

1. You have a clearly defined given initial situation.
2. You have a clearly defined goal (a desired end situation). Some writers talk about having multiple goals in a problem. However, such a multiple goal situation can be broken down into a number of single goal problems.
3. You have a clearly defined set of resources—including your personal knowledge and skills— that may be applicable in helping you move from the given initial situation to the desired goal situation. There may be specified limitations on resources, such as rules, regulations, and guidelines for what you are allowed to do in attempting to solve a particular problem.
4. You have some ownership—you are committed to using some of your own resources, such as your knowledge, skills, and energies, to achieve the desired final goal.

2. Explain various components of AI.

Ans :- AI techniques must be independent of the problem domain as far as possible. Any AI program should have knowledge base, and navigational capability which contains control strategy and inference mechanism

knowledge base: AI programs should be learning in nature and update is knowledge accordingly, knowledge base generally consists of facts and rules and has the following characteristics.

→ It is voluminous in nature and requires proper structuring.
→ It may be incomplete and imprecise
→ It may be dynamic and keep on changing.

* control strategy:- It determines which rule to be applied. to know this rule, some heuristics or thumb rule based on problem domain may be used.

* Inference mechanism: It requires search through knowledge base and derives new knowledge using the existing knowledge with the help of inference rules

* Learning: Learning is distinguished into a number of different forms.

→ Reasoning
→ Problem-Solving
→ preception
→ Language understing

And some of the needin learning and inferencing: Processing, storage, Energy needs, and Networking.

→ And some more key components that are essential to AI. The following represent the core building blocks that are needed.

1.) AI Applications:- packaged applications that solve a business problem (i.e, virtual agents, financial planning).
2.) Data prep and cleansing:- Make your data ready for AI
3.) Model, Build, train and Run:- The studio of a data science artist to build, train and Run models (ML).
4.) Consumer features:- speech, images and vision, primarily user in consumer use cases
5.) Natural Language processing:- The nervouse system of enterprise AI
6.) Lifecycle Management:- Managing the Lifecycle of AI models and understanding how they perform

* Database:- The machines can be made more intelligent and introsepective relations only by incorporating them with database and feed them form of rules to process
* Hardware:- Graphic processing units aka GPUs are an important part of AI hardware to extend from one approach to a physical entity using the technology.
* framework:- for better ML processing a sound framework is required. The most commonly used are Python & R
* API's are required to publish AI services.

5. Discuss the procedure to solve

SEND + MORE = MONEY

Ans:- Cryptarithmetic problem is a type of constraint satisfication problem where the game is about digits and its unique replacement either with alphabets on other symbols.

In Cryptarithmetic problem, the digits (0-9) get substitutes by some possible alphabets or symbols.

The task in crytarithmetic problem is to substitute each digit with an alphabet to get the result arithmetically correct.

Rules on or constraints on a crytarithmetic problem:

→ There should be unique digit to be replaced with a unique alphabet.

→ The result should satisfy the predefined arithmetic rules, eg:- 2+2=4 nothing else

→ Digits should be from 0 to 9 only

→ There should be only on carry forward, while performing the addition operation on a problem.

→ The problem can be solved from both sides left hadnd side (L.H.S) or (R.H.S).

→ ~~from~~ let us solve

$$\begin{array}{r} c_4\,c_3\,c_2\,c_1\;\; \\ S\,E\,N\,D \\ +\;M\,O\,R\,E \\ \hline M\,O\,N\,E\,Y \end{array}$$

steps

1) Starting from L.H.S, the terms we know that M is a carry the carry can be 0 (or) 1 but Here M cannot be 0 as it is 1st digit

$\therefore \boxed{M=1}$

2) then in the second stage, we know that

$C_3 + S + M > 9$ ($\because$ it produced carry)

$\Rightarrow C_3 + S > 8$

if $C_3 = 0 \Rightarrow S > 8 \Rightarrow S = 9$ (Assumption)

At this moment

$C_3 + S + M = O + 10$ ($\because$ C_4 is 1)

$0 + 9 + 1 = O + 10$

$\Rightarrow O = 0$

$\therefore C_3 = 0$ $\boxed{S=9}$, $\boxed{O=0}$

3) In the third step, we know

$C_2 + E + O \le 9$ ($\because C_3 = 0$)

$C_2 + E + O = N$

$C_2 + E = N$

if $C_2 = 0 \Rightarrow E = N$ which is impossible

$\therefore C_2 = 1$, $\boxed{N = E + 1}$

4) ~~C~~ In the 4th stage,

$C_1 + N + R = E$ & $C_1 + N + R = E + 10$ ($\because C_2 = 1$)

$C_1 + R = 9$

:

if $C_1 = 0$ $R = 9$ ($\because S = 9$ this is not possible)

$\therefore C_1 = 1$ $\boxed{R = 8}$

5) In the 5th stage,

$D + E = Y + 10$ ($\because C_1 = 1$ we are taking $Y + 10$)

Let us assume, $y = 2$.

if $Y = 2$

$D + E = 12$

Let us assume D, E values

possible values to get 12 are

3 9 — x
9 3 — x ($\because S = 9$ it cannot be assigned)
7 5
5 7 } These are possible.

if $D = 5$, $E = 7$

$N = E + 1 \Rightarrow N = 8$ x (this is is impossible $\because R = 8$ we cannot have same number)

$\therefore D = \cancel{8}7$, $E = 5$ & $N = 6$

$\therefore$ The obtained values are $M = 1$, $S = 9$, $O = 0$, $R = 8$, $Y = 2$, $D = 7$, $E = 5$, $N = 6$

SEND	9567
+ MORE	1085
MONEY.	10652

$\therefore$ We are obtained with all the values.

(A·S)

Axiomatic System (version2) :- It allows only one symbol i.e, $\Rightarrow$ (implication). It do not allow $\vee, \wedge, \Leftrightarrow$

* output of inference system is True|False
* only modos pollen rule is allowed in Axiomatic System
* Natural deduction system is a system which do not have any rejections all symbols are accepted
* Rules: ① Natural deduction System → Axiomatic System

② version2

* Rules

$A \vee B = \sim A \rightarrow B$

$A \wedge B = \sim(A \rightarrow \sim B)$

$A \Leftrightarrow B = (A \rightarrow B) \wedge (B \rightarrow A) / \sim((A \rightarrow B) \rightarrow \sim(B \rightarrow A))$

* Rules :- when α, β, γ are true then below are true

1.) $\alpha \rightarrow (\beta \rightarrow \alpha)$

2.) $(\alpha \rightarrow (\beta \rightarrow \gamma)) \longrightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))$

3.) $(\sim\alpha \rightarrow \sim\beta) \rightarrow (\beta \rightarrow \alpha)$

The only inference rule that can be applied

$P \rightarrow Q, P \Rightarrow Q$ (Modus Pones)

Example:-1

* $A \to B$
* $B \to C$ prove $A \to C$

Ans 1) $(B \to C)$ $\qquad \alpha = (B \to C)$

2) $(B \to C) \to (A \to (B \to C))$ $\quad [\alpha \to (B \to \alpha)]$

3) Inference (1,2) $A \to (B \to C)$ $\quad [P, P \to Q \Rightarrow Q]$

$A \to (B \to C) \Rightarrow$

4) R-2, $((A \to B) \to (A \to C))$ $\quad [\alpha \to (B \to \gamma) \Rightarrow ((\alpha \to B) \to (\alpha \to$

5) Inference (3,4)

$((A \to B) \to (A \to C))$ $\quad [P, P \to Q \Rightarrow Q]$

6) $(A \to B)$

7) Inference (5,6) $A \to C$

Example:-2

* $\sim A$ prove $A \to \sim B$

1) $\sim A$

2) $\sim A \to (B \to \sim A)$ $\quad [R\text{-}1 = \alpha \to (B \to \alpha)]$

3) from (1,2) $(B \to \sim A)$

4) R-3 $(\sim \alpha \to \sim B) \to (B \to \alpha)$

$(B \to \sim A) \to (A \to \sim B)$

5) combine (3,4) $A \to \sim B$

Example:- 3

* $\sim A$ prove $A \to B$

Ans 1) $\sim A$

2) $\sim A \to (\sim B \to \sim A)$ [R-1 B can be anything]

3) $(\sim B \to \sim A)$ (1, 2)

4) R-3 $\Rightarrow$ $(\sim B \to \sim A) \to (A \to B)$

5) from (3, 4) $A \to B$

Ex :- 4

* $P \to Q$

$(P \vee Q) \to R$

$(R \wedge P)$ Prove Q

Rules of Inference (n)

$A \to B = \sim A \vee B$

$A \vee B = \sim A \to B$

$A \wedge B = \sim(A \to \sim B) = \sim(\sim A \vee \sim B)$

$A \Leftrightarrow B = (A \to B) \wedge (B \to A) / \sim((A \to B) \to \sim(B \to A))$

* $A \to B$

$B \to C$

$C \to D$ prove $A \to D$

Sol

1.) $B \to C$

2 $(B \to C)(A \to (B \to C))$ [∵ R1]

3. from 1,2 $(A \to (B - C))$

4.) $(A \to B) \to (A \to C)$ [R2]

5.) $A \to B$

6.) from 5,4 $A \to C$.

7.) $C \to D$

8.) $(C \to D)(A \to (C \to D))$ [R_1]

9.) from 7,8 $A \to (C \to D)$

10) $(A \to C) \to (A \to D)$ [R_3]

11.) from 6, 10 $A \to D$

Ex

* $\sim A$, $\sim B \to A$
  prove B

1. $\sim B \to A$
2. $(\sim B \to A) \to (\sim A \to B)$ [R.3]
3. from 1,2 $(\sim A \to B)$
4. from 1,3 B

* $\sim A$ prove B

1.)

* $P \to \sim Q$, $Q \vee R$ & $\sim S \to P$, $\sim R$ concolode 'S' using NDS

1.) $\sim R$
2.) $(Q \vee R)$
3.) $(Q \vee R \vee \sim R)$ from (1,2)
4.) Q from (3)
5.) $P \to \sim Q$
6.) $\sim P \vee Q$
7.) $\sim P$ from (4,6) $\sim P \vee Q \vee Q$
8.) $\sim S \to P$
9.) $S \vee \sim P$ from 8
10.) S from(7,9)

* $P \to (Q \to S)$, $\sim R \vee P$, Q, R proves S using NDS

1.) $\sim R \vee P$

2) R

3.) P from (1,2) $(R \wedge \sim R \vee P)$

4.) $P \to (Q \to S)$

5.) $Q \to S$ from (3,4) $P, P \to Q \Rightarrow Q$

6.) Q

7.) S from (6,5)

* $P \vee Q$, $Q \to R$, $P \to M$, $\sim M$ prove $R \wedge (P \vee Q)$ using NDS

1.) $P \to M$

2) $\sim P \vee M$

3) $\sim M$

4.) $\sim P$ from (2,3)

5) $P \vee Q$

6.) $\sim P \to Q$

7.) Q from (4,6)

8.) $Q \to R$ ~~Ro~~

9.) R from (7,8)

10.) $R \wedge (P \vee Q)$ from (9,5)

RESOLUTION

* you have a stmt in your innate language represent them in first order logic and arriving conclution
(f.OL)

Ex

a) Marcus was a man
b) Marcus was a pompeian
c) All Pompeian's are Roman
d) ~~All~~ Caesar was a Ruler
e) All romans were either loyal to ceasar or hated him
f) Everyone is loyal to someone
g) People only try to assasinate rulers they are not loyal to.
h) Marcus tried to assassinate ceasar

que

1) Was Marcus loyal to Caesar? (No)
2) Is Marcus a Roman? (Yes)
3) Is Caesar a Roman? (cannot be ans)

* Here we need convert (NLP) into f.O.L

a) Man (Marcus)
b) pompeian (Marcus)
c) $\forall(x)$: Pompeian (x) → Roman (x) [∵ if pompe, he is roman]
d) ruler (caesar)
$\forall(x)$: for all x
e) $\forall(x)$: Roman (x) → Loyal (x, caesar) ∨ Hate (x, caesar)
f) $\forall(x)$ $\exists(y)$: Loyal (x, y)

$\forall(x)\ \forall(y)$:

g) (People(x) and Ruler(y) ~~and~~ assascinate(x,y)) → ~Loyal(x,y)

h) assascinate(Marcus, Caesar)

~~⇒ was Marcus loyal to caesar~~

1) Did Marcus hates caesar?

AI Rules:-

1) No implication

$P \rightarrow Q \qquad \sim P \vee Q$

2.) Reduce the scope of ~

$\sim(P \vee Q) = \sim P$ and $\sim Q$

3.) Try to standerdize the variable

$\forall(x)\ P(x)\ \forall(x)\ R(x)$

$\forall(x)\ P(x)\ \forall(y)\ R(y)$

use diff variables

4.) Move all the quantifiers to L.H.S

$\underbrace{\forall(x)\ \forall(y)}\ P(x)\ R(y)$

5.) Eliminate Existential Quantifier

$E(x)$ ROMAN(x) write this as

ROMAN(x_1)

6.) Remove the Universal Quantifiers

7.) ~~$(a \vee b) \wedge c$~~ write this as $(a \vee c) \wedge (b \vee c)$

$(a \wedge b) \vee c$

8) seperate the clauses in 7

(avc)

(bvc)

→ Steps to convert FOL to clause form:-

* Apply the rules for the given information

1) All pompeian's were Romans.

$\forall x$: Pompeian(x) ⇒ Roman(x) [Rule-1] Eliminate ⇒

$\forall x$ ~Pompeian(x) ∨ Roman(x)

2) Reduce the scope of Negation

~(P∨Q) ⇒ ~P and ~Q

3) Standardize the variables

Evil(x) ⇒ Greedy(x) ∧ king(x)

Evil(x) ⇒ Greedy(y) ∧ king(z)

4) Move all the quantifier to the left

$\forall x$: Evil(x) ⇒ $\forall y$: Greedy(y) ∧ $\forall z$: king(z)

$\forall x \forall y \forall z$: Evil(x) ⇒ Greedy(y) ∧ king(z)

5) Eliminate Existential quantifiers

$\exists x$: Evil(x) ⇒ $\forall y$: Greedy(y)

$\forall y$: Evil(x_1) ⇒ Greedy(y)

6) Drop the universal prefixes

$\exists x$: Evil(x) ⇒ $\forall y$: Greedy(y)

Evil(x_1) ⇒ Greedy(y)

7.) ~~Confvert to~~ convert conjunction to disjunctions

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

8.) create a seperate clause corresponding to each conjuct

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

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

did
⇒ Marcus hates caesar?

Hate(Marcus, Caesar)

Ex: P, $(P \wedge Q) \rightarrow R$, ~~$(S \vee Q)$~~ $\rightarrow R$, T

~~$\sim P \vee \sim Q \vee R$~~

1.) P
2.) $\sim P \vee \sim Q \vee R$ (2)
3.) $\sim S \vee \sim Q$

Among given equations,

c) is not in clause form so converting it as

$\forall(x)$: $\sim$ Pompeian(x) $\vee$ Roman(x)

⇒ $\sim$ Pompeian (x) $\vee$ Roman(x)

e.) $\sim$ Roman(x) $\vee$ loyal(x, caesor) $\vee$ hate(x, caeasor)

f.) loyal(x, y_1)

g.) $\sim$ People(x) $\vee$ $\sim$ ruler(y) $\vee$ $\sim$ assassinate(x, y) $\vee$ $\sim$ loyal(x, y)

* $P, (P \wedge Q) \rightarrow R, (S \vee T) \rightarrow Q, T$ Prove R

Let us take $\sim R$ as fact (assume)

1.) P

2.) $\neg(P \wedge Q) \vee R$ Here we converted $\rightarrow$ into

3.) $\neg P \vee \neg Q \vee R$ clause form

4.) $\neg(S \vee T) \vee Q$

5.) $(\neg S \wedge \neg T) \vee Q$

6.) $(\neg S \vee Q) \wedge (\neg T \vee Q)$

7.) $(\neg S \vee Q)$

8.) $(\neg T \vee Q)$

9.) T

10.) $\sim R$

11.) $\neg P \vee \neg Q$ ~~from (1,3)~~ from (3,10)

12) $\neg Q$ from(1,11)

13.) $\neg T$ from (8,12)

14.) [ ] (9,13)

$\therefore$ we got empty set i.e, what you assumed is wrong so R is proved i.) Did Marcus Hate Caesar?

A) what you believe that is right take negation of that & prove. Since we feel the stmt is true we take.

~~1) ~ hate(Marcus, Caesar)~~

given information

1.) man(Marcus)

2.) ~~Pompeian(x) v Roman (x)~~ Pompeian (Marcus)

3.) Pompeian (x) v Roman(x)

4.) Ruler (Caesar)

5.) ~Roman(x) v loyal (x, caesar) v hate (x, Caesar)

6.) loyal(x, y1)

7.) ~ man(x) v ~ruler (y) v ~assasinate (x, y) v ~ loyal(x, y)

8.) assassinate(Marcus, Caesar)

Assume (from que)

9.) ~ hate (Marcus, Caesar)

10.) from 9 and 5 x = Marcus

~Roma(Marcus) v loyal(Marcus, Caesar)

11.) from 10 and 3

~pompeian (Marcus) v loyal(Marcus, caesar)

12.) from 11 and 2

loyal (Marcus, Caesar)

13.) from combining 12, 7 X = Marcus, Y = Caesar

$\sim$ man (x) $\vee$ $\sim$ ruler(y) $\vee$ $\sim$ assassinater (x,y)

14.) 1, 13

$\sim$ ruler (y) $\vee$ $\sim$ assasinate (x,y)

15.) 4, 14

$\sim$ assassinate(x,y)

16.) 15, 8.

NULL, EMPSET

This shows our assumption is wrong.

$\therefore$ hate(Marcos, Caesar) is true.

12.) Is Marcus loyal to Caesar?

Ans :- we know that Marcus is not loyal to Caesar

$\therefore$ Assume

9.) loyal (Marcos, Caesar)

10.) from 9, 7

$\sim$ man (x) $\vee$ $\sim$ ruler (y) $\vee$ $\sim$ assasinate (x,y)

11.) from 10, 1, 4, 8

Empty Set, Null Set

$\therefore$ our assumption is wrong.

$\therefore$ $\sim$ loyal (Marcos, Caesar)

iii) Is marcus a pompien?

Ans assume

9.) $\sim$ Pompiean (Marcus)

10.) from 9,2

Empty set

$\therefore$ Marcus is a Pompian

Inference Mechanisms

1.) Natural Deduction System (Universal Inf Mech)

2.) Axiomatic System ($\rightarrow$, P, $P \rightarrow G : G$)

3.) Resolution Mechanism (FOL, Cf, $\sim$H, Construct a inverted tree (Null set))

4.) Semantic Tableau

3 types in Inference Mech

1.) Process Symbols (NDS, AS)

2.) Statements and convert them into symbol (Resolution)

3) Semantic Based System

Ex

1.) John likes all kinds of food.

2.) Apple and chicken are food

3.) Anything anyone eats and does not killed is food.

4.) Bill eats peanuts and still alive

5.) Susan eats that Bill eats

Prove: John likes peanuts

Ans:-

1.) ~~like(John,~~ $\forall(x)$: $food(x) \rightarrow like(John, x)$

i) $\sim food(x) \vee like(John, x)$

2.) $food(Apple) \wedge food(Chicken)$

ii) $food(Apple)$

iii) $food(Chicken)$

4.) $Eats(Bill, Peanuts) \wedge alive(Bill)$

iv) $Eats(Bill, Peanuts)$

v) $alive(Bill)$

3.) $\forall(x)\ \forall(y)$: $Eat(x, y) \wedge alive(x) \rightarrow food(y)$

vi) $\sim Eat(x, y) \vee \sim alive(x) \vee food(y)$

5.) $\forall(x)$: ~~Bill~~ $Eat(Bill, x) \rightarrow Eat(Susan, x)$

vii) $\sim Eat(Bill, x) \vee Eat(Susan, x)$

we need to prove $like(John, peanuts)$

$\therefore$ we takes $\sim like(john, peanuts)$

viii) ~Like (John, Peanuts)

ix) combining viii, i we gen

~food(Peanuts)

x) combining ix, vi we get

~eat (x, y) V ~Alive (x)

xi) combining v, x we get

~eat (x, y)

xii) combining xi, iv we get

Empty Set, Null

∴ Like (John, Peanuts) ✓

Example:-2 (*** v.v. Imp)

1.) Marcus was a Man

2.) Marcus was a Pompeian

3.) Marcus was born in 40AD

4.) All men are mortal

5.) All pompeians died when the volcano erupted in 79 AD

6.) No mortal lives longer than 150 years.

7.) It is now = 1991

8.) Alives means not dead

9.) If someone dies, then he is dead at all the times later times died & dead are diff

1.) Man(Marcus)

2.) Pompeian(Marcus)

3.) Born(Marcus, 40AD)

4.) $\forall(x): man(x) \rightarrow mortal(x)$

$\sim man(x) \vee mortal(x)$

5.) $Erupted(Volcano, 79) \wedge \forall(x): Pompian(x) \rightarrow Died(x, 79)$

5.i) $Erupted(volcano, 79)$

5.ii) $Pompiaen(x) \rightarrow Dead(x, 79)$

$\sim Pompeian(x) \vee dead(x, 79)$

* 6.) $\forall(x): mortal(x) \wedge born(x, T_1) \wedge gt(T_2 - T_1, 150) \rightarrow dead(x, T_2)$

greater than func

$\sim mortal(x) \vee \sim born(x, T_1) \vee \sim gt(T_2 - T_1, 150) \vee dead(x, T_2)$

~~8) $\forall(x)$:~~

7.) now = 1991

* 8.) $\forall(x): Alive(x, T) \rightarrow \sim dead(x, T)$

$\sim Alive(x, T) \vee \sim dead(x, T)$

* 9.) $\forall(x): died(x, T_1) \wedge gt(T_2, T_1) \rightarrow dead(x, T_2)$

$\sim died(x, T_1) \vee \sim gt(T_2, T_1) \vee dead(x, T_2)$

* $$\begin{tikzcd}[column sep=small] & (A \wedge B) \to B \arrow[d] & \\ & \text{Tree} \arrow[dl, dash] \arrow[dr, dash] & \\ \text{Satisfyble} & & \text{UnSatisfyble} \end{tikzcd}$$

* This mainly depends on constructing trees

if we have $A \wedge B$ it is represented as

$$\begin{tikzcd} A \wedge B \arrow[d, dash] \\ A \arrow[d, dash] \\ B \end{tikzcd}$$

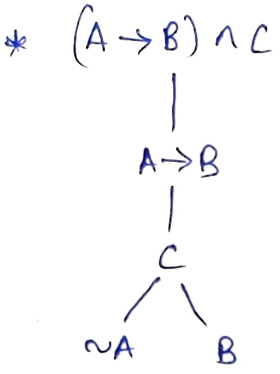
for $A \vee B$ representation is

$$\begin{tikzcd}[column sep=small] & A \vee B \arrow[dl, dash] \arrow[dr, dash] & \\ A & & B \end{tikzcd}$$

for $A \to B$, it is $\sim A \vee B$

$$\begin{tikzcd}[column sep=small] & A \to B \arrow[dl, dash] \arrow[dr, dash] & \\ \sim A & & B \end{tikzcd}$$

* This is not a inference mechanism. It takes an expression and constructs a tree and gives 2 outputs one is satisfyble & unsatisfyable



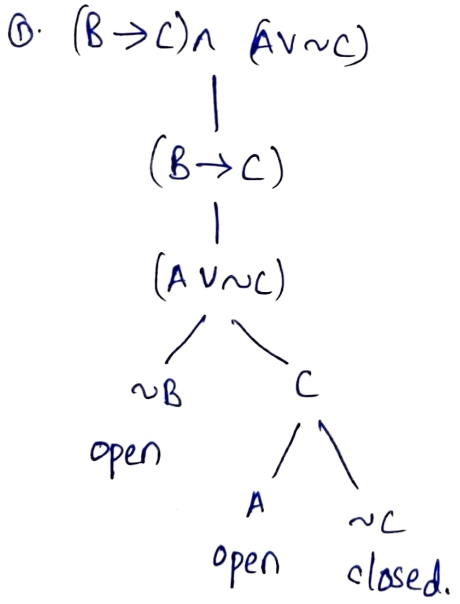
*) Satisfiable (At least one open) or Unsatisfiable (All closed)

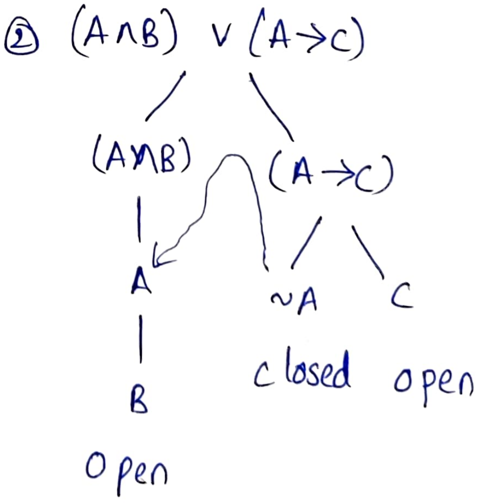
⇒ open path - if there is C no ~C exists & vice versa

⇒ closed path - if there is A and ~A exists

* If all the leaves ~~paths~~ are closed then it is unsatisfiable

* If we have at least one open path then it is satisfiable.





Satisfiable

Satisfiable

S

* $(A \wedge B) \wedge (B \to \sim A)$ find whether the expression is satisfyble or not. (Semantic tableau)

Sol:- $(A \wedge B) \wedge (B \to \sim A)$

$(A \wedge B)$

$(B \to \sim A)$

A

B

$\sim B$ (closed) $\qquad \sim A$ closed

2 leaves

* Since B & $\sim B$ exist and $\sim A$ & A exists both are closed

$\therefore$ all paths are closed so the output is unsatisfyble.

* $(B \to C) \wedge (A \vee \sim C)$

$B \to C$

$A \vee \sim C$

$\sim B$ open $\qquad C$

A open $\qquad \sim C$ closed

3 leaves are ther

* There is atleast one open path so the output is satisfyble

* $\sim(A \vee B) \wedge (C \to B) \wedge (A \to C)$

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

$(\sim A \vee B)$

$(C \to B)$

$(A \to C)$

$\sim A$

B

$\sim C$ — ~~ope~~ closed

B

$\sim A$ — open

C — closed

Satisfiable

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

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

$\sim A \vee \sim C$

$\sim B$

$\sim A$ — open

$\sim C$ — open

$A \to B$.

$\sim A$ — open

B — closed

4 leaves

Satisfiable.

* $\sim(A \vee B) \wedge (C \rightarrow B) \wedge (A \vee C)$

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

$(\sim A \vee B)$

$(C \rightarrow B)$

$(A \vee C)$

$\sim A$

$\sim B$

$\sim C$ $\quad$ B

closed.

A $\quad$ C

closed $\quad$ closed

The output is unsatisfiable