Artificial Intelligence

Contributed By:

Sankarsan Sahoo

Proff:- Sankarsan Sahoo

Airtificial Intelligence

~0~

Artificial Intelligence

Topic:

INTRODUCTION TO AI

Contributed By:

Sankarsan Sahoo

Intelligence :-

12/07/16

Data $\xrightarrow{\text{Process}}$ information

⇓ Process

Knowledge

⇓ process

Intelligence.

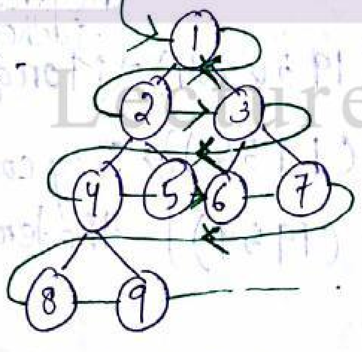
Database :- It is the collection of data and informat-ion.

Knowledgedatabase :- It is the collection of database and knowledge.

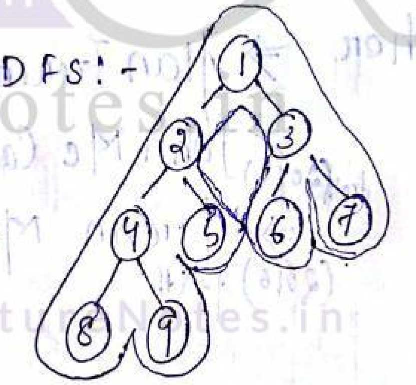
Searching :-

- → Linear
- → Binary
- → BFS
- → DFS

BFS :-



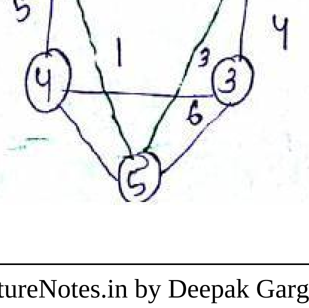
DFS :-



1, 2, 3, 4, 5, 6, 7, 8, 9
(priority queue)

1, 2, 4, 8, 9 5, 3, 6, 7.

→ Best First search :- → Hill climbing .



What is AI ?

AI is a branch of computer science which deals with the study and creation of computer systems that exibits some form of intelligence.

Application area :-

→ Game playing.

→ natural language processing.

→ Speech Recognition.

→ Reasoning.

→ Expert system.

→ Medical Diagnosis

→ simulation for drive and flight.

→ Heuristic problem solving.

History of AI :-

Father ⇒ Alan Turing. (1950-54) → (regarded as father of Morden compu

death (2012) ⇒ John Mc Carthy (1956) } → They coined/nam

(2016) death ⇒ Marvin Minsky (1959) } the term AI

Task classification of AI :- 14/07/2016

Basically 3 types of classification, they are

→ Mundane Task (common or ordinary

→ Formal Task

→ Expert task.

1. Mundane Task :-

a) perceptual (vision & speech)

b) Natural language processing (NLP)

(i) understanding.

(ii) generation.

(iii) Translation.

c) Commonsense Reasoning.

2. Formal tasks :-

a) Game playing.

b) Mathmatics

* Theorem proving.

* Problem solving.

3. Expert task :-

a) Enggineering. design

b) scientific analysis

c) Medi cal diagnosis

d) financial prediction.

15/07/2016

Problem : $\xrightarrow{\text{Solved}}$ Design Technique

→ linear / incremental

→ Divide-and-conquere

→ Dynamic prog.

→ Greedy technique

→ Branch and Bundary

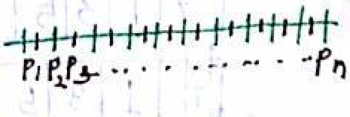
→ NP-complete.

⇒ Linear/incremental :-

•) linearsearch

•) insertion sort , Bubble sort.

⇒ Divide and conquer :-



consists 3 step

1- Divide

2 -Conquer (recursively solving) (B.S

complete.

•) It can be use to narrow the range of possibilities that must usually be considered.

what are different types of AI technique are available?

•) search.

•) use of knowledge.

•) Abstraction. (not in detail)

These 3 are basic AI-technique.

whatre the reasons to model human performance?

The level of the model

→ To test the psycological theory of performance.

→ To enable computers to understand human reasoning.

→ To enable people to understand computer reasoning.

→ To exploit what knowledge we can glean from people. (gather after harvest → glean)

Problem space & Search :-

Date :- 21/07/16

In order to solve a complex problem an AI technique has to follow following 4 steps.

i) Defining problem

ii) Generating alternative solⁿ

iii) Evaluation

iv) Applying the best solⁿ to solve the problem.

Statespace search :-

Q. What do u mean by statespace searching explain with an Example.

(i) many problems can be represented as a set of states also called state space and a set of rules; of how one state is transform to other

(ii) state space can be represented as a graph in

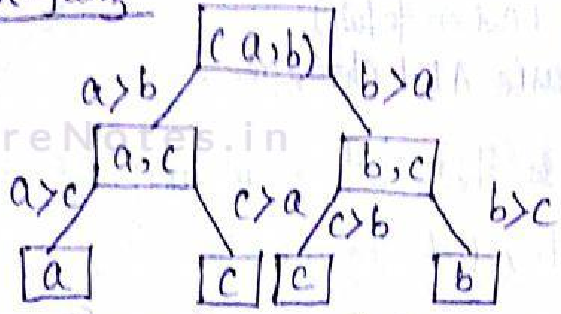
Scanned by CamScanner

Which nodes represents states and links represent actions to solve a problem

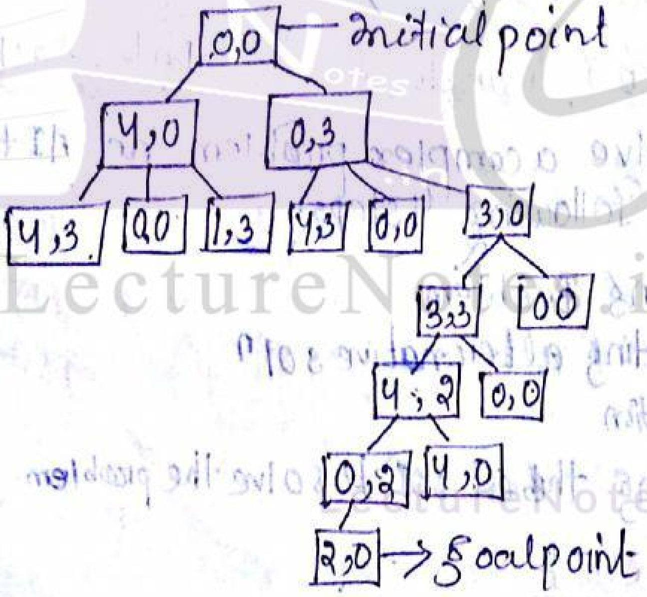
Ex :- water-Jug problem, tic-tac-toe
imp

Que :- Draw a state space representations to find the larges among 3 nos.

① water-jung

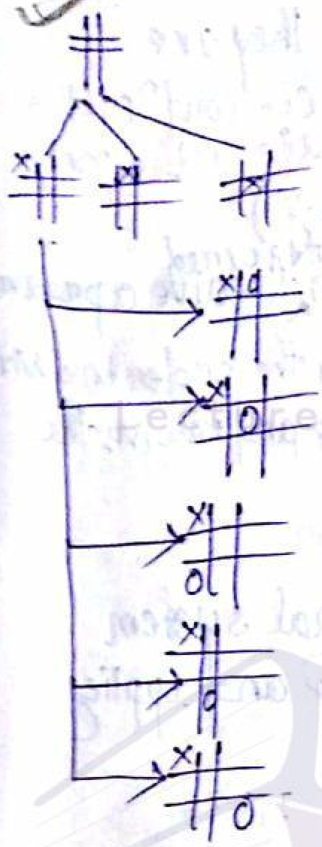


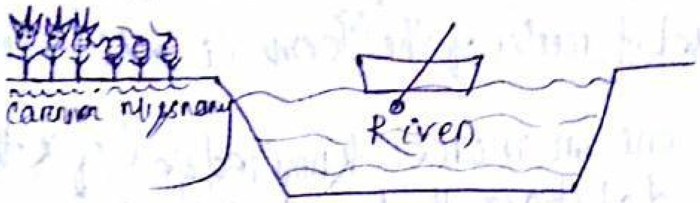
4lit
3lit
3lit



② ~~Tic-Tac-toe game~~

Missnary and carnivals





So state space searching solving a particular problem consist following 4 steps

i) Define a state space that contains all the possible states for solving a problem

ii) specify. one one more states from which the problem solving process may starts they are called initial states.

(iii) specify 1 or more states that would be acceptable as solution to the problem and called goal states

(iv) specify a set of rules that describes the actions available production rules.

Production system:-

It consists of 4 basic components they are

1) set of rules of the form $Ci \to Ai$ (where Ci = condⁿ part, ai = action part)

2) one or more knowledge (if x then y) database that contain information required to solve a probl

3) Control strategy (plan) - that determines the order in whi the rules are applied to the data base and resolve the conflicts if any.

4) A rule applier; which is computational system that impliments the control strategy and applies the rules.

22/07/2016

Solve water jung problem using state space searching technique.

waterjung problem:- 10 mark

U have given 2 jug of a 4 lit and 3 lit one. neith has any measuring markers, there is a premp that can be use to fill the jugs with water. How can u get exactly 2 lit water into a 4 lit. jug.?

step-01

U need to define the state space for the problem set of all order pair of integer x, y

$x = 0, 1, 2, 3$ or 4 and

$y = 0, 1, 2$ or 3

where x is the num of water that can fill into 4 lit jug.

y is the num of lit of water that can fill into 3 lit jug.

Step-②

initial state (0,0)

Step-③

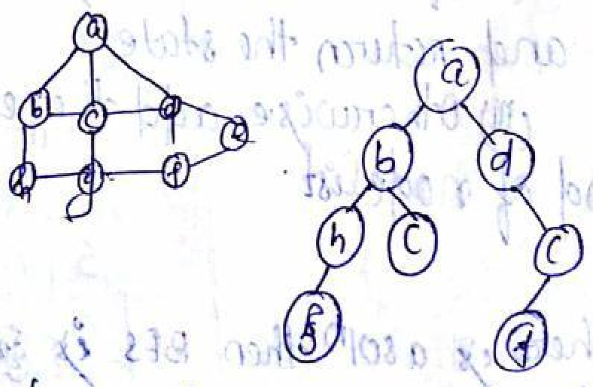
Goal state (2,n) where n is the number of lit water that can be filled in 3 lit jug.

Step-④

Production Rules.

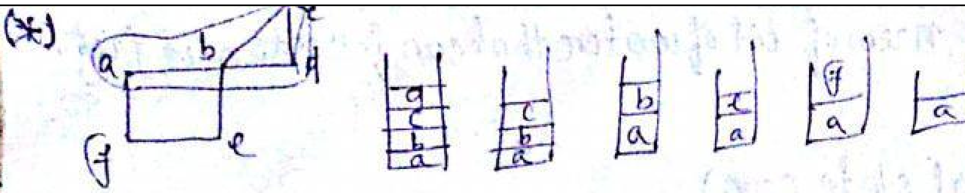
	Cond: part	Action part	Explanation
Rule-①	(x,y) and x=0	→ (4, y)	Fill 4 lit jug with water completely
Rule-②	(x,y) and y=0	→ (x,3)	
Rule-③	(4,0)	→ (0,0)	
Rule-④	(0,3)	→ (0,0)	
Rule⑤	(4,3)	→ (0,3)	
Rule⑥	(4,3)	→ (4,0)	

Breath First search :-

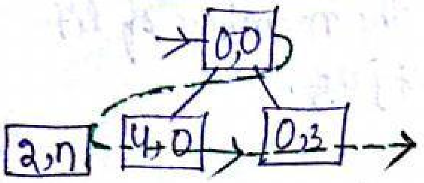


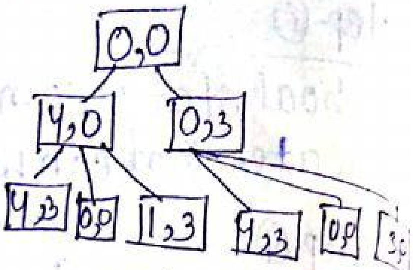
Depth First search

(*)



Initial state





1 level Breadth First search tree 2 level BFs tree.

① BFS algorithm

i) create a variable called node list and set it to initial state

2) until goal state is found, on node list is empty perform the following steps.

a) remove the 1st element of the noadlist and call it 'E'. if noad list was empty then; quit or terminate

b) for each way that each rule can match the state described in 'E' do (i) again we have further apply the rule to generate a new state

(ii) if the new state is a goal state then quit and return the state

(iii) Otherwise add the new state at the end of nodelist.

advantages:-

•) if there is a soln then BFS is guaranted to find it.

•) BFS will never get trapped by unwanted nodes.

iii) If there are multiple soln then, BFS returns the minimal soln (in minm no of steps)

disadvantages:

i) It requires more memory as it stores all the nodes of present level to search the next level.

ii) If soln is far away it consumes more time

② Depth First search (DFS) algorithm

i) step-①:- If initial state is a goal state then, quiet & return success.

step-②:- Otherwise do the following until success or failure obtained

(i) Generate a successor 'E' of the initial step. If there are no success then failure is return.

(ii) call depth first search 'E' as initial state.

(iii) If success is returned then terminate the search otherwise continue in the loop. E①

26/07/16

Advantages

→ less memory requirement, less time consuption; soln can be found without much more search.

Disadvantages

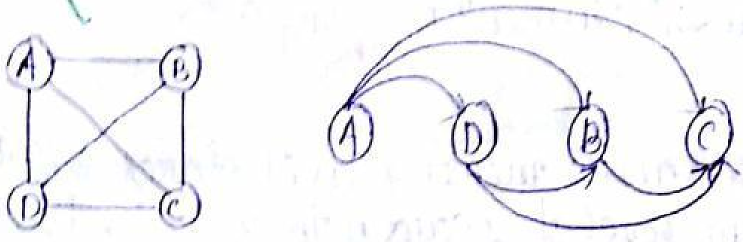
→ not guaranted soln

→ determination of depth until the search may consume the time.

Application

It is used to find connected component in a graph

③ Topological shorting :>



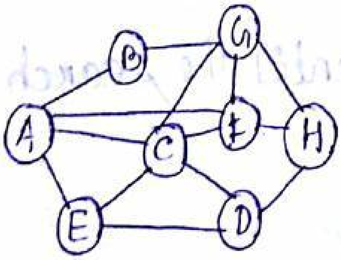
(A - D - B - C)

Heuristic search:-

i) When more information than the initial states, operator(Production Rules) and the goal step is available and then the size of the search space can be constraind → (not natural)

ii) Heuristic information is called heuristic search which is a rule of thumb of judgemental techn that leads to a solⁿ which has no guaranted success.

ex → Travelling sales person problem
→ 8- Puzzle problem



(Travelling sales person problem)

1	2	5
4	3	8
7	6	

→

initial state.

1	2	3
4	5	6
7	8	

Goalstate.

or

1	5	3
4	6	
8	7	2

intitcalstate.

→

1	2	3
4	5	6
7	8	

Goalstate.

→ The heuristic information which may require to solve a problem may be

i) The nature of the states

ii) The cost of transforming from 1 state to another

iii) The promise of taking certain path.

iv) characteristics of goals

Search technique :-

It is 2 type

1) uninformed search (Blind search) (why)? because no information, path, cost are not given

2) informed search (Heuristic search) best

uninformed search

↳ ① Breath-first search

② Depth-first search.

③ Uniform cost search

④ Iterative deepening Depth-first search

informed search

↳ i) Hill climbing search

ii) Best-first search (BFS + DFS)

iii) Generate-and-Test.

iv) Problem Reduction

v) Constraint satisfaction

vi) Branch and Bound

vii) Mean-End-Analysis

Problem characterestic :-

① In order to solve a problem it is necessery toanalize the problem along several key dimensions such as

(i) is the problem decomposable into a set of independent smaller and easier sub problem.

eg :- $\int (x^2 + 3x + \sin^2 x \cos^2 x)\, dx$

$\int x^2 dx \quad \int 3x\, dx \quad \int \sin^2 x \cdot \cos^2 x$

$\frac{x^3}{3} \quad \frac{3x^2}{2} \quad \int (1-\cos^2 x)\cos^2 x\, dx$

$= \int \cos^2 x\, dx - \int \cos^4 x\, dx$

(ii) Can soln steps be ignored or undone eg :- 8-puzzle problem

(iii) Is the universe predictable, weather the planing is possible to find desired soln or not.

(iv) Is a good soln absolute or relative (Follow book)

(v) is the soln a state or path

vi) Role of knowledge

Production system characteristics :- 28/07/2016

4 types of 1's

MPs :- monotonic production system.

NMPs :- Non monotonic production system.

PCPS :- Partially commutative production system.

NPCPS :- NON-PCPS
non partially commutative productn system.

•) MPS (Monotonic Production System):-

It is a ps in which the application of a rule never prevents the application of another rules.
(रोकना)

•) NMPS (Non monotonic Production System):-

It is just opposite of MPS in which the application of one rule prevents the application of another rules.

•) ~~PCPS~~ PCPS (Partially commutative Production system):-

It is a ps in which if the application of a particular sequence of rules transform state x in state y, then any purmutation of those rules also transforms state x in to state y.

•) NPCPS (Non Partially commutatve Prodruction System)

It is opposite of PCPS; in which

Issues in Design of search programs

There are 8 imp. issues

① → The directn in which the search will proceed -

→ Forward or backward reasoning.

→ Forward Reasoning starts from initial state and terminate desired goal state

→ The backward Reasoning or searching starts from goal state and terminate at any one of the initial state.

② How to select applicaple rules.

i) How to represent from each node of the search process : knowledge representation

Heuristics Search Technique

① Generate-and-Test:-

It is the simplest hereristic search technique.

Algo: Generate-and-test

→ Generate a possible sol^n

→ Test to see if this is actually a sol^n by compairing with desired sol^n.

→ If a sol^n has been found then quit other wise repeat step①.

advantage:-

→ It's simplest.

→ It's easy to impliment.

disadvantage:-

→ It's not very effecient Search technique

→ Many wrong sol^n may be generated.

→ It doesn't provide feed back facility to rectify errors.

imp ② Hill climbing:-

imp → It's a variant of generate & test search technique

→ In this technique provide feed back facilities to test procedure that are used to decide which direc to move in the search space.

There are 3 type of hill climbing.

i) simple hill climbing.

ii) steepest hill climbing.

iii) simulated annealing.

(i) Simple hill climbing technique :→ 29/07/16.

It is a varient of generate and test search
It is also an optimisation algorithm.

algorithm :-

step-01
① → Evaluate the initial state, if it is goal state then return it and quit otherwise continue with the initial state as current state.

② → 2nd steps :- loop until a sol^n is found

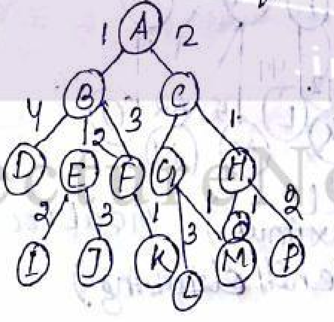
a) select an operator (Rule) that has not yet that applied to the current state & apply it to produce new state.

b) Evaluate the new states as follows

(i) if it is goal state return it & quit

(ii) if it is not goal state but better than current state then make it as current state.

(iii) if it is not better than the current state then continue in the loop.

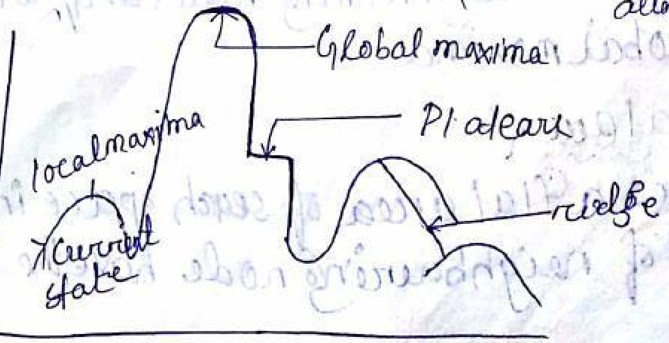


*(See which is the shortest path from the root node according to node which is given)

initial state = 1
Goal state = 0

operator
natural state = current state. 1→B→E → ① 1→2→② →② (other procedure may be allowed)

cri



3 disadvantage
→ Global maxima
→ local maxima
Current state
Plateau
→ ridge.

how to check wether a state or node betterthan other

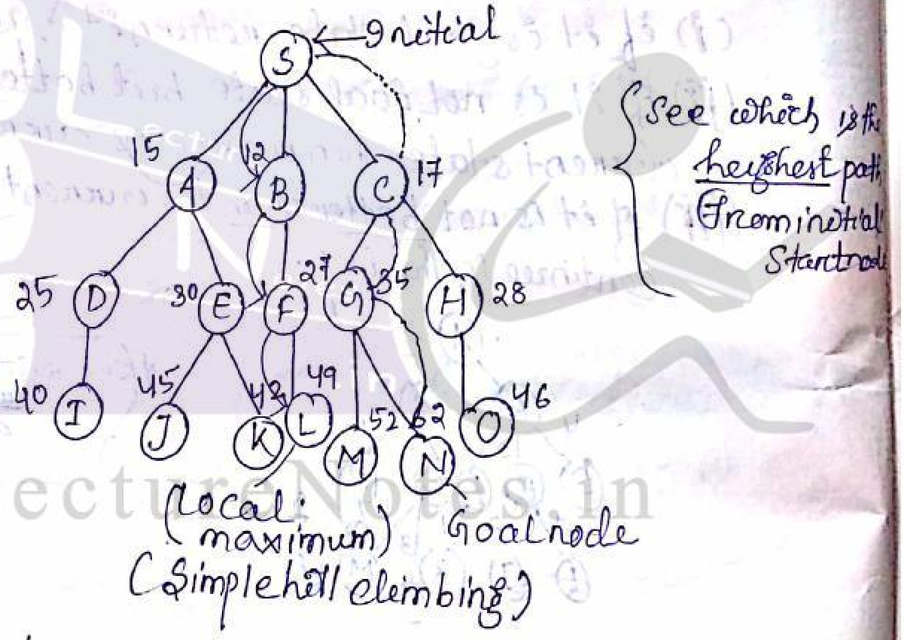
→ it's done by heuristic Function:-

→ Sometime it's alsocalled objectiveFunction or Evalulation Function.

→ Hereristic is a techniqree that improves the efficiency of the searched process.

→ Hereristic Fun^n guides the search process in the most profeatable direct^n by sugesting which path to fallow fast when more than one is available.

① Simplest hill climbing:→ (02/08/2016)



(Simple hill climbing)

Localmaxima:-

It's a state or node that is better than all of the/its neighbouring nodes and/but not better than global maxima.

Plateaus:-

It's a flat area of search space in which a set of neighbouring node have the same value.

Ridge :-

It is a special kind of local maxima for an area which has a slope and which can't be obtained by a single move.

steps/Methods to recover the 3 disadvantages

(a) Backtrack:- It is used to move to some earlier node and try going in different direction and it is use to dill with local maximum.

(b) Make Big jump :-

Make big jump in some direction to try to get a new section of the search space and it is used to deal with plateau

(c) apply 2 or more rules :-

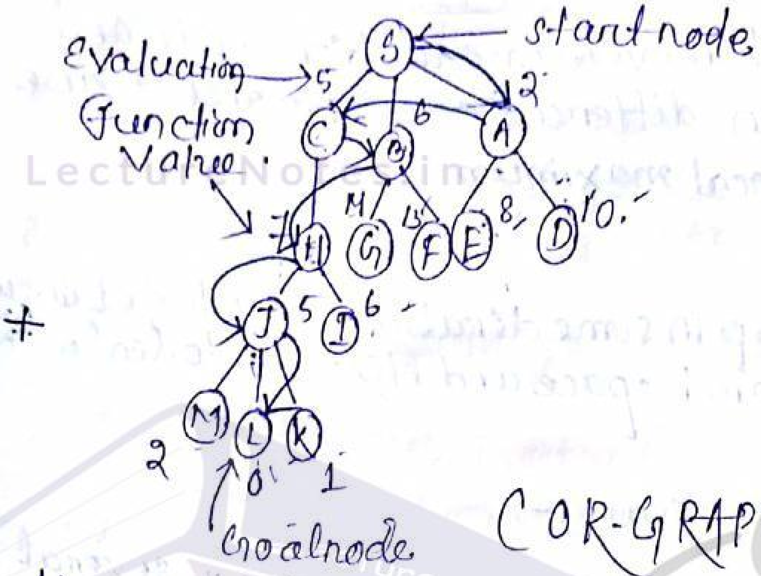
Before doing the test this cause moving in several direct^n at once. and it is used to deal with ridge problem.

Advantage of Hill climbing :-

Hill climbing search technique definitely gives optimal sol^n but it requires more memory and it is time consuming.

Best-First search

- → Graph based problem solving techniquee like hillclimbing search.
- → It combines features of BFS and DFS



(OR-GRAPH)

Algo. best First search

- → Place the starting node 's' on a prioreity quee
- → If the (P.Q) is empty return failure and stop.
- → If the 1st element of the (p.q) is a goal node the return succes and stop. otherwise;
- → Remove the 1st element of the pq expand it.
- → Compute the estimated goal distances for each child then place all the children pq and arra in ascending order corresponding to goal distances from the front of the pq
- → Return to step ②.

□

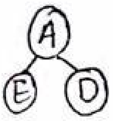
PQ

PQ	A	C	B
	2	5	6

→ start

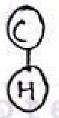
PQ	C	B	E	D
	5	6	8	10

→



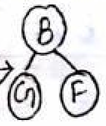
PQ	B	H	E	D
	6	7	8	10

→



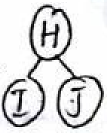
PQ	H	E	D	F	G
	7	8	10	13	14

→



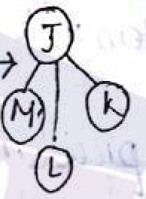
PQ	J	I	E	D	F	G
	5	6	8	10	13	14

→



PQ	L	K	M	I	E	D	F	G
	0	1	2	6	8	10	13	14

→



Goalnode

→In best-First search we jump all around in the search graph to identify the node with the min^m evaluation fun^n value.

Advantage:-

Best-first-Search is a greedy algorithm which definitely gives an optimal sol^n.

disadvantage

→ It requires more memory

→ It's time consuming.

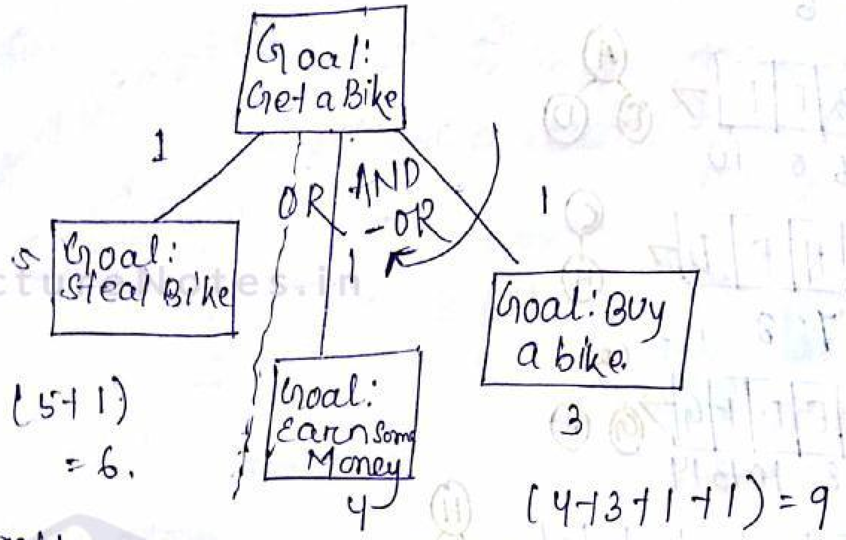
This type of graph is also called 'OR' Graph

→ The Graph on tree use to solve a problem using Best-First search is also called OR GRAPH OR TREE becoz at each step an alternative decission avail

- ie towards goal state.

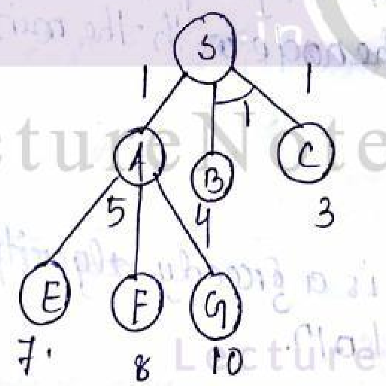
Problem Reduction

4/08/2016



→ problem reduction is a process of Reducing or decomposing a problem into sub-problem in order to get the soln.

→ The graph used in problem reduction is called <u>and or graph.</u>



(AND OR GRAPH)

Defferenceiate OR graph AND-OR graph.

→

OR Graphe	AND-OR graph
→ It represent alternative paths towards the solⁿ of the problem	→ It also represent alter-native paths towards the solⁿ but some of them may contain subpath (AND arc) all of which must then be solved inorder to solve the problem

Differenciate hill climbing and Best 1st search.

→

Hill climbing	Best 1st search
→ In hillclimbing search in each step 1 move is selected and all others are rejected.	At each step one move is selected but the others are kept around for later consideration.
→ It stops when no better succesor available	→ It doesn't stop, reconsider expanded nodes to find the solⁿ.

~o~

Artificial Intelligence

Topic:

KNOWLEDGE REPRESENTATION

Contributed By:

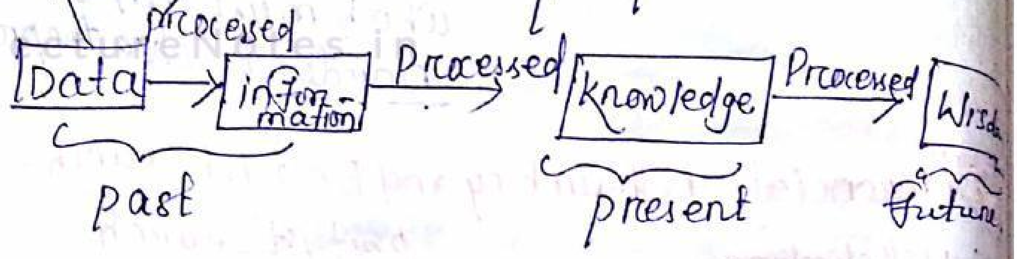
Sankarsan Sahoo

New chapter

KNOWLEDGE

Representation

It is a process of representing information about the real world in a form that a computer can utilise understands to solve complex problem



Representation & Mappings

knowledge representation system basically deals with 2 entity they are such as

i) Facts (truth about the real world)
ii) Representation of Facts.

This 2 entity can be structured in 2 levels

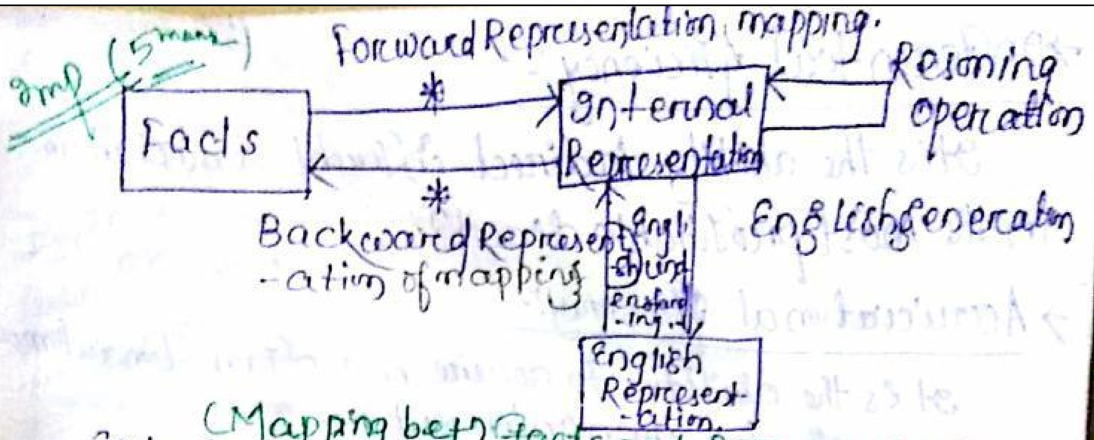
-) knowledge levels.
-) symbol level

{knowledge level :->

knowledge level at which the Facts are described

Symbol level :->

Representation of object at knowledge level and defined in term of symbols.

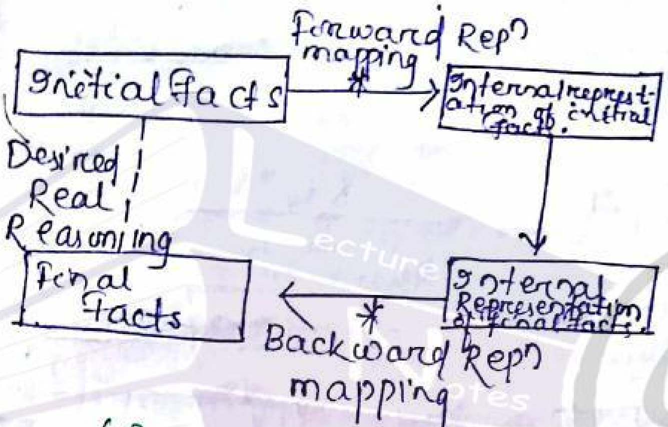


(Mapping betn facts and Representation)

Ex:- Spot is a dog ⟶ dog(spot)

All dogs have tails ⟶ $\forall x : dog(x) \rightarrow havetail(x)$

Spot has a tail ⟵ hastail(spot)



(Representation of facts)

Properties of Good knowledge Representation System

There are 4 imp. properties are present.

(i) Representational Adequacy.

(ii) Inferental Adequacy.

(iii) Inferental Efficiency

(iv) Acquisitional Efficiency.

05/08/2016

→ Representational Adequacy:-

→ It is the ability to represent all of the kinds of knowledge that are needed for given domain..

→ Inferential adequacy

It is the ability to infer new knowledge from old

Scanned by CamScanner

→ Inferential Afficiency:-

It is the ability to direct inferential mechanism in the most profitable directi.

→ Accuicational Afficiency:-

It is the ability to accure new information autocally without human intervention.

— o —

Predicate logic.

25/08/16

It is a way of representing Real world facts that an AI system might need

→ Symbols used in propositional logic:-

Name	Symbol	Meaning
Conjuction	$\wedge$	AND
Disjuction	$\vee$	OR
Negation	$\neg$	NOT
Implication	$\rightarrow$	IF .. THEN...
Double Implication	$\leftrightarrow$	If & only if (iff)
Universa qualifier	$\forall$	For all
Existential quantifier	$\exists$	There exist.

Well Formed Formula (wff's)

5 marks

Real world facts can easily be represented as logical proposition called well from formula.

Facts	wff
It is raining	RAINING
It is sunny	SUNNY
If it is raining then it is not sunny.	RAINING → ¬SUNNY.
It is not raining	¬RAINING.
Gandhi was a man	Man(Gandhi)
Gadhi was an Indian	Indian(Gandhi)
All Indians are not Pakistanies	$\forall x$: Indian(x) → ¬pakistani(x)
All childeren have parent	$\forall x$: children(x) → Has parent(x)

Not all guardians have child.

$\forall x$: Guardian(x) $\rightarrow$ Has child(x)

Every one is loyal to someone.

$\forall x \exists y$: loyal$\overset{to}{}$(x,y)

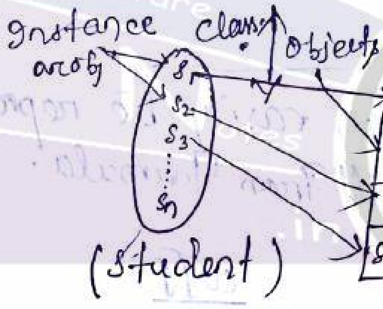
All Romans are either loyal to Ceaser or hated him.

$\forall x, \exists y$: Roman(x) $\wedge$ ceaser(y) $\longrightarrow$ loyal(x, y) $\vee$ hat(x,y)

or

$\forall x$, Roman(x) $\rightarrow$ loyal to(x, ceaser) $\vee$ hat(x, ceaser)

Representing Instance & Isa Rel^nship



student INFO

S.name	S.Regd.	S.Branch.
Shree	145	CSE(1)
Sonu	149	CSE(1)
Swasti	162	CSE(1)

* see the xerox from date-9/8/16

plz write (8/8/16) one of the copy in which is written in the middle part

First order predicate logic

Syntax of FOPL

08/08/2016

let P and Q are wff

(1) $\underbrace{\neg P}_{1}, \underbrace{P\wedge q}_{2}, \underbrace{P\vee Q}_{3}, \underbrace{P\rightarrow Q}_{4}, \underbrace{P\leftrightarrow Q}_{5}$ are also wff.

(2) Que: Parenthesi e the given Expression.

$P\wedge Q\vee\neg R\rightarrow S\leftrightarrow T\wedge U\vee\neg V$

i) $P\wedge Q\vee(\neg R)\rightarrow S\leftrightarrow T\wedge U\vee(\neg V)$

$=((P\wedge q)\vee(\neg R))\rightarrow S\leftrightarrow((T\wedge U)\vee(\neg V))$

$=(((P\wedge q)\vee(\neg R))\rightarrow S)\leftrightarrow((T\wedge U)\vee(\neg V))$

$=((((P\wedge q)\vee(\neg R)))\rightarrow S)\leftrightarrow((T\wedge U)\vee(\neg V)))$

ii) $P\rightarrow q\leftrightarrow\neg R\vee S\vee T\wedge\neg U$

$=((P\rightarrow q)\leftrightarrow(((\neg R)\vee S)\vee(T\wedge(\neg U)))))$

(3) Equivalence laws

(i) Idempotency: $P\vee P=P$, $P\wedge P=P$

(ii) Associativity: $(P\wedge Q)\wedge R=P\wedge(Q\wedge R)$,
$(P\vee Q)\vee R=P\vee(Q\vee R)$

(iii) Commutativity: $P\wedge q=Q\wedge P$

(iv) Distributivity: $P\wedge(Q\vee R)=(P\wedge Q)\vee(P\wedge R)$

(v) Demorgans law: $\neg(P\vee Q)=\neg P\wedge\neg Q$
$\neg(P\wedge Q)=\neg P\vee\neg Q$.

(vi) Conditional Elimination

Imp. $P \to Q \equiv \neg P \vee Q$.

(vii) Biconditional Elimination

Imp. $P \Leftrightarrow Q \equiv (P \to Q) \wedge (Q \to P)$

Inference Rule

These are used to perform logical proofs or derivation for deriving new sentences.

1. Modus Ponens:-

From P and $P \to Q$ infer Q.

also can be written as

$$\begin{array}{l} P \\ P \to Q \\ \hline Q \end{array}$$

It is raining

If it is raining the sky is cloudy

the sky is cloudy.

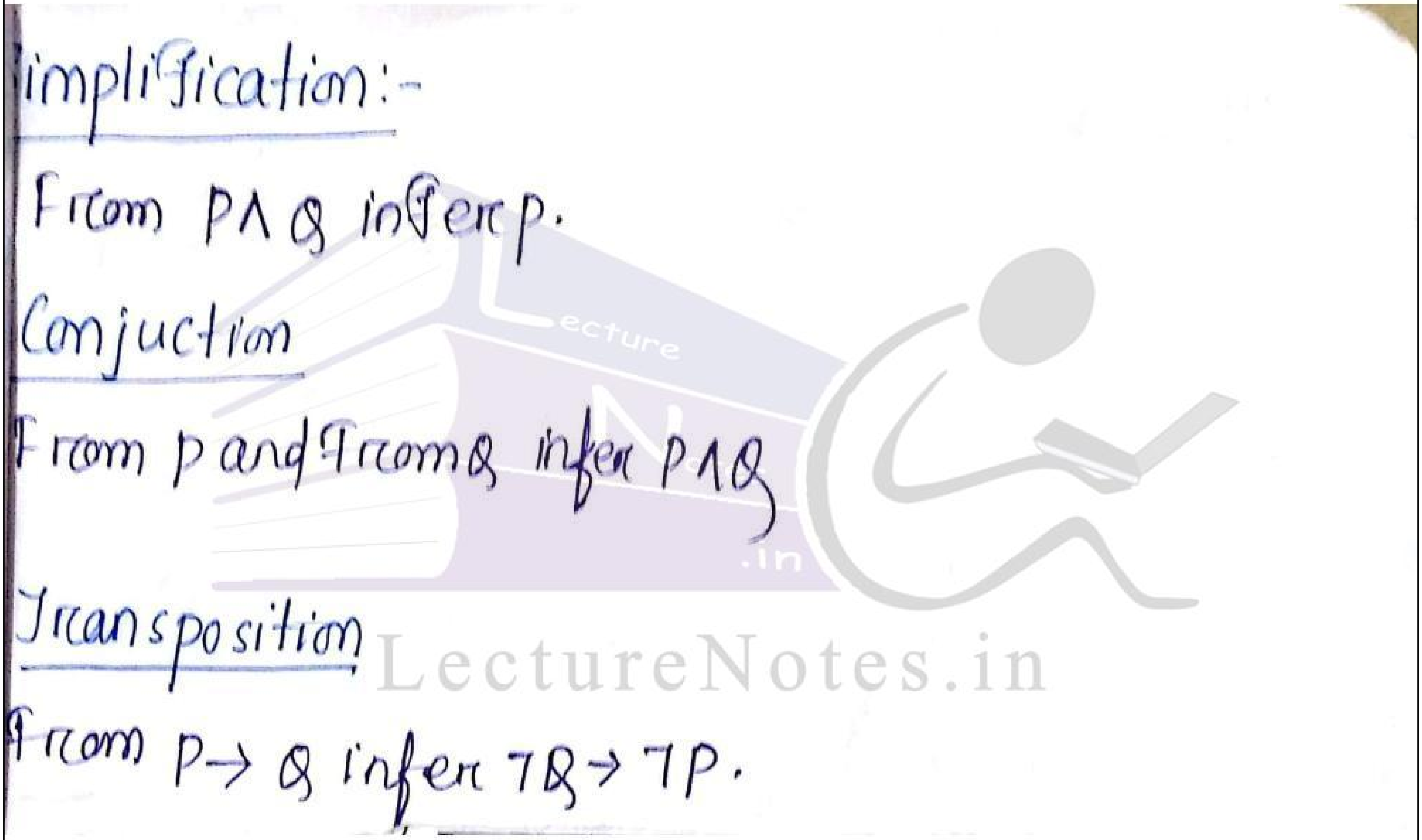
2. chain Rule:-

From $P \to Q$ and $Q \to R$ infer $P \to R$

$$\begin{array}{l} R \to Q \\ Q \to R \\ \hline P \to R \end{array}$$

3. Substitution:-

If $P \vee \neg P$ is valid then $Q \vee \neg Q$ is also valid



implification:-

From P∧Q inferP.

Conjuction

From P and from Q infer P∧Q

Transposition

From P→Q infer ¬Q→¬P.

9/8/16

Properties Of wff :-

1) **Valid :-**

A wff is said to be valid if it is true for every interpretation.

Eg :- $P \vee \neg P$

2) **Incosistent / Unsatisfiable :-**

A wff is said to be inconsistent / unsatisfiable if it is false for every interpretation.

Eg :- $Q \wedge \neg Q$

3) **Invalid :-**

A wff i.e not valid (one that is false for some interpretation) is called invalid.

4) **Satisfiable / Consistent :-**

Similarly a wff i.e true for some interpretation then it is called consistent / satisfiable.

$\neg P$	P	Q	$\neg Q$	$P \to Q$	$(P \vee Q)$.	$\neg(P \vee Q)$	$\neg P \wedge \neg Q$
T	F	F	T	T	F	T	T
F	T	F	T	F	T	F	F
T	F	T	F	T	T	F	F
F	T	T	F	T	T	F	F
				Invalid		satisfiable	satisfiable

Conjuctive Normal Form (CNF)

→ Given wffs $F_1, F_2 \ldots F_n$ each possibly consisting of the disjunction of literals only then we say $F_1 \wedge F_2 \wedge \ldots \wedge F_n$ is a conjuctive normal form.

Eg:- $\underbrace{(\neg P \vee Q \vee R)}_{F_1} \wedge \underbrace{(\neg P \vee \neg Q)}_{F_2} \wedge \underbrace{\neg R}_{F_3}$

Disjunctive Normal Form :- (DNF)

→ If each literals consists only conjuction of literals then, $F_1 \vee F_2 \vee \ldots \vee F_n$ is called disjunctive normal form.

Eg:- $\underline{(\neg P \wedge \neg Q \wedge R)} \vee \underline{(\neg P \wedge R)}$

Q. Convert the following expressions into CNF

(i) $P \rightarrow Q \rightarrow R$

$\Rightarrow ((P \rightarrow Q) \rightarrow R)$

$\Rightarrow (\neg P \vee Q) \rightarrow R$ (∵ conditional elimination)

$\Rightarrow \neg(\neg P \vee Q) \vee R$

$\Rightarrow (\neg\neg P \wedge \neg Q) \vee R$ (∵ Demorgan's law)

$\Rightarrow (P \wedge \neg Q) \vee R$ (∵ Distributive)

$\Rightarrow \boxed{(P \vee R) \wedge (\neg Q \wedge R)}$ → in CNF

Q. Convort to DNF :-

(i) $\neg(P\wedge Q)\wedge(P\vee Q)$

(ii) $P\rightarrow((Q\wedge R)\leftrightarrow S)$

(i) $((\neg(P\wedge Q))\wedge(P\vee Q))$

$=(\neg P\vee\neg Q)\wedge(P\vee Q)$ (Demorgan's law)

$=(\neg P\wedge(P\vee Q))\vee(\neg Q\wedge(P\vee Q))$ (Distributive)

$=\underline{(\neg P\wedge P)}\vee\underline{(\neg P\wedge Q)}\vee\underline{(\neg Q\wedge P)}\vee\underline{(\neg Q\wedge Q)}$

ii) $(P\rightarrow((Q\wedge R)\leftrightarrow S))$

$=\neg P\vee((Q\wedge R)\leftrightarrow S)$ (Conditional Eliminan)

$=\neg P\vee((Q\wedge R)\rightarrow S)\wedge(S\rightarrow(Q\wedge R))$

$=\neg P\vee(\neg(Q\wedge R)\vee S)\wedge(\neg S\vee(Q\wedge R))$

Prenex Normal Form (PNF) :-

(*) Skolemization :-

It is a process of removing existential quantifiers from a given expression using following steps.

step - I

a) If the left most quantifier is existential quantifier ($\exists$) then replace all occurrences of the variable it quantified with arbitary constant not appearing elsewhere in the expression, and delete the quantifier.

Eg :- $\exists x \forall y : f(y) \longrightarrow Q(x, y)$

$$\Downarrow$$

$$\forall y : f(y) \longrightarrow Q(a, y)$$

step - II

b) The same procedure to be followed for all existential quantifiers not preceeded by a universal quantifier.

$$\exists z \exists x \exists y :: f(y) \rightarrow Q(x,y,z)$$

$$\Downarrow$$

$$\forall y : f(y) \rightarrow Q(a,y,k)$$

step-II

For each existential quantifier i.e preceeded by one or more universal quantifiers replace all such variables by a function Symbol not appearing else where in the expression.

Eg:- $\exists u \forall v \forall x \exists y : P(f(u), v, x, y) \rightarrow Q(u, v, y);$

$$\Downarrow$$

After appling step-1

$$\forall v \forall x \exists y : P(f(c), v, x, y) \rightarrow Q(c, v, y)$$

After applingcstep-II

$$\boxed{\forall v \forall x : P(f(c), v, x, f(y)) \rightarrow Q(c, v, f(y))}$$

$$\downarrow$$

(said to be in PNF)

Conversion Of Clausal Form :-

→ AI system uses resolution principle for automated reasoning & theorem proving.

→ It is achieved by a process or conversion to clausal form techniques:

Algorith: conversion to Clausal Form :-

(i) Elliminate all implication (→) and equivalence (⇔) connectives.

$P \vee (Q \wedge R) = P \vee Q \wedge P \vee R$

if $P \to Q$ replace with $\neg P \vee Q$

if $P \leftrightarrow Q$ replace with $(P \to Q) \wedge (Q \to P)$

21 (2×3)

$(\neg P \vee Q) \wedge (\neg Q \vee P)$

(ii) Moave all negations in to individual atoms.

Eg:- $\neg \forall x : F(x)$ ⇓ (replace by $\exists x : \neg F(x)$

$\neg \exists x : F(x)$ ⇓ $\forall x : \neg F(x)$

(iii) Rename variables if necessary so that all quantifiers have different variable assignments.

Eg:- $\forall x: (P(x) \to \exists x (Q(x)))$

⇓

$\forall x: (P(x) \to \exists y (Q(y)))$

fc)

(iv) Skolemize by replacing all existentially quantifier variables with Skolen func^n and delete the corresponding existential quantifiers.

(V) Move all the Universial quantifiers to the left ~~th~~ of the expression and put the expression on the right into CNF.

(vi) step-⑥:- eliminate all universal quantifiers conjuction and the resulting expressions a said to be in equal form.

16 Convert the Following Expression into Clausal Expression Form V.Imp

$\exists x \forall y [(\forall z P(f(x),y,z)) \rightarrow (\exists u Q(x,u) \wedge \exists v R(y,u))]$

After applying step-01

$\exists x \forall y [\neg \forall z P(f(x),y,z) \vee (\exists u Q(x,u) \wedge \exists v R(y,u))]$

After applying step 2

$\exists x \forall y [\exists z \neg P(f(x),y,z) \vee (\exists u Q(x,u) \wedge \exists v R(y,u))$

Step-03

not required.

Step-04

After apply step-4

Remove $\exists x$ and replace x with const a

$\forall y [\exists z \neg P(f(a),y,z) \vee (\exists u Q(a,u) \wedge \exists v R(y,u).)]$

Remove $\exists z$, $\exists u$ and $\exists v$ with skdem function $h(y)$, $g(y)$ and $l(y)$

$\forall y [\underbrace{\neg P(f(a),y,h(y))}_{P} \vee (\underbrace{Q(a,g(y))}_{Q} \wedge \underbrace{R(y)\, l(y)}_{R}))]$

$\Rightarrow \forall y [(\underbrace{\neg P(f(a),y,h(y))}_{P} \vee \underbrace{Q(a,g(y))}_{Q}) \wedge$

$(\underbrace{\neg P(f(a),y,h(y))}_{P} \vee \underbrace{R(y)\, l(y)}_{R}))]$

Step-05

After applying step-5

$(\neg P(f(a),y,h(y)) \vee Q(a,g(y))) \overset{\forall x}{\wedge} (\neg P(f(a),y,h(y)) \vee R(y),l(y)).]$

After applying step-06

$(\neg P(f(a), y, h(y) \vee Q(a, g(y)) \wedge$

$(\neg P(f(a), y, h(y)), \vee R((y, l(y)))$

(Both expressions are in clausal form)

Resolution

It is a powerful inferencing technique used by AI systems for automated reasoning & extraction of answers from the knowledge base.

given 2 clauses C_1 & C_2 with no variable in common if there is a literal l_1 in C_1 & l_2 in C_2 which are compliment to each other then l_1 & l_2 are deleted and a disjuncted 'C' is form which is called resolvent of C_1 & C_2

literal:- It is an atom with no variable

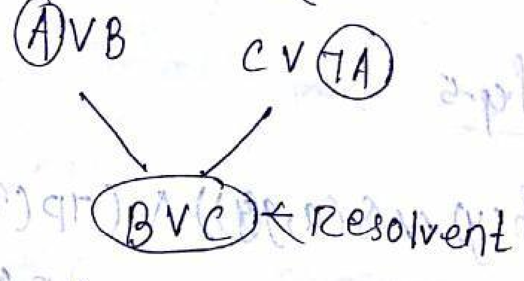
eg:- $P, P \wedge Q$

$\forall x \exists y$: loyal-to (x, y)

Clause:- It is disjunct of literals

$P \vee Q, \quad \neg P \vee \neg Q \vee R$

$C_1 \qquad C_2$



›Resolution is performed for both propositional logic & for predicate calculus.

Resolution Principle

A set of clauses $C_1, C_2 \cdots\cdots C_n$ & we wish to prove (clause D).

1st we negate D and add negation of D to the set of clauses $C_1 \wedge C_2 \wedge C_3 \wedge \cdots\cdots \wedge C_n$ and we will get

$$\boxed{\neg D \wedge C_1 \wedge C_2 \wedge C_3 \wedge \cdots\cdots \wedge C_n} \text{ — Expression (1)}$$

Resolution is to prove the 'D' we require to des prove expression (1) that also means we need to prove expression (1) is unsatisfiable

Resolution using Propositional logic

Given set of clauses C and a statement S

1. Convert all the proposition of C into clausal form.
2. Negate 'S' and add into set of clauses obtained in step 1
3. Repeat until either of the contradiction found or no progress can be made.

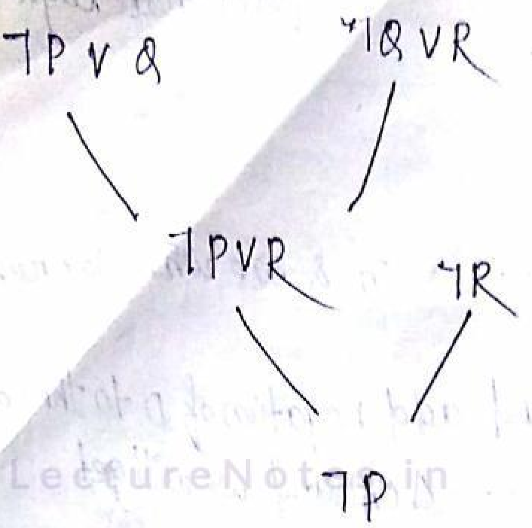
(a) select 2 clauses (parent clauses) which has complimentary literals.

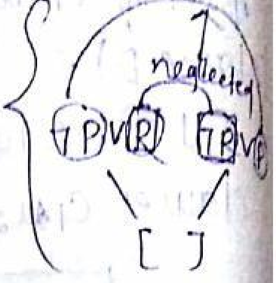
(b) Resolve them together to create the resolvent.

(c) If the resolvent is the empty clause (ϕ)(null) then a contradiction has been found otherwise add it to the set of clauses available

Ex: $P \to Q \wedge Q \to R \wedge \to R$

$(\neg P \vee Q) \wedge (\neg Q \vee R) \wedge (\neg R)$





Q. empty clauses can be written as [ ]

Given C1: if it is hot then it is humid = T→H

C2: if it is humid then it will rain = H → R

C3: it is hot = T.

D: it will rain?

T = It is hot.

H = It is humid

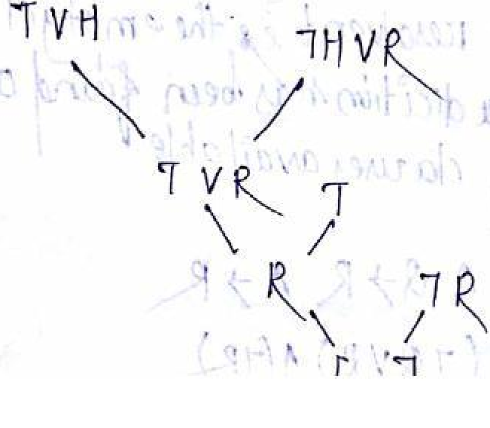
R = it will rain

negate D = ¬R

then add with C (C ∧ ¬R)

$(T \to H) \wedge (H \to R) \wedge (T) \wedge -(\neg R)$

$\Rightarrow (\neg T \vee H) \wedge (\neg H \vee R) \wedge (T) \wedge (\neg R)$



Q. Given $A \to B$

$B \to C$

$C \to D$

$D \to E \vee F$

From the above clauses

prove $(A \to F)$

$C_1 : A \to B$

$C_2 : H \to R$

$C_3 : C \to D$

$D = E \vee F$

$D \to E \vee F$

$D = A \to F$

$\neg D = \neg(A \to F)$

$= \neg(\neg A \vee F)$

$= (A \wedge \neg F)$

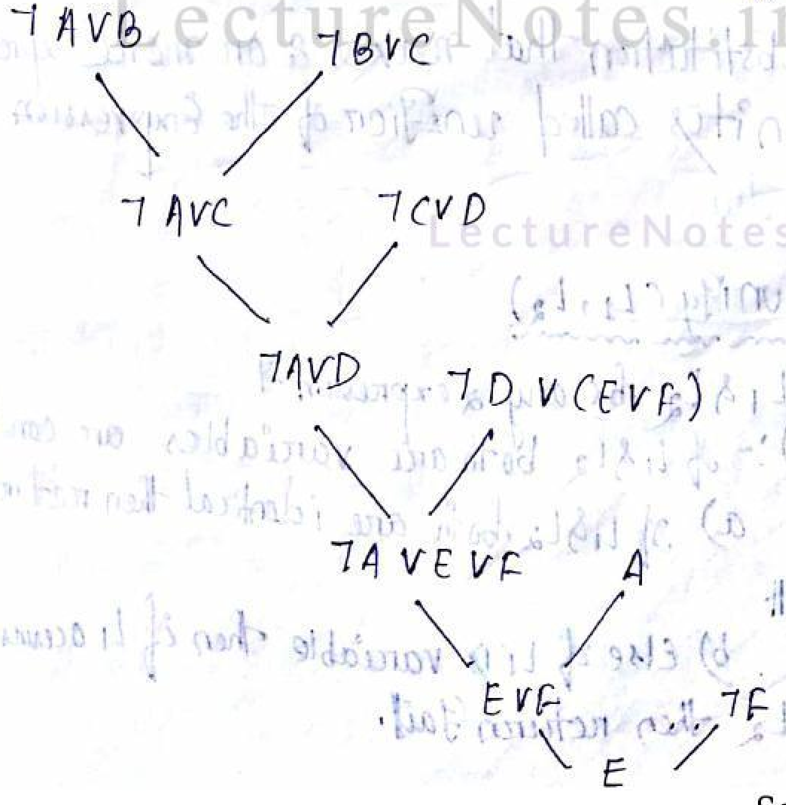
$\Rightarrow (A \to B) \wedge (H \to R) \wedge (C \to D) \wedge (D \to E \vee F)$

$\Rightarrow (\neg A \vee B) \wedge (\neg H \vee R) \wedge (\neg C \vee D) \wedge (\neg D \vee (E \vee F))$ $\wedge \neg(\neg A \vee F)$

= ~~$\boxed{\neg A \vee (E \vee F)}$~~

$\Rightarrow (\neg A \vee B) \wedge (\neg H \vee R) \wedge (\neg C \vee D) \wedge (\neg D \vee (E \vee F)) \wedge (A \wedge \neg F)$

(Apply D' law)



∴ SOA → F clause can't be derived.

12|08|2016

Substitution

{ Proposition – Literals with no variables.
  Prediction – Literals with varciables.

→ A substitution is defined as a set of paircs $\{t_i/v_i\}$ $(t_i \neq v_i)$

[{ t_i ; substitution of subst
    v_i ; Varable at the of vari

→ Such that t_i replace or substitutate for corresponding v_i in any expression for which sub stitution is applied.

Ex: $P(x,y) \vee Q(x) = L_1$

$P(a,b) \vee Q(a) = L_2$

substitution $\alpha = \{a/x, b/y\}$ is applied to L_1 to obtain L_2

Unification

Any substitution that makes 2 or more Expression Equal then it is called unifier of the Expression.

Procedure

Algo: unify (L_1, L_2)

Let L_1 & L_2 be any 2 expression +

① :- If L_1 & L_2 both are varciables or consts then
a) If L_1 & L_2 both are identical then return nill.

b) Else if L_1 is varciable then if L_1 occurs in L_2 then return Fail.

c) then else if L_2 is variable then return Fail. we happy

d) Else return Fail.

② if the initial predicate symbol in L_1 & L_2 are identical then return Fail.

③ if L_1 & L_2 have different no. of argument then Fail

④ set subset to Nil (at the end of the procedure subset will return all the substitution) use to unify L_1 & L_2

⑤ For $i = 1$ to num of argument of L_1

a) call unify with the 'ith' argument of L_1 and argument 'L2' and putting the result in 's'

b) if 's' contain nill then return Fail.

c) if 's' is not equal to nill

$s \neq Nil$

(i) apply 's' to the remainder of both L_1 & L_2

(ii) SUBSET = APPEND(S; SUBSET)

⑥ Return subset.

Eg:- $L_1 = P(q(x), y, z) \wedge Q(p, y)$

$L_2 = P(a, b, c) \wedge Q(m, n)$ (wrong)

$L_1 = P(q(\check{x}), \check{y}, \check{z}) \wedge Q(\check{p}, y)$ (here 4 argument)

$L_2 = P(q(\check{a}), \check{b}, \check{c}) \wedge Q(\check{p}, b)$ (,,)

$\frac{L_2}{L_1}$ Remember

SUBSET = ϕ, $i = 1$, $s = a/x$

SUBSET = $\{a/x\}$ $i = 2$, $s = b/y$

(so we are use to solve this procedure)

SUBSET = $\{a/x, b/y\}$

$i = 3, S = c/z$

$$\boxed{\text{SUBSET} = \{a/x, b/y, c/z\}}$$

Resolution in Predicate logic

Given a set of statements F and a statement to be proved 'P'.

Algorithm

① → Convert all the statement of 'F' to clausal form

② → ¬P negate P and convert the result to clausal form

③ → Add it to the set of clauses obtained in step-1.

→ Repeat until a contradiction is found or no progress can be made.

a) Select 2 clauses as parent clauses and (b) resolve them together with appropriate substitution & unification. and deleat complimentary item

c) If the resolvent is empty clause(N then a contradiction has been found

If it is not then add it to set of clau available to the procedure.

Ex:- S_1: All people who are graduating are happy

S_2. All happy people smile

S_3. Some one is graduating

Is some one is smiling?

C_1 $\forall x$: graduating(x) $\longrightarrow$ happy(x)

C_2 $\forall x$: happy(x) $\longrightarrow$ smile(x)

C_3 $\exists s$: graduating(x)

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

C_1: $\forall x$: $\neg$graduating(x) $\vee$ happy(x)

C_2: $\forall x$: $\neg$happy(x) $\vee$ smile(x)

C_3: $\exists x$: graduating(x) ~~or~~ convert to C_3: $\forall y$:graduating(y)

P : $\forall x$: $\neg$smile(x)

$\neg$graduating(x) $\vee$ happy(x) $\qquad$ $\neg$happy(x) $\vee$ smile(x)

$\neg$graduating(x) $\vee$ smile(x) $\qquad$ graduating(y)

{y/x}

smile(x) $\qquad$ $\neg$smile(x)

[ ]

$\vee$

$\forall x$: $\neg$P(x) $\wedge$ smart(x) $\rightarrow$ happy(x)

Artificial Intelligence

Topic:

KNOWLEDGE REPRESENTATION USING RULES

Contributed By:

Sankarsan Sahoo

1st chapter

23/08/16

Knowledge Representation using Rules

What is knowledge:-

It is fact, information, skill accured through Experience or education. It can be further classified into 2-types they are

(i) Declarative knowledge
(ii) Procedurral knowledge.

Declarative knowledge	Procedrercal knowledge
→ It Represents facts - what things are	→ It tells us facts & procedrere for solving Problem. (what & how)
→ It is expressed using propositional logic or predicate logic	→ It is expressed using logic programming.
Ex:- $\forall x: pet(x) \wedge small(x) \rightarrow appartment pet(x)$	→ PROLOG Ex:- apperctment pet(x):- pet(x), small(x)
ex:- $\forall x: cat(x) \vee dog(x) \rightarrow pet(x)$	ex:- pet(x):- cat(x). pet(x):- dog(x)

ex:- $\forall x: poodle(x) \rightarrow dog(x) \wedge small(x)$	dog(x):- poodle(x)
	small(x): poodle(x)

→ Quantifier are explicit to the variable.	→ quantifier are implicit to the variable

Imp

PROLOG

A prolog program is described as serise of logical assertions each of which is a horn clause

A hornclause is aclause which has atmost one positive leteral.

Eg:- $P, \neg P \vee q$

Predicate logice	PROLOG
→ and ($\wedge$) and or ($\vee$) are 2 explicit of symbols	→ For and (,) and none for or. Disjunctions represen-ted as a cost of alternative statements.

→ $\forall x$: cat(x) $\vee$ dog(x) → pet(x)	pet(x) :- cat(x)
	pet(x) :- dog(x)

→ "P implies q" is written as $P \rightarrow q$	→ q :- P because the prolog interpreter works backwards from a goal.

Forward vs Backward Reasoning

Imp

(plz see the next page) - Reasoning is the process of making conclusion and judgements from given facts or condition

Rule -based System

→ Rules has 2 component parts L.H.S & R.H.S

→ Rules can be considered as subset of predicatelogic

2 component are such as

1. LHS
2. RHS

→ L.H.S describes the cond? or situation and RH.S describes the conclusion or action

Eg: If : The sky is cloudy (L.H.S)
THEN: It will rain (R.H.S)
If: A&B
THEN: C

→ A rule based production System has 3 important parts and they are

i) a KB (knowledge base) which consists a set of rules.

ii) a set of rules.

iii) a working memory

iv) a rule interpreter or inference engine.

PROLOG

Predicate logic	Prolog
$\forall x : P(x) \rightarrow Q(x)$	$Q(x) :- P(x)$
$\forall x : \neg P(x) \vee Q(x)$	$Q(x) :- P(x)$
$\forall x : cat(x) \vee Dog(x) \rightarrow Pet(x)$	$pet(x) :- Cat(x)$ $pet(x) :- Dog(x)$

Prolog uses syntax of Predicate logic to perform symbolic & logical computation.

Database :-

IF : Parent (x,Y)
and parent (Y,Z)
and Male (X)
THEN : Grand Father (x,Z)

Database

sister (Sue, bill)
parent (ann, sam)
(mama) (I)
parent (joe, ann)
(mama's papa/nana)
male (joe)
female (ann)

Queries

?- parent(x, sam)
x = ann
?- male(joe)
yes.
?- Grand Father(x,y)
x = joe Y = sam
?- female(joe)
no.

Advantage :-

→ It is able to derive new rules from the existing content with in the knowledge base.

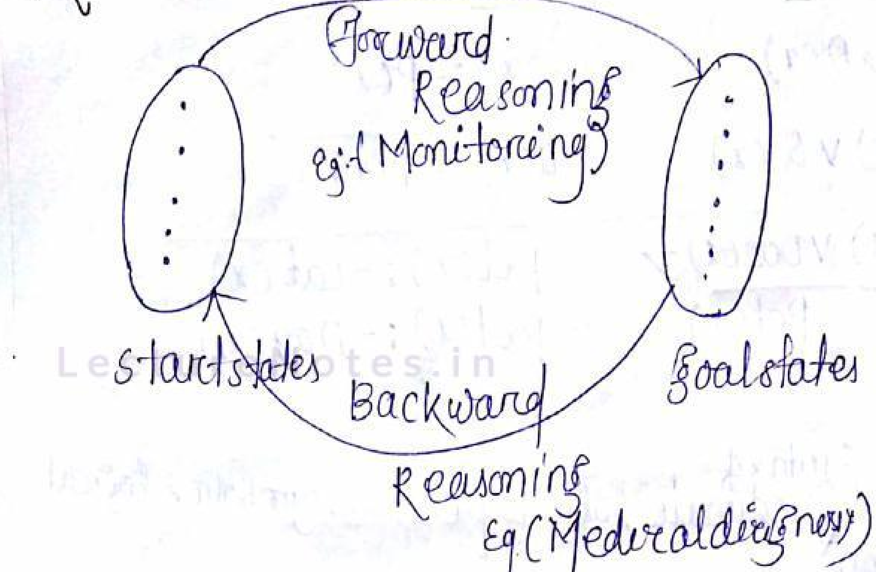
→ It is also used for Queries purpose

Disadvantage

It is inefficient for solving complex arithmatic computation.

Forward & Backward Reasoning

Imp.



Forward chaining / Forward Reasoning :-

→ It starts from the starts state & proceeds toward goals state as shown in the diagram

Eg: The resolution principle in predicate logic uses forward reasoning.

Modus Ponens

IF : P

and P→Q

$\overline{}$

Q.

P   ¬P V Q

Q

Backward Reasoning / Backward chaining :-

→ It is also called back chaining. its starts from the goals state & proceeds towards the starts state

Eg: ProLog is an example of Backward reasoning .

Rules for forward & Backward Reasoning

[Before write this ans you have to write what is Reasoning]

There are 2 classes of Rules that can be used for both Forward & backward reasoning & they are

a.) Forward Rules

b.) Backward Rules

a.) It describes how to respond a certain i/p

b.) It describes how to achieve a particular goal.

There are 3 systems by which the above 2 class of a rules can be used for problem solving & they are

i) backward chaining rule system.

2) Forward chaining rule system

3) combination of both (forward & backward) chaining rule system.

1/09/2016

Forward Reasoning →

Facts	Rules	Conclusion
a = 1	if a = 1 & b = 2 then c = 3	d = 4
b = 2	if c = 3 then d = 4	

← Backward Resoning

1. * Backward chaining rule system :-

Ex:- of backward chaining rule system is PROLOG

Ex: PROLOG

P → Q in prolog q :- P

→ It is a goal driven problem solving approach

Ex 2: MYCIN

It uses backward reasoning from its goal to determine the caues of a patients illness.

Forward chaining Rule system:-

Ex1: Predicate logic

→It uses incoming data directed problem solving

Ex2: Matching

Combination of Forward & backward reasoning

→It is also called hybrid reasoning.

→It uses by dierectional search for problem searching/solving.

→ sometimes certain aspect of a problem are best handeled via- forward reasoning & other aspects by backward reasoning..

\ it is called hybrid reasoning.

Matching:-

→It refers to matching betn the current state & the preconditions of the rules, that are most likely to lead to a solution.

→Matching in rule based system can be achieved by following 4 proposals.

a) indexing

b) Matching with variables

Imp c) complex & Approximate matching.

d) conflict Resolution.

a) Indxing :-

It is the process of compairing of each rules pre-cond^n to the current states and extracting all the one's that match

b) Matchingwith variables :-

It is applied when pre cond^ns are not stated as required

Ex :-

a = 1
b = 2   if a = 1 & b = 2 then c = 3

Given that at b

a) x = 1   then apply the above rule after applying
b) y = 2   substitution a/x , b/y

Complex & Approximate matching :-

it is required when precond^n of a rule specify requi-red property that are not describe in current ~~state~~.

3) Conflict Resolution :-

In conflict resolution it is the process of resolving the order in which the rules will be applied if conflicts occure.

There are 3 basic approaches.

① Preference based on rule

- * FCFS based (Fast come Fast searche)
- * Priority based

② Preference based on object

Ex: ELIZA. (It is an AI system used in NLP)

③ Prefference based on states

→ If there are several rules waiting to Firer (exicute) then Fire all of them temporarily & examine the resu

→ each. then using heuristic Funn evaluate each select the prefferedone & discard the remain

Control knowledge

→ Knowledge abod which paths most likely to le Quickly to a goal state is called control knowle

→It can takes many Forms such as

·) knowledge about which states are more preferable than others.

·) knowledge about which rule to apply in a given situation

·) knowledge about the order in which to persue subgoals.

·) The knowledge about useful sequences of rules to apply.

Artificial Intelligence

Topic:

SYMBOLICREASONING UNDER UNCERTAINTY

Contributed By:

Sankarsan Sahoo

new chapter

Symbolic Reasoning under Uncertainty

06/09/2016

Limitations of conventional Reasoning

Ex:- FOPL
PL

⇒ Limited in expressing power

⇒ unable to express uncertain imprecise and hypothetical knowledge

⇒ Only true or false.

⇒ Inefficient inference Method.

⇒ unable to produce new knowledge.

⇒ It can only add New knowledge i.e derived from existing knowledge.

⇒ So the conventional reasoning also called monotonic reasoning

⇒ Our dynamic world consists of different from of

(i) inconsistencies
(ii) uncertainties
(iii) Possibility & beliefs

⇒ Differentiate bet?:-

Monotonic Reasoning (MR)	Non Monotonic Reasoning (NMR)
⇒ The size knowledge base alwayes increases Monotonically	⇒ The size of the knowledge base(kb) increases non-monotonically
⇒ No retractions (removal) of rules holding/pulling	contradicts negative/false ⇒ It allows retraction of contradicting facts

are allowed.

⇒ All new knowledge or Facts added to kb must be consistent with the previous knowledge or Facts

⇒ Eg:- FOPL, PL (logical, order mannerby acordenly)

⇒ In uncertenities add of new fact may contrad or valided old fact so it's inconsistent

⇒ Eg:- TMS (Truth mentainance System)

Logics For Non Monotonic Reasoning

•) default Reasoning:-

It draws conclusions based on what is most likely to be true

There are 2 approches they are

•) Non monotonic Logic

•) default logic

-(Firstorder predicate logic - FOPL)-

•) Non Monotonic Logic (NML)

i) It augment the Fact of FOPL with a modal operator 'M' which can be read as "Is Consistent

Eg :-

$\forall x, y$: Related $(x, y) \wedge M$ GetAlong$(x,$

$\rightarrow$ will defend (x, y)

It should be read as. $\forall x \& y$, if $x \& y$ are related and if 'x' gets along 'y' is consistent with everything else i:e belived, then conclude 'x' wil defend 'y'.

•) Default Logic (DL)

i) It allows inference rule of the form,

$$\boxed{\frac{A : B}{C}}$$

which is read as;

if A is provable & it is consistent to assume B then conclude 'C'.

A: Ram is a professor
B: Professor makes lecherenotes

C. Ram makes lecturenotes.

differentiate

Non Monotonic logic (NML)	Default logic (DL)
⇒ It does n't cause extension to the database	⇒ It causes extension to the data base.
⇒ Here non monotonic expressions are expressed in language so, they can be Manipulated	⇒ Here non monotonic expression are rules of inference & they can't be mannipulated with other rules of inference.

Abduction

⇒ It is opposite of deduction

⇒ eg:- of deduction

$$\forall x : Measeles(x) \rightarrow spots(x)$$

$$Measeles(A) \rightarrow spots(A)$$

The reverse many not true.

⇒ deriving conclusion in reverese direction is the another form of default reasoning & it is called abductive reasoning.

Eg:- Some people can't see
Tim continued walking into objects

Jim can't see.

Inheritance

class of entity — attribute — value

$$\text{Eg:- } \frac{\text{Baseball player}(x) : height(x, 6.1)}{height(x, 6.1)}$$

⇒ It is a rule that inherit atribute, values for a class of entities.

8-09-16

Minimalist Reasoning :-

>It is a kind of NMR (Non-monotonic Reasoning)

>It assumes only true statements inorder to maintain the consistency of the KB (knowledgebase)

>There are 2types of logics used for MR (minimal Reasoning) & they are

- •) closed world Assumption (CWA)
- •) Circumscription.

•) Closed world Assumption (CWA)

→It is a simple minimalist Reasoning (MR)

→It is a powerful MR used in the database.

Eg:- student.

R.no	S.name	branch
151112	Swasti	CSE
151102	Sonu	CSE
151093	Shree	CSE
152130	Rakesh	EE
153102	Sambit	ECE

Eg:- Mr.x is a citizen of Israel.

Is Mr.x is a citizen of USA = Ans:- No. (CWA)
unknown (OWA)

→CWA is a assumption that what is not known to be true must be false

>OWA is just opposite of CWA.

>CWA display result that satisfy any predicate (Condition) p.

ii) Circumscription

-> Circumscripe means restrict with a limit

-> A set of value (for which a particular is true is to be circumscribed.

Eg :- $\forall x : Adult(x) \wedge \neg AB(x) \rightarrow Pensioner$

possible value of AB = {student, business man, minor .....}

Implementation Issues

There are 4 imp challenges or problems that arise by implementing NMR in problem solving

(i) Deriving non-monotonic conclusions with wasting time.

(ii) updating knowledge incrementally to maintain the truth status of the rest of the KB.

(iii) Multiple interpretations of the knowledge facts make it difficult to manage.

(iv) These are not computionally effective some are semidecidable.

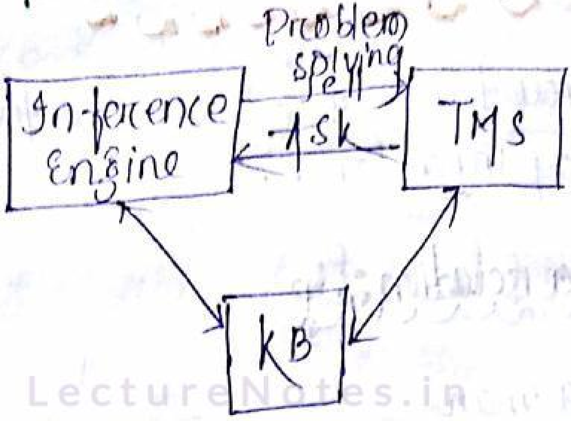
Solutions :-

The reasoning process in NMR is now separated in two parts and they are

i) Problem solver

ii) Truth Maintainance System (TMS)

-Augmenting Problem Solving:-



→ Problem solving can be done using either Forward reasoning or backward reasoning using following 2-types of approaches.

(i) Reason Forward from what is known

(ii) Reason Backward to dertermine whether some expression 'p' is true or not

→ Implementation of problem solver can be doneby using

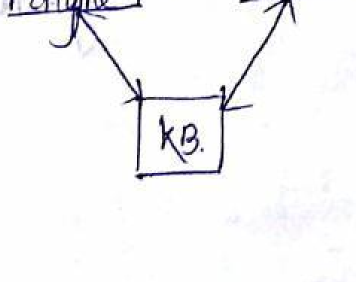
a) depth First search (DFS)

b) Bredth First search (BFS)

Truth Maintainance System

→ It is used to maintain the consistency in KB.

→ It maintains complete list of reasoning For belief



Artificial Intelligence

Topic:

SLOT AND FILLER STRUCTURE

Contributed By:

Sankarsan Sahoo

new chapter

Slot and Filler Structure.

Slot and Filler structure's are considered as de, to support property inheritance.

Instance and is a relationship

Eg:- Marcus was a man

→ Man (Marcus)

→ Instance (Marcus, Man)

Marcus was a pompeian.

→ instance (Marcus, pompeian)

All pompeians are Romans.

→ is a (Pompeian, Roman)

09-09-2016

Knowledge Representation

- unstructured Knowledge Representation
- structed knowledge Representation.

Eg:- FOPL
PL
Non monotonic logic
Default logic
CWA
Circumscription

Eg→ slot-and-filler system.

Question:- what is slot & filler System?

> knowledge in slot & filler system is structured as a set of entities and their attributes

> there are 2 types of slot & filler structure

- •) weak slot-n-filler structure
- •) strong slot-n-filler structure.

weak slot-n-filler structure	strong slot-n-filler structure
> It uses weak methods for problem solving	It uses strict-rules for > problem solving.
> Eg:- * semantic Nets * Frames	> Eg:- * Conceptual Dependency * Scripts * CYC (artificially project trademark)

(*) Weak slot & filler structure:-

•) There are 2 types of weak slot-n-filler structure they are 1. semantic nets
2. Frames.

Semantic nets (SN)

> It represents semantic rel^nship bet^n concept.
meaning

> it is introduced by Quillian to model semantics of sentences & words.

> In SN the information is present as a set of labeled nodes connected to eachother by a set of labeled links which represents rel^nship among the nodes.

Eg: Tweety is a bird. it can fly. It has wings. it's colour is Yellow.

Fig-①

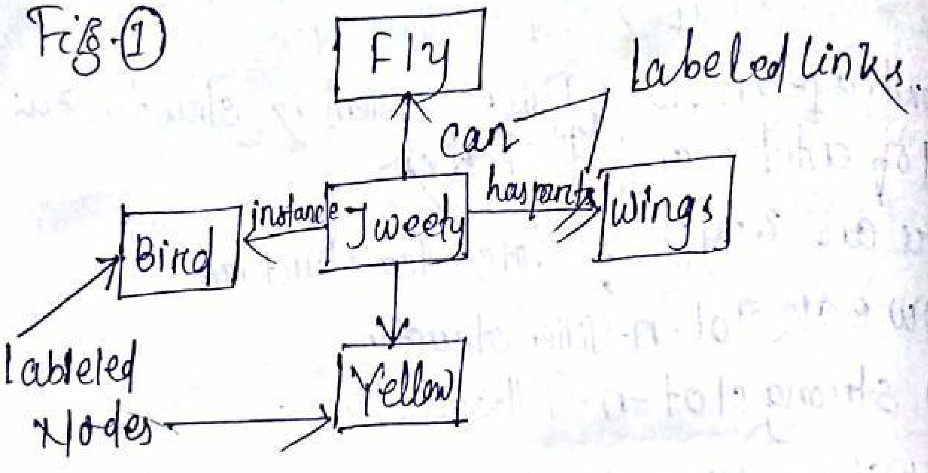
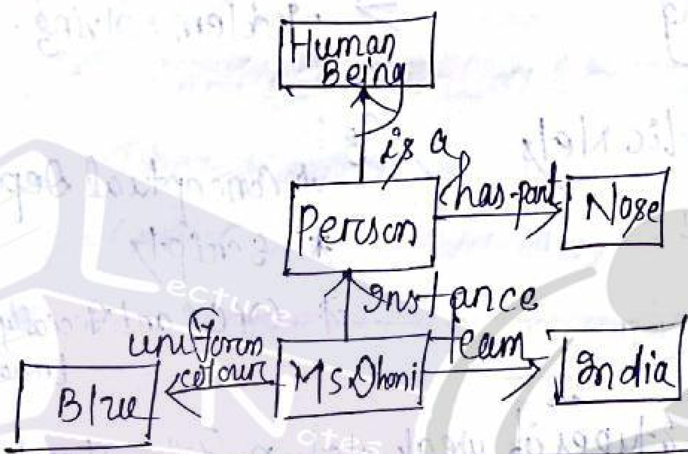


Fig-②

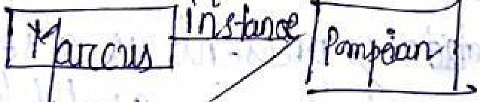


Question:- construct semantic net representation for the following.

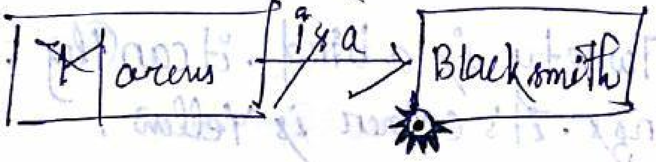
a) Pompeian (Marcus)

b) Black smith (Marcus)

a) instance (Marcus, Pompeian)



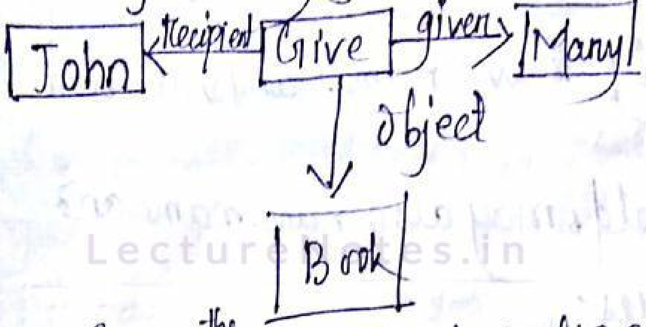
b) is a (Marcus, Black smith)



Non binary Rel^n ship

'The' is a instance, 'has' parts are all binary rel^n ship

eg:- Mary gave a book to John



Give ~~is~~ the event which is also another labeled nodes.

Q. Assignment

Construct semantic net For Following Fact

"Marry gave the green Flowered vase to her Favourite Cousine"

Advantages

1→Simple to implement

2→Easy to understand.

3→More expressive than logic representation.

4→Permit simple approach For problem solving.

Disadvantage

1- No difference bet^n individuals and classes

2- Attributes are not specified.

Frames

Frames are general record like structure which consists of slots and slot values.

slot typically have names and values or subfields.

Again subfields may also have names and any no of values.

```
                 slot      subfield/slot      Value
     (Bob
Eg:- (Profession (values Professor))
     (Age (value 50))
     (wife (value sandy))
     (children (value sue joe))
     (Address (street (value 100 elm))
              (city (value dallas))
              (state (value texas))
              (zip (value 73301))))
```

Q. Give a frame based representation of the following facts.

"Ramesh is a 52 years Professor of Mathmatics in Delhi University; The name of his wife, son, daughter are respectively Seema, Yash, Kabita.

Advantages of Frame

→Frames are Flexible as compaired to prodcution rule based representation.

→Frames are easily understandable by a non progr-ammer

→Frame structure is possible in which new slots and values can easily be added.

disadvantage of Frame

→ It can't be used for reasoning purpose

→It has no standards, it only contents slot & Filler values.

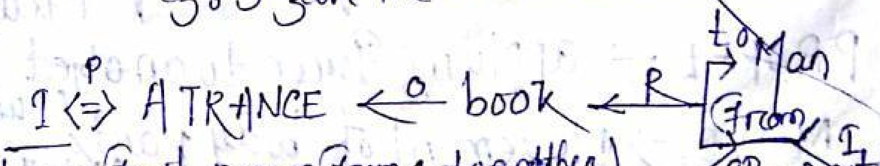
(*) Strong slot and Filler structures

3 types 1- Conceptual dependency
2- Scripts.
3- CYC

① Conceptual dependency :-(C.D)

→It is based on the use of limited no of primitive concept & rules of formation to represent any natural language statement.

Eg: I gave the man a book



(transfer of person from 1 to another)

P = past tense.
R = Recipient
O = object

PTRANS → physical transfer
MTRANS → mental trasfer.

15/09/16

There are 5 types of ontological building block are used in cD and they are .

i) Entities
ii) Actions
iii) Concept Cases
iv) semantic rule
v) Conceptual tenses

Entities :

PP (Picture producer) - actors or physical objects that perform.

PAC (Picture Aider) :- supporting properties of PP.

Actions

These are primitive actions .

Eg:-

Primitive actions	Meaning
ATRANS →	Transfer of abstract entity. (give)
PTRANS →	Physical transfer from 1 location to another (go)
MTRANS →	Transfer of mental information (tell)
PROPEL →	applying force to an object (Push)
MOVE →	Movement of body parts (kick)
Speak →	production of sound (say) .

Conceptual cases

O - Obj. case
d - directive case
i - instrumental case
R - Recipient case.

Semantic Rules

These are used for formation of dependency structur-ed such as rel^n ship bet^n such as an actor & an event or bet^n a prievitive action on instrument.

Conceptual tenses

past(P), present(nil), future(F), conditional(c), contimuing(K), negative(1) etc.

Ex:-

Rule	Example of use	eg. sentence
PP $\Leftrightarrow$ ACT	Bird $\overset{P}{\Leftrightarrow}$ ATRANS	Bird flew
PP $\Leftrightarrow$ PA	John $\Leftrightarrow$ doctor	John is a doctor
ACT $\xleftarrow{O}$ PP	JOE $\overset{P}{\Leftrightarrow}$ PROPEL $\xleftarrow{O}$ door	JOE pushed the door.
ACT $\leftarrow \left[\begin{matrix}\rightarrow PP \\ \rightarrow PP\end{matrix}\right.$	JOE $\overset{P}{\Leftrightarrow}$ ATRANS $\xleftarrow{O}$ flower $\leftarrow \left[\begin{matrix}\xrightarrow{to} sue \\ \xleftarrow{from} joe\end{matrix}\right.$	joe gave sue a flower

Advantage:

→ It involves fewer inference rule
→ It provides both structured and specific set of primitive for information construction.

Disadvantage

→ It is dificult to find correct set of primitive
→ A lot of inference still may be required
→ Complex representation for simple action

-> It requires more memory for storage

② Scripts

It is a structure that describes a sterio-type situation are events like going to the movies, shopping to a super market etc.

- It is similar to Frame structure but with small specialize rows

script name	food market
Track	super market
Roles	shopper daily attendant Check out clerk sacking clerk other shopper.
Entry condition	shopper needs groceries Food market open
props	shopping cart display chart market items check out stands. casheer Money

Scanned by CamScanner

Scene 1 : Enter Market

Scene 2 : Shop for items

Scene 3 : check out

Scene 4 : exit market

Result : shopper hass less money
shopper has items
Market has items.
Market has more money.

Advantages :-

•) Ability to predict events.

•) A single coherent interpretation may be buildup from a collection of observations

disadvantages :-

•) It less general than Frames

•) It may not be suitable to represent all kinds of knowledge

20/09/16

strong slot & filler

② CYC

→ It is a very large project used to encode huge knowledgebase.

→ like conceptual dependency it can be used in natural language understanding.

→ It aims to enable AI application to perform reasoning

Application

→ Encyclopedia

→ Terrorism KB

→ CYC's knowledge is encoded in a representation language called CYCL

CDS	CYC
→ It is less comprehensive	→ It is more Compreh
→ It only specify representation of event	→ It specify represent of event, object, att and so forth
→ less size	→ huge size

Artificial Intelligence

Topic:

UNDERSTANDING

Contributed By:

Sankarsan Sahoo

Module-II

4chapter

-) Game playing
-) planning
-) Understanding
-) Natural language processing.

UNDERSTANDING :-

Q What is Understanding?

To understand something is to transferit from 1 representation in to another in order to perform appropriate action for it.

imp. 5 marks 2A

What makes understanding hard?

There are 4 major factors they are such as

i) complexity of target Representation

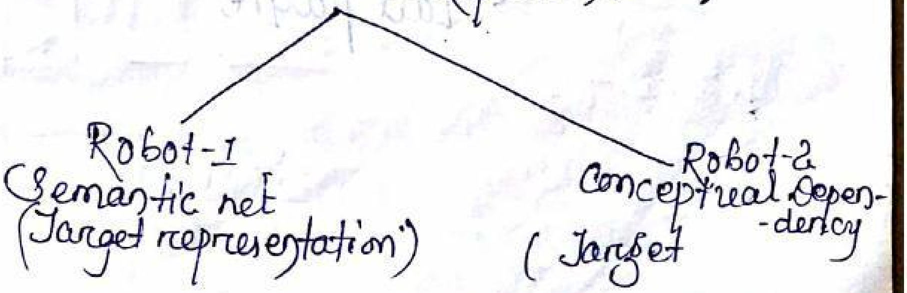
ii) Types of mapping

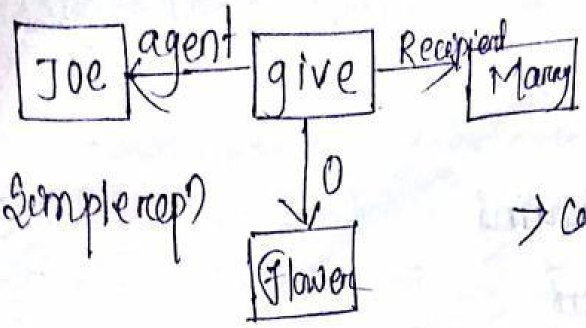
iii) Level of interaction of the components of the source representations.

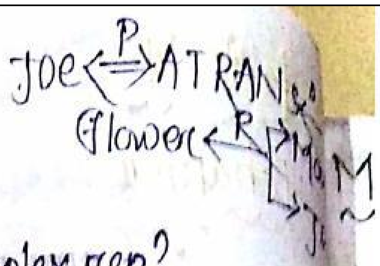
iv) presence of noise in the i/p to the understander

•) Complexity of target representation:-

Joe gave a flower to Mary
(Source Representation)







Simple rep^n → complex rep^n

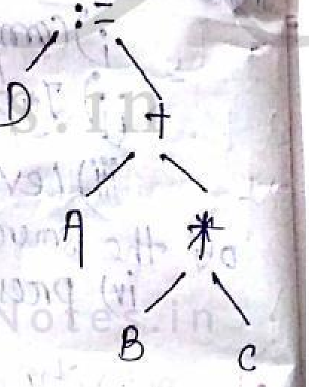
→ uses weak method → uses strict rules.
→ Easy to construct → Difficult to construct.

•) Types of Mapping

There are 4-types of mapping

complexity increases
•) one-to one
•) one-to many
•) many-to one
•) many-to many

(i) One to one
Eg:- D:=A+B*C $\xrightarrow{\text{mapped}}$



(ii) One to many
Eg:- tall — tall giraffe
tall — tall poodle

Many to one

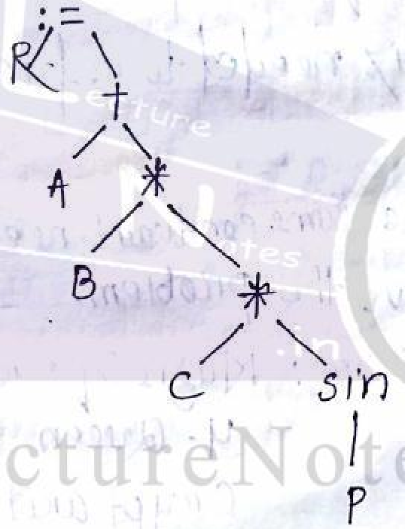
Natural language spoken ⟩ teach
Natural language written

Many to many

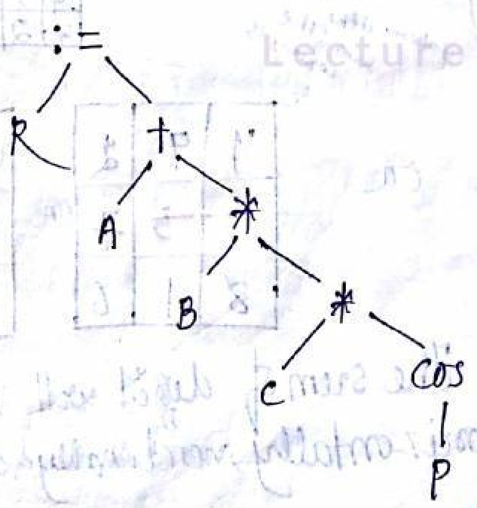
They are (Flying planes).
(They are) Flying planes

•) level of interaction among component

Eg:-
R = A + B * C * sin (P)



Eg:- R = A + B * C * sin(P)



→ otherwise wise the interaction will affect rest of the components and which makes the understanding difficult

•) Noise in the input

→ Sources of noise available in sound and image

→ These noise affects understanding.

(Module-1)

Understanding in Constraint Satisfaction Problem (CSP)

22/09/2016

CSP (constraint satisfaction problem) is a

→ heuristic techniqree used for problemsolving.

→ understanding is needed to reduced the complex. of CSP.

→ In CSP there are some constraint need to be followed in order to solve the problem.

→ Ex:- of some csp are :- Magic Square problem
4-Queen problem
Crypt arithmetic problem

Constraint means condition Limitation

•) Magic square problem

4	8	3
9	5	6
7	2	6

always it the sum of the column is 15

4	3	8
9	5	1
2	7	6

or

4	9	2
3	5	7
8	1	6

or

4	3	8
9	5	1
2	7	6

Constraint:-1:→ The sum of digit will be 15 (horizontally, vertically & diagonally)

2:→ Every digit (1-9) will be used exactly once

4- Queen Problem

	Q		
			Q
Q			
		Q	

Constraint:- NO 2 Q's should be the same row, same column or in same diagonal.

or

		Q	
Q			
			Q
	Q		

	Q		
		Q	
			Q
Q			

•) Crypt arithmatic problem

```
  S E N D
+ M O R E
---------
M O N E Y
```

O	M	R			E	N	D	R	S
0	1	2	3	4	5	6	7	8	9

```
 9 5 6 7
 1 0 8 5
---------
1 0 6 5 2
```

S = 9, E = 5, N = 6
D = 7
M = 1, O = 0, R = 8
E = 5
Y = 2

```
BASE      CROSS
BALL      ROADS
-----     ------
GAMES     DANGER
```

```
  C R O S S
  R O A D S
-----------
D A N G E R
```

```
  9 6 2 3 3
  6 2 5 1 3
-----------
1 5 8 7 4 6
```

C → 9
R → 6
O → 2
S → 3
A → 5
D → 1
E → 4
G → 7
N → 8

G A

0 1 2 3 4 5 6 7 8 9

BASE - 4

BALL - 4

GAMES - 5

according to B value let us we assume B as 5

as carry is 1 to value of G is 1

S+L = 1 0 + E

E+L = S

~~E+~~ L+L = 10 + ~~E~~

⇒ L = 5

B = 5, A = 0, M = 0 ×

B = 6, A = 2, M = 4 / ~~5~~ L = 5 E = 3

B = 7, A = 4, M = 8 / 9 S = 8

```
  6 2 8 3
  6 2 5 5
---------
1 2 5 3 8
G A M E S
```

or

```
  7 4 8 3
  7 4 5 5
---------
1 4 9 3 8
G A M E S
```

Artificial Intelligence

Topic:

NATURAL LANGUAGE PROCESSING

Contributed By:

Sankarsan Sahoo

new chapter

Natural Language Processing (NLP)

What is Natural language?

The language used for communication such as Hindi, English, French, Odia. is know as Natural language.

What is NLP?

It is the ability for a computer program to understand human speech & in response take correct action. is known as NLP

NLP can be further subdivided into following steps after spell checking.

- Morphological Analysis
- syntactic Analysis
- Semantic Analysis
- Discourse Integration
- Pragmatic Analysis

Morphological Analysis:-

It relates word construction from basic units called morphemes

ex:- Construction of friendly from (Friend & suffix (ly))

Syntactic Analysis:-

It relates how the words are put together to form grammatically correct sentences.

Scanned by CamScanner

Semantic Analysis :-

It is concerned with meanings of words and phrases & how they combined to from sentence meaning.

Dis course entegration :-

The meaning of an individual sentence may depend on sentence that precide it and may affect the meanings of the sentences that follow it. Eg:- paragraph how they are interelated with each other.

Pragmantic Analysis :-

It is the study of what is intainded by a speaker how it is or should be interpreted by the listener.

---

all of the above steps are sometimes performed at once or they may be perform in sequence.

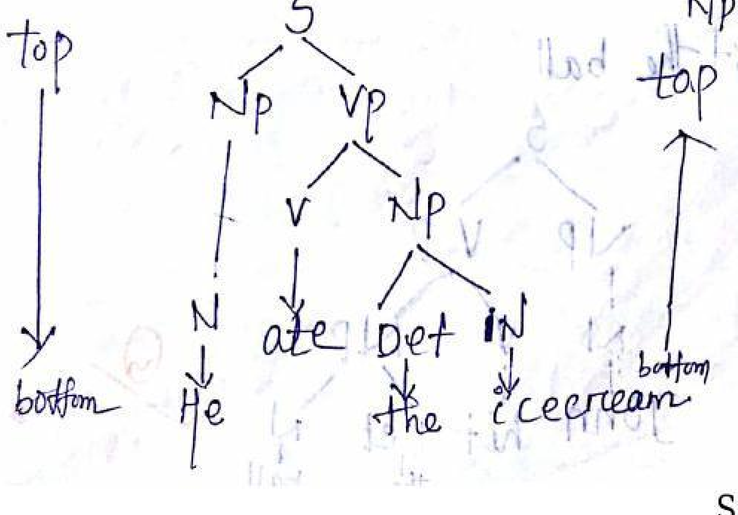
① SYNTACTIC PROCESSING 23/09/16

In this step the i/p sentence is converted into a hierchical structure which corresponds to the meaning of the sentence & this process is called parsing

every parsing is based on some grammer & the most common way to represent grammer is as a set of production rules. also called phrase structure Rule

Eg:- He ate the icecream

S → Np Vp
Vp → V Np
Np → N
Np → Det N



Phrase structure Rules.

Non-terminal symbol Terminal symbol

S → Np (Aux) VP

S → NP (conj) VP

NP → (Det) (Adj) N

VP → V (NP) (PP) (Adv)

PP → P (NP)

(Plz REMEMBER)

{ Det = ~~[illegible]~~ Determiner
{ PP = pre positional

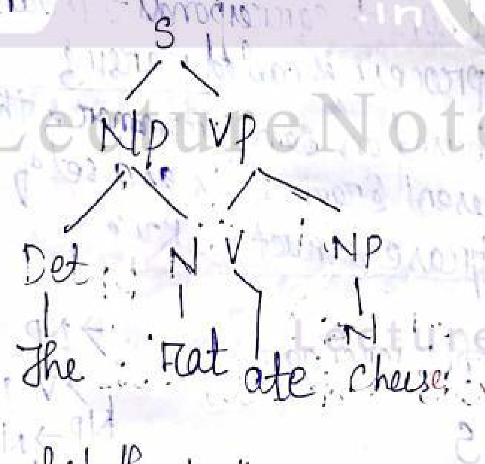
[(–) optional part "
Aux – Eg :- will
conj – Eg :- and, but, or…

Question :- Construct the parse tree for the following sentences.

- The rat ate cheese.
- John hit the ball
- The chef cooks the soup
- The boys ate soup at home.

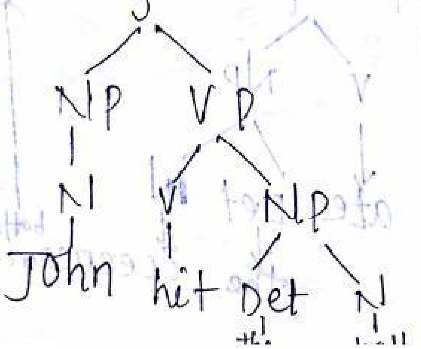
15/20

The rat ate cheese



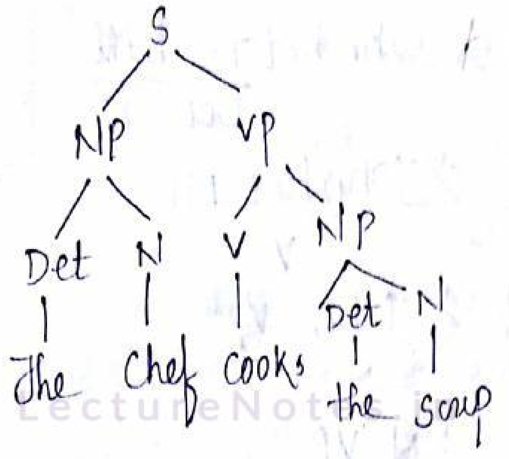
(5)

john hit the ball

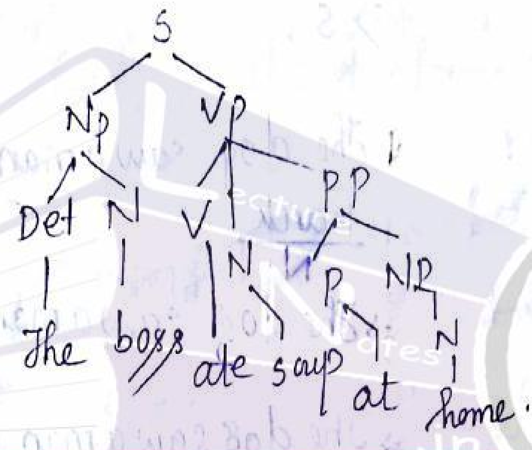


(5)

The chef cooks the soup



The boss ate soup at home



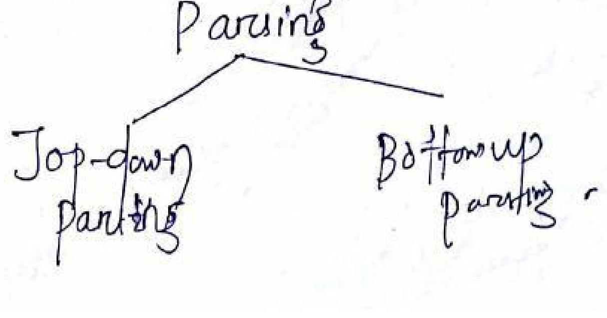
→ left side of a rule is called non terminal. and right side of the rule are called Terminals.

→ Every node of the parse tree Corresponds either to an i/p word or to a non terminal in our Grammer.

→ There are 2 type of parsing

i) Top-down parsing.

ii) Bottom-up parsing.



Top-down Parsing

Bottom up Parsing

* John hit $\underbrace{\text{the}}_{Det}$ $\underbrace{\text{ball}}_{N}$

⇒ John $\underbrace{\text{hit}}_{V}$ NP

⇒ $\underbrace{\text{John}}_{N}$ VP

⇒ N VP

⇒ NP VP

⇒ S

* The dog saw a man in $\underbrace{\text{a}}_{Det}$ $\underbrace{\text{Park}}_{N}$

⇒ The dog saw a man $\underbrace{\text{in}}_{P}$ NP

⇒ The dog saw $\underbrace{\text{a}}_{Det}$ $\underbrace{\text{man}}_{N}$ PP

⇒ The dog $\underbrace{\text{saw}}_{V}$ NP

⇒ $\underbrace{\text{The}}_{Det}$ $\underbrace{\text{dog}}_{N}$ VP

⇒ NP VP

⇒ S

① Topdown parsing

It begins with the start Symbol (s) and apply grammer rules (forward) until the Syombol of the terminals of the tree corresponds to the words of the sentence in process.

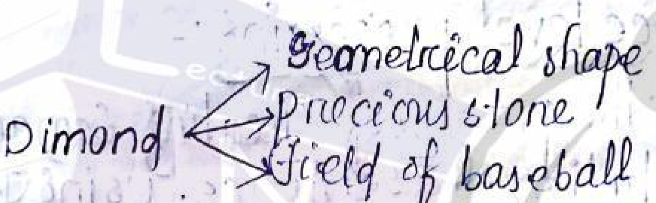
(ii) similar to forward Reasoning.

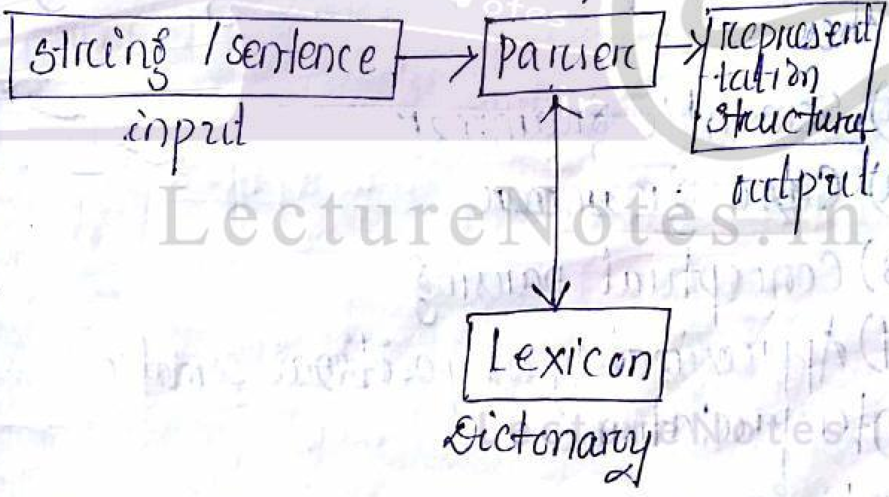
Bottom-up-parsing

① It begins with the sentence being parse and apply the grammer backward until a sin tree has been produced

(ii) similar to backward Reasoni

27/09/16 ② Semantic Analysis





I am going to pure

word	Type	Feature
a	article	- - - -
am	- - - - - - -	
go	verb	is.2s.3s

going

Semantic analysis must do following 2 corresponding things

i) Lexical processing

ii) Sentence level processing.

<i> Lexical Processing :->

-> It is a process of looking of the individual words (of the sentence) in a dictionary (called lexicon) and extract their meaning.

-> Lexical disambiguation is the process of determining the correct meaning of an individual word

◈ Sentence level processing :-

It is the process of creating semantic representation of a sentence. using following approches.

1) semantic grammer
2) Case grammer
3) Conceptual parsing
4) Approximately compositional semantic interpretation.

Semantic grammer :-

It combine syntantic, semantic & pragmatic knowledge in to a single set of Rules, in the form of grammer.

advantage:

- It reduces additional processing.
- Many ambiguities can be avoided.

•) Syntactic issues that don't affect semantic can be ignored.

disadvantages:-

•) The no. of Rules can become very large.

•) Parsing Process may be expensive(time consuming)

case grammer :-

It is a form of grammer in which the structure of sentence is analized interms of semantic case relationship.

eg:- Milli went to shop
subjective case — She bought a toy

Possesive case. — Milli's bag.
Her bag.

Advantage :

→ It can be applied in reverse by a parser to determine the syntax of the sentence.

Disadvantage

Result of parsing may no give complete; semanti representation.

Conceptual parsing

→) It is used to find both structure and meaning of a sentence in one step. like semantic grammer

→) It is driven by a dictonary that describes meaning of words as conceptual dependency structured (CD)

•

Approxmately compositional semantic interpretation (ACSI).

It is in which semantic processing is applie to the result of performing a syntactic pa

③ DISCOURSE AND PRAGMATIC ANALYSIS

This steps are used to find rel^nship among multiple sentences spoken by the speaker and to extract exact meaning of the speaker.

1. Identical entities

Bill had a red balcon
John wanted it.

Imanu imp The word "it" should be identified as refering to the red balloon this type of reference are called anaphoric References or anaphora / anaphora resolution

2. Parts of entities

Sue opened the book. she had just purchased
The titile page was torn.

3. Casual chains

There was a big snow storm yesterday
The schools were closed today.

Statistical NLP

It is requeired to process long sentences which requeire large analysis to get it's meaning

spellchecking

It is one of the basic word for language processing. ~~It is~~ preprocessing task used in verities of task such as word processing character and text recognition system, speech recognition system & generation.

eg:. Henery sar on the box.

There are 3 types/cause of spell checking error.

i) Insertion of extra letter while typing.
eg: schoool

ii) Deletion or missing of a letter.
eg: scho_l

iii) Substitution of wrong letter in place of correct one.
eg: skool

Error can be classified ① typo-graphical errors:-
These are caused due to mistakes commited while typing.

② Orthographic error :-
due to lack of comprehension on the concerned language.
eg: writting
wellcome

③ Phonetic errors :- due to poor cognition of the listener phonetic errors are occured
eg: rough (speaker) → ruff (listener).

29/09/16

Spell checking Technique

misspell-
spelling

i) Non-word error detection

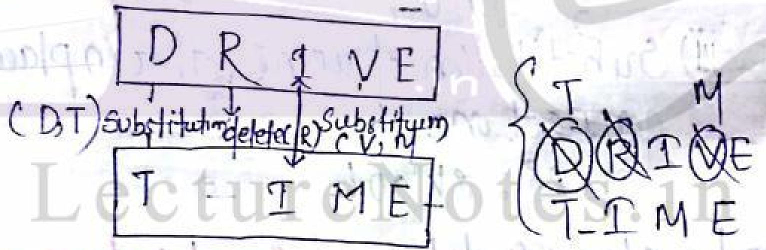
It involves detection of misspelled word & Correct it using dictionary

ii) isolated word error Correction:-

It Focuses on the correction of an isolated non-word by Finding it's nearest & meaningful word & makes an attempt to rectify it.

eg: minimum edit distance technique:

It uses min^m no. of edit operation (insertion, deletion, substitution) of single character to transform the misspelled word to the correct one.



→ In the above example the min^m no. of edit operation
→ It is also call Levenshtein distance.

iii) Context dependent error detection & correction

This method in add^n to detect errors try to find, wether the corelated word in feeds to the context or not.

Peace comes From within
or
Piece comes From with in

It uses both traditional and statistical NLP to extract the correctone.

SOUNDEX ALGORITHM

Imp

It is a simple phonetic based spell checker. It uses a code to check for the closest word

Phonetic based spell checker code

Remember

Convert in to (CAPITAL) if small word is given

U have to see where A, E, I, O, U, E H, W, Y like letter present they deleted.

Letter	substitute with no.
BFPV	1
QCGJKSXZ	2
DT	3
L	4
MN	5
R	6

Ex: ① aeroplane

= AEROPLANE

= Ⓐ~~E~~R~~O~~PL~~A~~N~~E~~

= ARPLN

= $\boxed{A614}5$

= A614

Ex: ② torn

TORN

Ⓣ~~O~~RN

TRN

$\boxed{T650}$

Ex: ③ horn

HORN

H~~O~~RN = H65

$\boxed{H650}$

Ex: ④ worn :- $\boxed{W650}$

The codes for a word consists of it's 1st letter followed by 3 no. that encode the remaining consonant.

Steps of SOUNDEX ALGORITHM

1. Remove all punctuation Mark & capitalized.

2) The letters of a given word. written the 1st letter of the word.

3) Remove any accurence of letter {AEIOUHW

→ Except the 1st one.

4) Replace the letter (other than the 1st) by the as shown in the table.

5) If 2 or more adjacent letter not separated by viowels have the same numeric value then written only one of them.

6) Return the 1st 4 characters, if they are less than 4 characters then the vacent will be padded with zero.

ex-5 Networking.

N~~E~~T~~WO~~RK~~I~~NG

NTRKNG

N365 2

N365.

ex-6 Backcheeking.

B~~A~~CKC~~HE~~CK~~I~~NG

BCKCCKNG

B225 2

B225

—o—

kn=2

N5

N36

Artificial Intelligence

Topic:

LEARNING

Contributed By:

Sankarsan Sahoo

1od.3 newchapter
=

रटा - muggingup

LEARNING

It is the process of accureing knowledge through study, practicing or through experence

eg:- The more v ride a bicycle or tennis the better you get.

Mechine can't be called intelligent until they are able to learn to do new things and to addopt to a new situation

Learning – i) Rote learning
ii) By taking advice.

i) Rotelearning :-

→ It is the most basic learning activities that involves simple storing of computed information.

→ It is a memorisation techniqee based on repetation.

simple example of wrote learning are : Alphabate and no.

Advantages:-

→ It allows the program to perform better in future

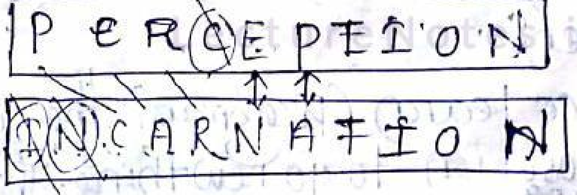
→ avoid recomputation.

→ It saves time.

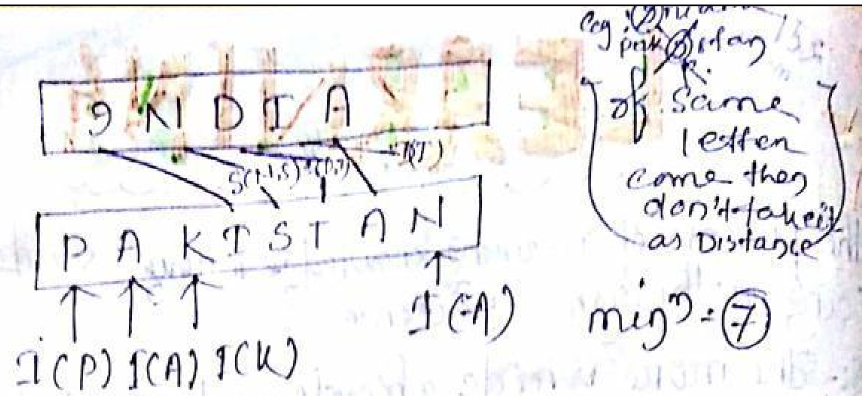
30.09.2016

Find Minimum Edit Distance

① Perception
Incarnation



minm no. of edit distance = 7
Substitutes (P, A)
Substitutes (E, N)
Delete D (C)
Substitudes (E, A)
Substitutes (P, C)
Insertion (I)
Insertion (N)



Find the codes for India & Pakistan using Soundex Algo

I N D I A   P ~~A~~ K ~~I~~ S T ~~A~~ N

I N D ~~I~~ ~~A~~ $\boxed{P\ 2\ 2\ 3}$ 5

I 5 3 0   P 2 2 3

S I N S T I   S W A T I

= S ~~I~~ ~~N~~ ~~A~~ S T ~~I~~   S ~~W~~ ~~A~~ T ~~I~~

= S 2 3 0   S 3 0 0

⑪ Learning by taking Advice :-

→ It is a simple form of learning. Suppose a Programmer writes a set of instructions to instruct the computer what to do; then the programmer is a teacher & the computer is a student.

→ Once learn (program) the system will be in position to do new thing. This type of learning is learning by taking Advice.

Induction: Learning by Example (Induced/Concept learning)

The system try to induce a general rule from a set of observed instances

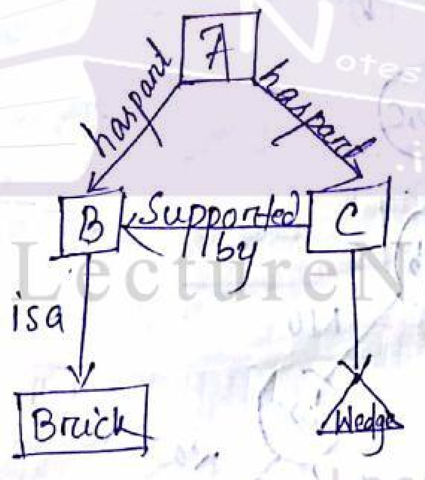
The learning method Extract rules & patterns
→ out of massive-data set (data mining)
→ It is also called concept learning.
→ There are 3-techniqee used for concept learning.

- Winston's learning program.
- Version space.
- Decission Trees.

Winstons learning Program:-

It operates simple block domains & the goals is to construct representation of defination of concept using blocks.

Block Domains	Meaning
Rectangleular □	Brick $\boxed{B}$
Triangular △	Wedge $\boxed{C}$

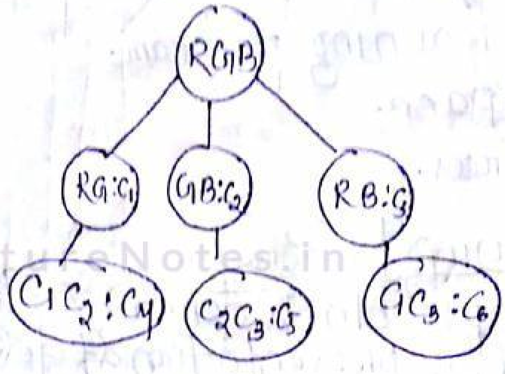


Semantic net to represent concept House

Version Space:-

It is a hierachical representation of knowledge that keeps track of all the useful information supplied by a sequence of learning Example without rembering any of the Example.

R = Red
G = Green
B = Blue.

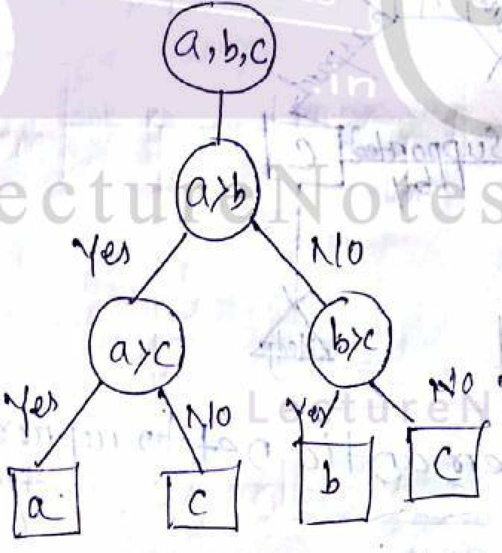


3. Decision tree :

-> It is a powerful tool classification & prediction

-> It represent rules that are easily express and use to retrieve useful information.

-> There are 2 types of nodes used in decession tree.



4/10/16

Explanation Based Learning (EBL)

An EBL system attempts to learn from a single eg. x by explaining why x is an example of the target concept.

The explanation is then generalised and the system performance is improved.

output eg:- $Lender(x,Y) \rightarrow relative(x,Y) \wedge Rich(Y)$

$relative(x,Y) \longleftarrow uncle(Y,x)$

$rich(Y) \longleftarrow Ceo(Y,B) \wedge Bank(B)$

$rich(Y) \longleftarrow own(Y,H) \wedge house(H)$

input

Learning by DISCOVERY

It is a restricted from of learning in which one entity accquires knowledge without the help of a teacher.

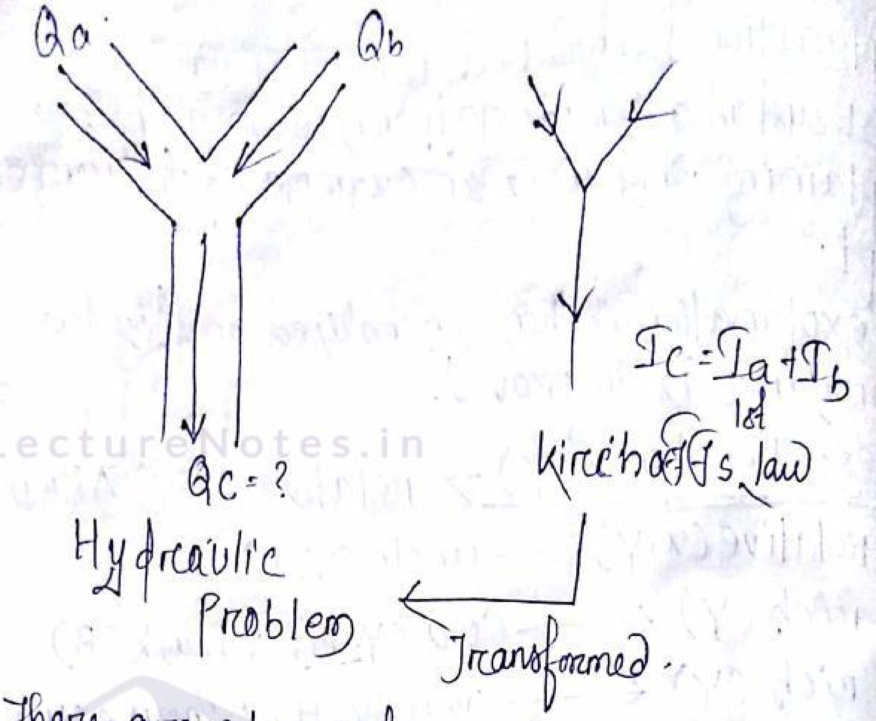
→ Discovery can be up 3 types

- Theory-Driven Discovery. eg:- (Mathmatical hypothesis)
- Data-Driven Discovery. eg:(queering data base)
- Clustering.

Clustering:-
It is a way to form natural groupings or clusters that exists for the objects.

Analogy of Learning

It is a kind of learning in which new knowledge can be accuired (about an input entity) by transforming it from a known similar entity.

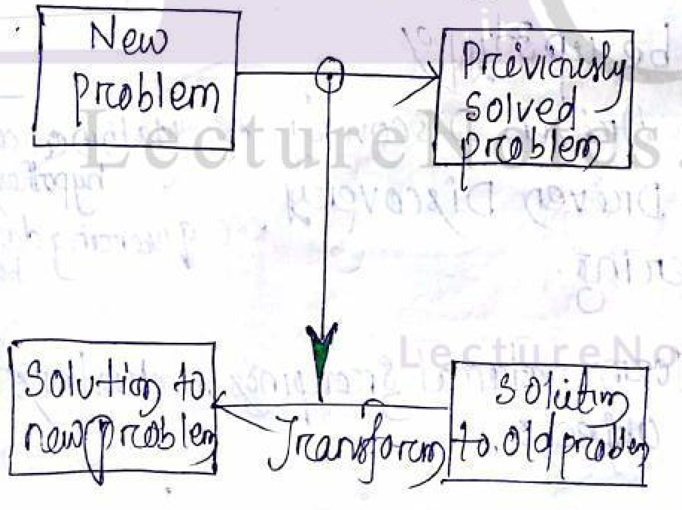


There are 2-types of analogy

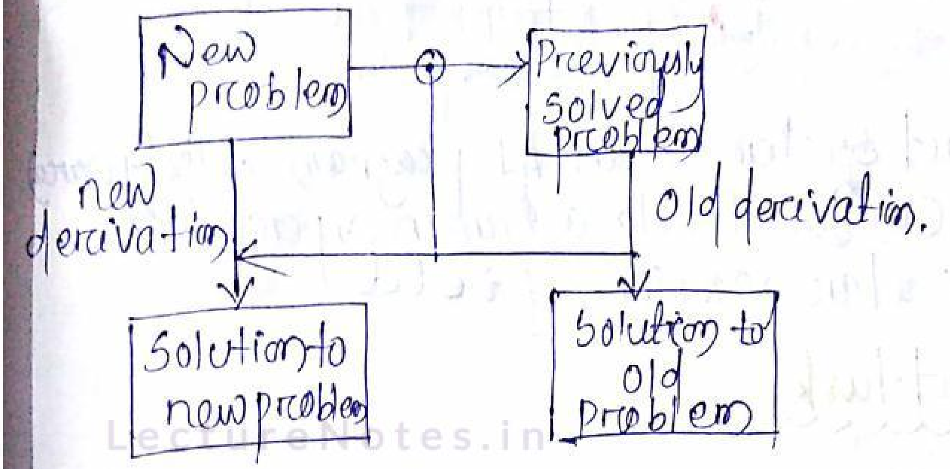
i) Transformational Analogy.

2) Derivational Analogy.

1. Transformational Analogy:-



2. Derivational Analogy :



Difference

Transformational Analogy	Derivational Analogy
In 1st approach (TA) doesn't look how the old problem was solved;	but in case of 2nd approach Checks how the old problem was solved using the history

Neural net learning / genetic learning:-

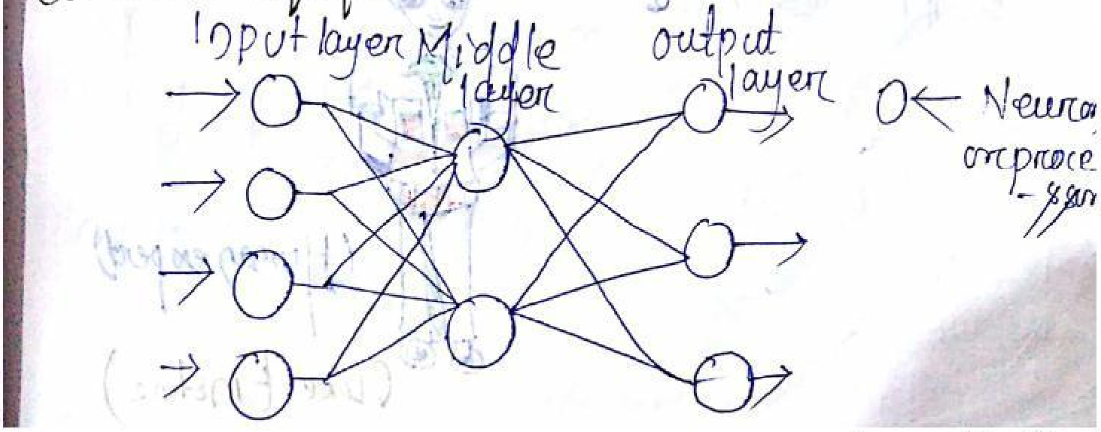
These are biological inspired learning technique used to mimic (copy) animal learning at neural level

Neural Net

It is an inter linking of neurons in brain that activate a thought or learning process among animals.

It is also called ANN (Artificial Neural Network)

Architecture of ANN



Artificial Intelligence

Topic:

EXPERT SYSTEM

Contributed By:

Sankarsan Sahoo

20/10/2016

new chapter

defn & Ex v.v.v.imp

Expert System

Expert system is an AI program or (software) that can substitute a human expert in a particular domain / field.

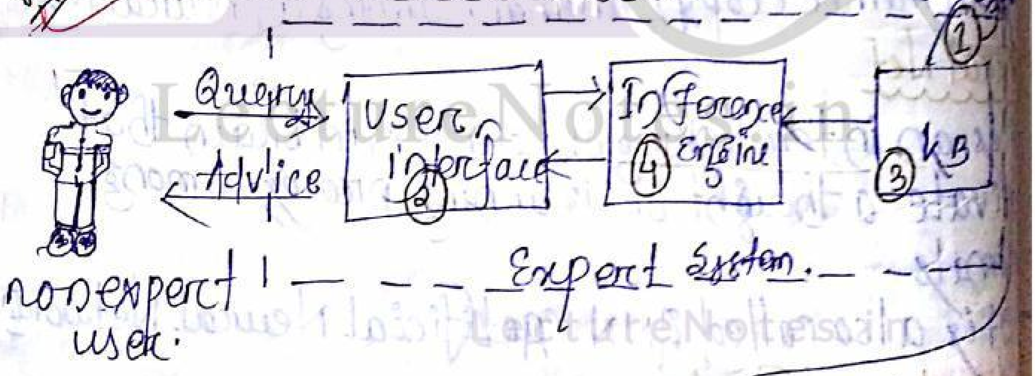
Expert task

1. Engineering
   - * Design
   - * Fault finding.
   - * Manufacturing.
2. Scientific Analysis.
3. Medical diagnosis,
4. Financial Analysis

(query opp. of answer)

COMPONENT OF EXPERT SYSTEM

imp



K'B Knowledgebase

① An experts system knowledge obtained from expert sources
→ and coded in a form suitable for the system to use
in it's inference and reasoning process.

② It allows a non expert user to queary the
→ expert system and receive advice.

③ It is a collect of facts and rules and it is created
→ for information provided by human expert.

④ It uses the user queary to search the knowledge
→ base and then provides and answer or some advice
to the non expert user.

Examples of expert system:-

1. DENDRAL:

→ Developed at stanford university in late 1960s
→ used to determine the structure of chemical compoun

2. MYCIN :-

It was used to dignose infectious blood deases and
determine a recomanded least of therapies of
patient.

3. Prospector:-

It was used to assist geologest in the discovery
of mineral deposit.

4. RI(XCON):-

It is a system used to select and configure
component of a complexcomputer system.

Features of Expert System

① High performance

They should perform at the level of human expe

②. Adequated Response time

It is the ability to respond in Reasonable able time

③ Reliability

They must be reliable and should not crash.

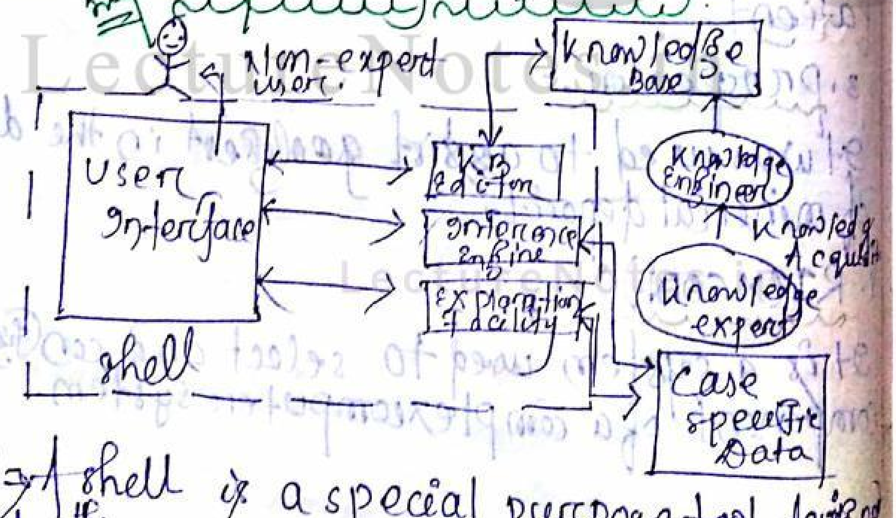
④ Understandable

It should justify it's conclusion in the same way a human expert explains.

⑤ Updated

It aquire new knowledge and modify old knowledge.

Expert system shell



Shell :→ A shell is a special purpose tool designed based on the requirements of a particular applicati

→ A shell is nothing but an expert system without knowledge base

→ A shell provides the developer with knowledge

Acquisition, inference engine, user interface & explanation facility.

Explanation facility :-

It is a part of expert system that allows a user or
→ decission maker to understand how the expert system arrived at certain conclusion or results.

→ By looking an explanation the knowledge engineer can determine how the system is behaving how the rules & data are interacting.

Knowledge Acquisition :-

→ It is the process of adding new knowledge to a knowledge base and refining or improving knowledge that was previously acquired.

→ It may consists of facts, rules, concepts, Procedures heuristics, formulas, relationship, statistic or other useful information

eg EMYCIN and IMYCIN
(Empty) (Student information)

CLIP :- It is a public domain expert system used by NASA.

IESS :- Successor of CLIP.

— 0 —

Artificial Intelligence

Topic:

GAME PLAYING

Contributed By:

Sankarsan Sahoo

Mod-② "newchapter" Imp chapter

"Game Playing"

21/10/16

Importanceof Gameplaying

→Game playing can be used for machine learning due to following region.

→The state of game is easy to represent

→The rules of the game are limited & precise

→They provide a structural task so success or failure canbe easily measured.

→Human experts can easily explain the logic behind game playing moves.

→Game simulate a Real life situation.

Game playing is a kind of Searching Problem

In game playing while we are doing our best to find the sol^n our opponent (adversary) oppositeparty opponent is alsotry to bitus i.e why in AI game playing is also called Adverserial search which is a kind of huristic search

Componants of Gameplaying Search

The game playing can be formally defined as a kind of searchproblem with the following components.

1. Initialstate of the game:-

It is used to specify the starting cond^n. of the

Scanned by CamScanner

game.

2. <u>Plausible move general:-</u>

move which is valid

It is used to expren or generate Only selected moves

3. <u>Static Evaluation Function:-</u>

It is based on heuristic it is generated for each & every move i.e made.

4. <u>Terminal test defining end of game:-</u>

It is used to test weather a particular node is goal node or not.

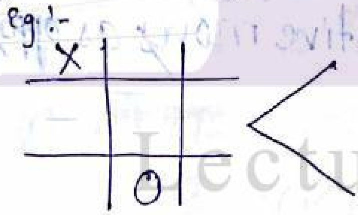
5. <u>Goal state:-</u>

It is used specify end of a game.

6. <u>Path cost or utility Function:-</u>

It is the sum of all the cost from starting state to the goal state and it is used to compaire alternative game solving strategies.

Eg:-

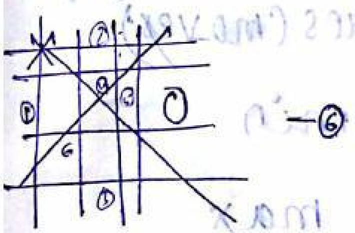


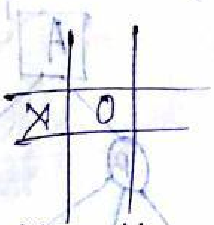
X has 6 possible win paths.

O has 5 possible win paths.

Heuristic/static Function $\boxed{E(n) = M(n) - O(n)}$

n - a particular state.





X = 4

O = 5 4 - 6 = -2

There are 3 important game playing techn
Such as.

1. minmax search Procedure.
2. Alpha-Beta cutoff
3. Iterative Deepening.

1. Minimax Search Procedure

2 player game

Characteristics of minimax search Procedure are.

1. 2 persons :-

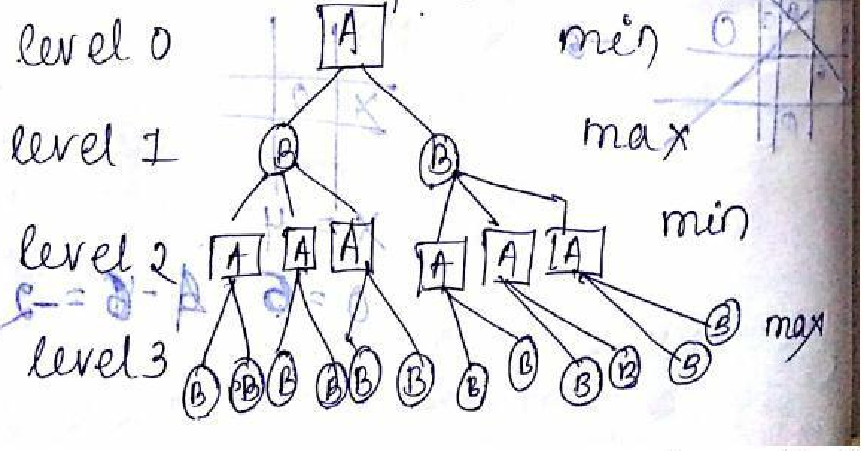
It is a kind of game that is played bet2 2 oponent (no 3rd oponent, no team playing)

2. Turn taking :-

The player gets alternative moves as oppo. to each other.

3. Zero-sum

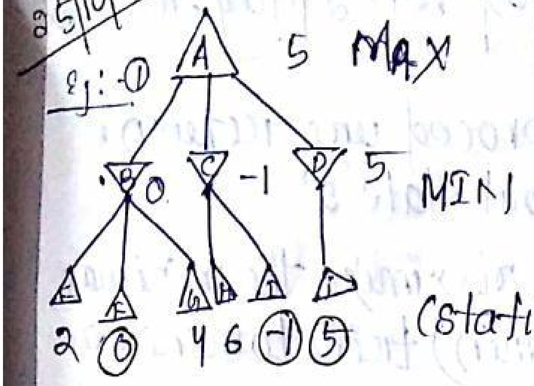
When 1 player wins other looses. The possible game state can be organised into tree or graph with the nodes linked by arcs (moves).



25/10/16

Remember
* For maximize → △
* For minimize → ▽

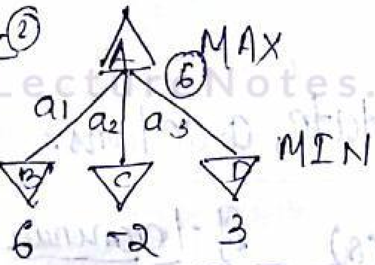
eg:-①



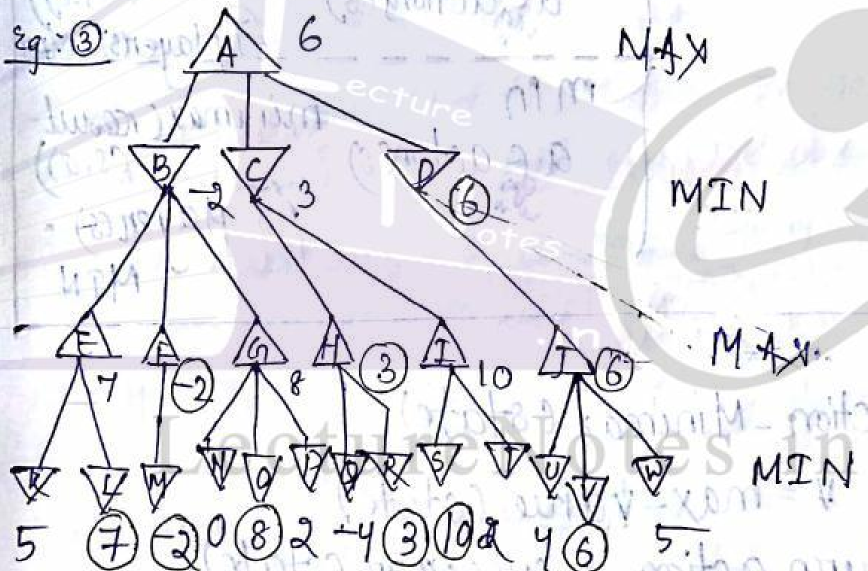
(static eval.) (function value)

5 - minimax value.

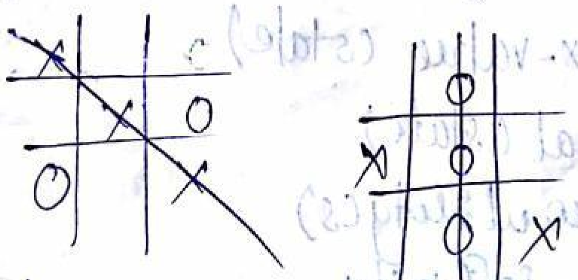
eg-②



eg:③



Forces remember: For tic-tac-toe game.



Max win = +∞ (or +15)

Min win = -∞ (or -15)

→ minimax game is played betⁿ 2 players max and min

→ The minimax search procedure records minimaxvalue (for a goal state 's'

→ Player max(you) want to maximize the minimax value but ur opponent(min) tries to minimize the minimax(s) value.

Algo: where s - goal state, a- actions.

$$\text{minimax}(s) = \begin{cases} \text{utility}(s) & \text{if terminal}(s) \\ \max_{a \in \text{actions}(s)} \text{minimax}(\text{result}(s,a)) & \text{if player}(s) = \text{MAX} \\ \min_{a \in \text{actions}(s)} \text{minimax}(\text{result}(s,a)) & \text{if player}(s) = \text{MIN} \end{cases}$$

```
1. Function -Minimax (state)
        v = max-value (state)
   return action in successor (state)
     with value v.

2. Function max-value (state)
        if terminal (state)
            return utility (s)
        v = -infinity
        for a,s in successor (state) do
            v = max(v, min-value(s))
```

```
return v
3. (Fuction min-value (state)
    (If terminal (state)
      return utility(s)
        v = + infinity
    (For a, s in successor (state) do
        v = min (v, max-value (s))
      return v,
```

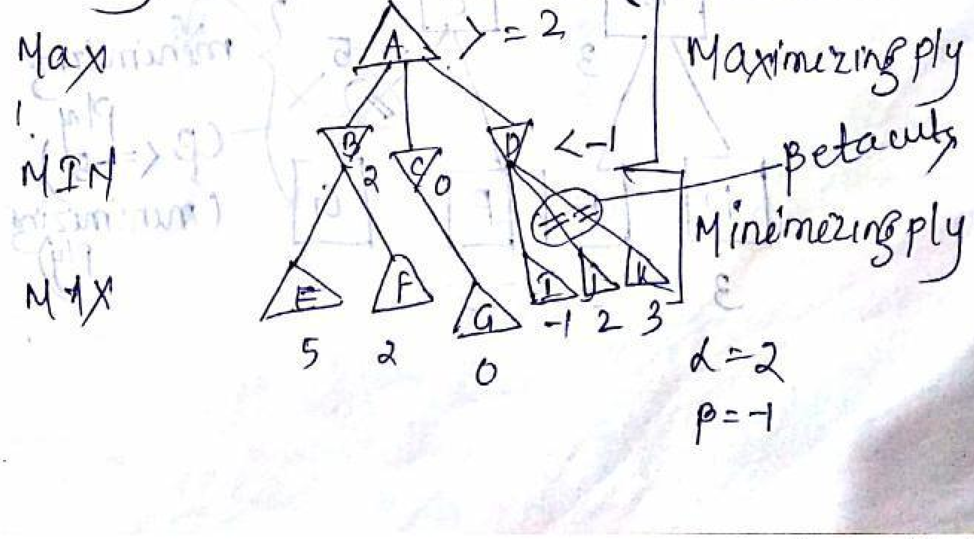
The min max search procedure is a depth 1st, →depth limit search procedure

→The idea is to start at the current position & use the plausible move generatore to generate the set of plausible successor state.

→Now static evaluation (Fun) is applied to those states and the best one is choosen (for (Max or Min)

→After doing so, we can back that value up to the starting position to represent our evaluation of it

→ Our goal is to maximize the value of the static evaluation (Fun) of the next board position & to find utility value for the wining state.



α = 2
β = -1

Difficulties In Minimax Search

It is a depth 1st search in which 1 path is
→ explored as per as time allows and the state Evaluation (fun) is computed at the last step of the path; then the value can be passed of the path 1 level at a time.

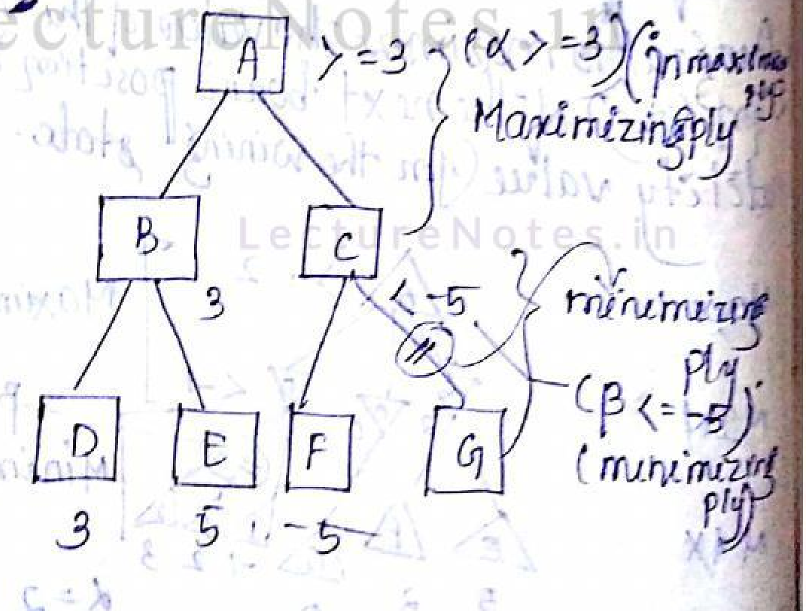
→ It may cause unnecessary expansion of bran of minimax search tree and they require more times

→ So the efficiency of minimax search can be impr by a branch and bound technique called αβ pruning (Alpha-Beta pruning / cutoff)

Alpha beta pruning

{Alpha cut = maxi
(Beta cut = minimi

In minimax search if partial expansion or branching, that are worse than no sol, then they can be cancell early.



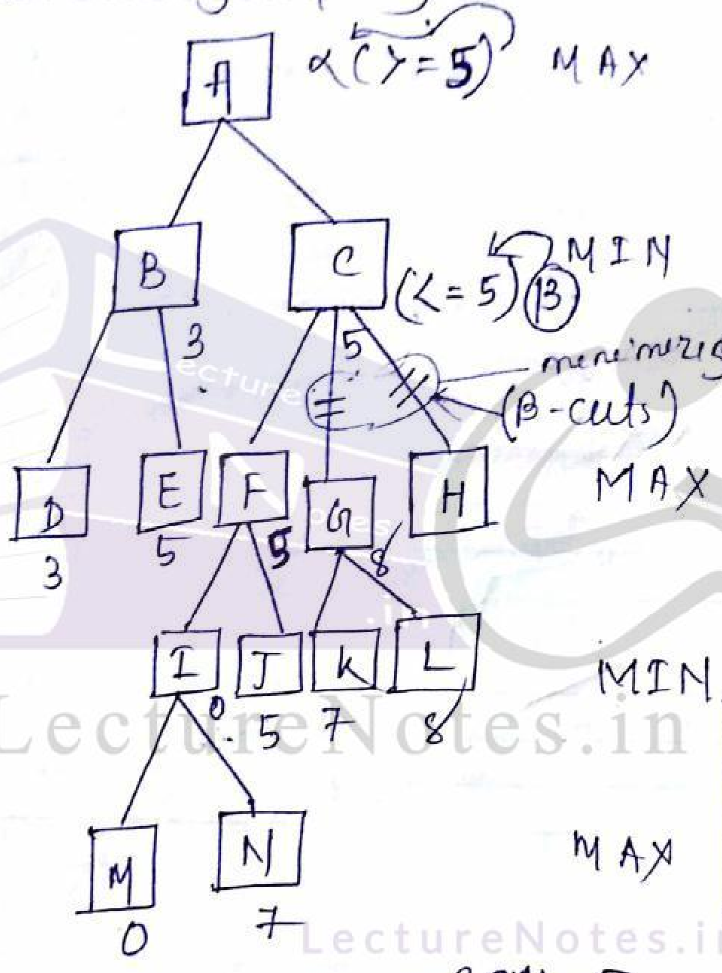
→ To do so (alpha-Beta cut ops) 2 threshold values 'α & β' are used

•) Alpha (α)

It is used to represent a lower bound on the values that a maximizing node may assigned.

•) Beta (β)

It is used to represent an upper bound on the value that a minimizing node may assigned.

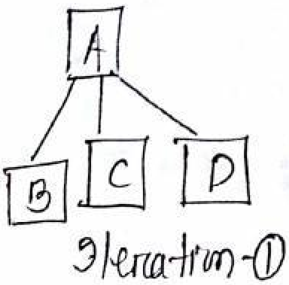


Remember
in max >
in min <

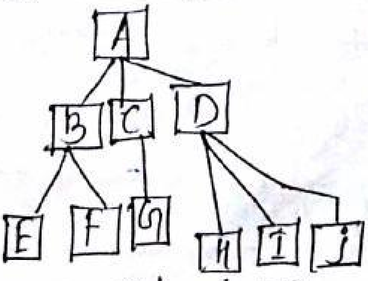
{α>=5
β<=5

3) Iterativ Deepening Search

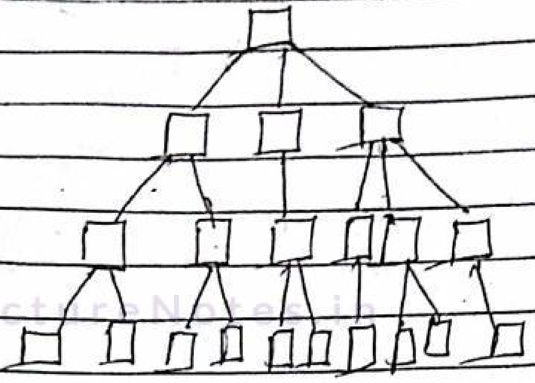
28/10/16



Iteration-①



Iteration-2



Iteration-3 (Iterative Deepening)

The name iterative deepening derives from the fact that on each iteration the tree is searched one level deeper like BFS.

→ This process seems wasteful but it is required when the game playing programs are subject to time constraint

→ An algorithm called depth first iterative deepening (DFID), combines the best aspects of DFS and BFS

Algorithm: DFID

There are three steps:

1. ~~Search~~ Set SEARCH_DEPTH=1

2- Conduct a depth first search to a depth of search depth. If solution path is found then return it.

3- Otherwise increment search depth by 1 and go to step 2.

Advantages: [illegible]

→ DFID avoids the problem of choosing cut offs without sacrificing efficiency
→ IFD is an optimal algorithm in terms of space and time for uninformed search.

Iterative deepening can be also be useful in improving the performance of the search algorithm.

Algorithm: Iterative Deepening A*

1. Set THRESHOLD = the static evaluation value of the start state.
2. Conduct a depth first search pruning any branch when the total cost function exceeds threshold. If a solution path is found during the search return it
3. Otherwise increment threshold by the min. amount it was exceeded during the previous step and then go to step 2.

Disadvantages:

1. It requires large amount of memory to maintain the search node left.

Artificial Intelligence

Topic:

PLANNING

Contributed By:

Sankarsan Sahoo

Chapter Important

"PLANNING"

Q. What is planning?

It refers to the process of computing several steps of a problem solving procedure before executing any of them.

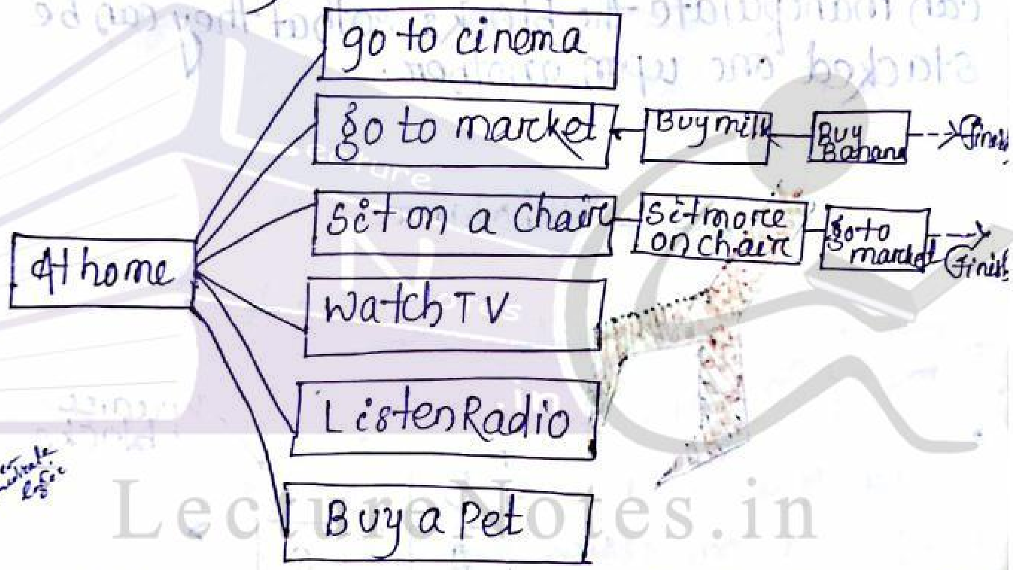
2 basic plannings are

- Frame problem
- Decomposable problem

1/11/2016

Differentiate Problem solving and Planning :-

Ex:- Task: Buy Milk, banana and chicken



- planning and problem solving method both can often solved same sort of problem.
- Planning is more powerful because of the representation and method's used.
- states goals and action's are decomposed into set of sentences (using FOPN)
- subgoals can be planned independently reducing the the complexity of planning problem.
- search often proceeds through plan-space rather than states space by considering only relevant actions.

In this chapter, we will discourse the following planning technique.

Planning techniques are.

① → Goal stack Planning

② → Non linear planning

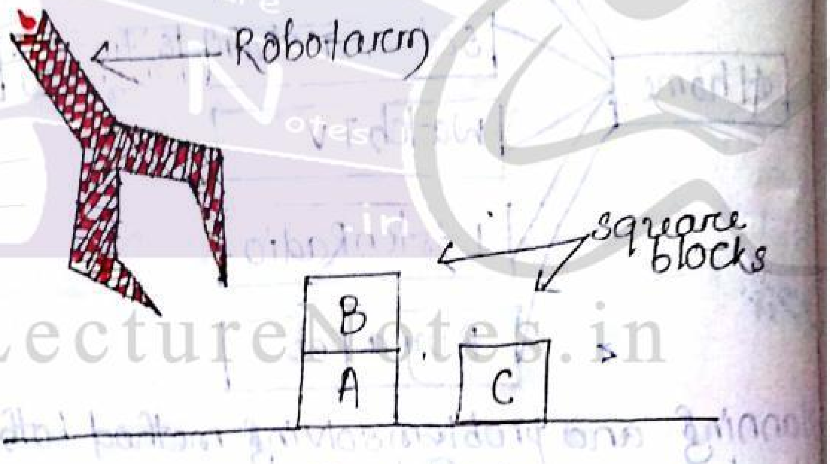
③ → Hierarchical planning.

④ → Reactive system

⑤ → Other planning technique

Blocks World Problem: An Example Domain

Given a no. of square blocks and a robot arm which can manipulate the blocks so that they can be stacked one upon another.



Actions

Unstack(A, B): pick up block A from the top of B assuming the arm is empty.

stack(A, B): place the block A on the top of B.

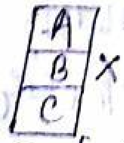
pickup(A): Pick up block A from the surface of the arm is empty.

putdown(A): put down A on the surface and now the arm is empty.

condition

The robot arm can hold only one block at a time.

→ A block can have at most one an other block directly on top of it.



→ In order to specify ~~both~~ the condⁿ we need following predicates (Restricted)

Predicates

1. on (A,B) – Block A is on block B
2. ontable(A) – Block A is on the table.
3. clear(A) – Top of block A is clear
4. Holding (A) – The robot arm is now holding block A.
5. Armempty(A) – The arm is now empty.

Que :- write FOPL expressions for following statement.

1. If the arm is holding any-thing then it is not empty.

$$\boxed{[\exists x : Holding(x)] \rightarrow \sim Armempty.}$$

2. If a block is on the table, then it is also not on another block.

$$\boxed{[\forall x : ontable(x)] \rightarrow \neg \exists y : on(x,y)}$$

3. Any block with no block on it, then it is clear.

$$\boxed{\forall x : \neg \exists y : on(y,x) \rightarrow Clear(x)}$$

or $[\forall y : [\neg \exists x : on(x,y)] \rightarrow Clear(y)]$

Component OF planning System <span>03/11/16</span>

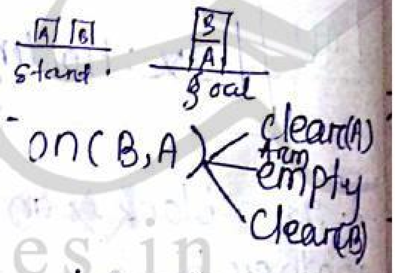
Components may be consider by performing each of the following fun?

1) Choose the best rule to apply next
2) Apply the chosen rule to compute new problem state.
3) Detect when a sol? has been found
4) Detects Dead-ends so that they can be cancelled
5) Detect when an almost correct sol? has been found and used special technique to make it totally correct.

Dr. Jyoti Jamankar

Goal stack planning :-

Eg :-



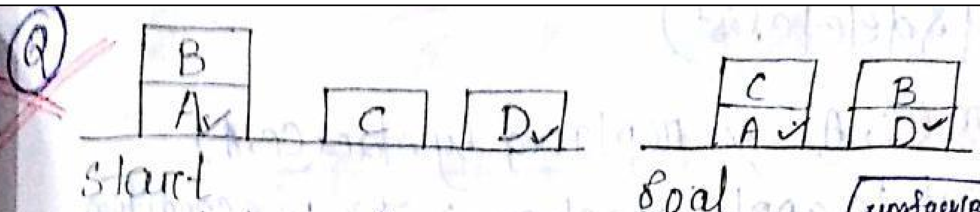
→ It's one of the earliest technique for solving compound goals using goal stacks.

→ In this method a single stack is used that contain's both goals and operators

Clear(A)
Clear(B)
Arm empty
Clear(A) ∧ Clear(B) ∧ Arm empty
Stack(B,A)

→ Goals may be subproblem or main problem and the operators are the actions needed to perform problem and subproblems

Q)



(unstack(B,A) means B is on A)

according to human being

•) unstack(B,A)

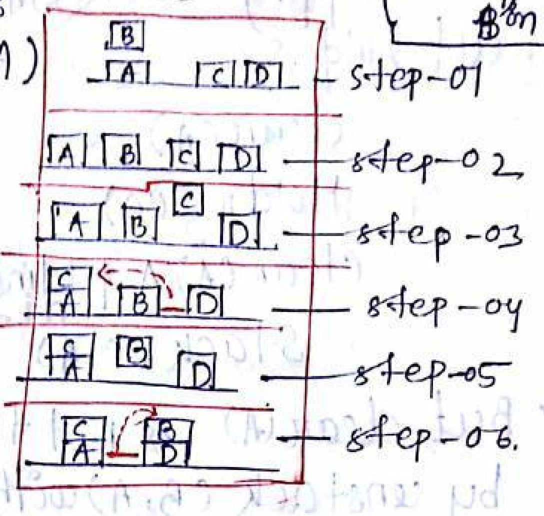
•) Put down(B)

•) Pick up(C)

•) stack(C,A)

•) pick up(B)

•) stack(B,D)



according to machine

start on(B,A)
∧ ontable(C)
∧ ontable(D)
∧ ontable(A)

goal: on(C,A) ∧ on(B,D)
ontable(A) ∧
ontable(D)

→ 2 of the subproblem ontable(A) or ontable(D) are already true in the initial state so the remaining two need to be solved

→ We may assume to goal stack depending on the order to deal subproblem

Goal stack 1

on(C,A)
on(B,D)
ontable(A) ∧ ontable(D)
∧ on(B,D) ∧ on(B,D)

(or)

Goal stack 2

on(B,D)
on(C,A)
on(B,D) ∧ on(C,A) ∧
ontable(A) ∧
ontable(B)

Plz remember

→ In Order to solve the above problem solver (eg: STRIP) used a database that describes the current situations and a set of operators (precond?

add & delete list)

→ on (C, A) is replaced by stack (C, A)

→ But to apply stack (C, A) it's precondition must hold i.e

Precondition
- clear (A)
- Holding (C)
- Clear (A) ∧ Holding (C)
- Stack (C, A)

→ But clear (A) is not true so replace it by unstack (B, A) with precondition.

- on (B, A)
- Clear (B)
- Arm empty
- on (B, A) ∧ Clear (B) ∧ Arm empty

→ So now on (B, A) is on the top of goal stack, which is true so pop it up from the top.

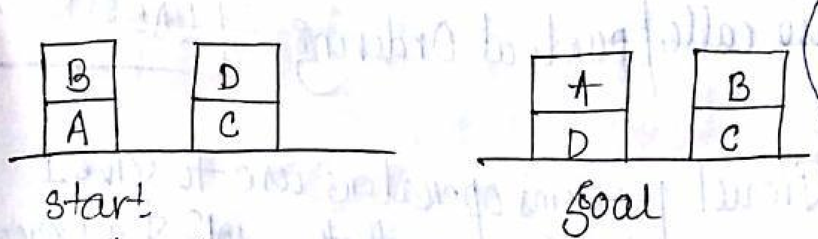
→ Similarly clear (B) and Armempty and combi goals can be poped up off from the top

→ Final Goal stack contain

1. unstack (B, A)
2. stack (B, D)
3. pickup (C)
4. stack (C, A)

4|11|2016

Question :-



How (A,D)
3 conditions
→ on table(D) ^
→ clear(A) ^
→ arm empty

Solve the above using STRIPS planning (Goal-stack planning)

unstack (B, A)
put down(B)
unstack (D, C)
put down (D)
PICK UP (A)
stack (A, D)
PICKUP (B)
stack (B, C)

A B C D

correct-

unstack (B, A)
putdown (B)
unstack (D, C)
putdown (D)
Pick up (A)
stack (A, D)
Pick up (B)
Stack (B, C)

Properties of planning Library

* Soundness:- A planning algorithm is sound if all sol's found are legal plans
* Completeness:- A planning algorithm is complete if the sol can be found whenever one actually exists
* optimality:- The planning algorithm is optimal if the order in which the sol's are found is consistent.

Advantages of Goal stack planning :- (also called linear planning)

→ Reduce search space since goals are solved one at a time in order
→ It is sound

Disadvantages:

→ It is time consuming.
→ It is incomplete.

Non Linear Planning/(TWEAK)

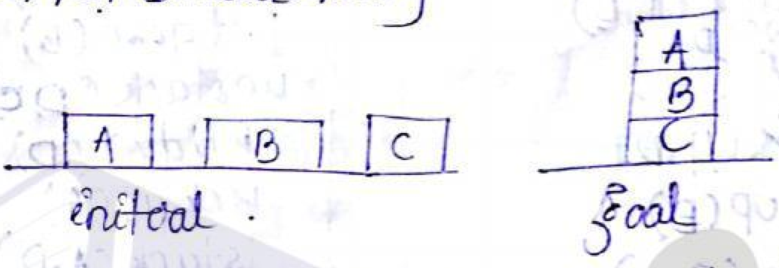
Remember → and hierarchical planning (ABSTRIPS)

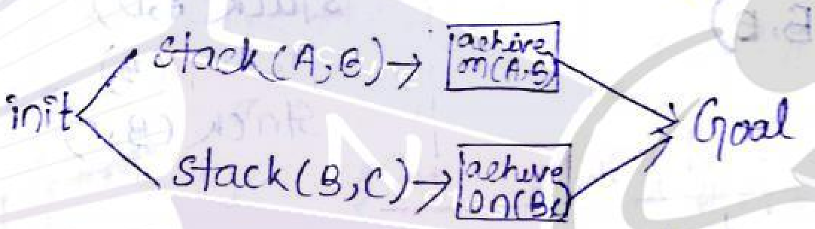
→ it is also called partial Ordering planning

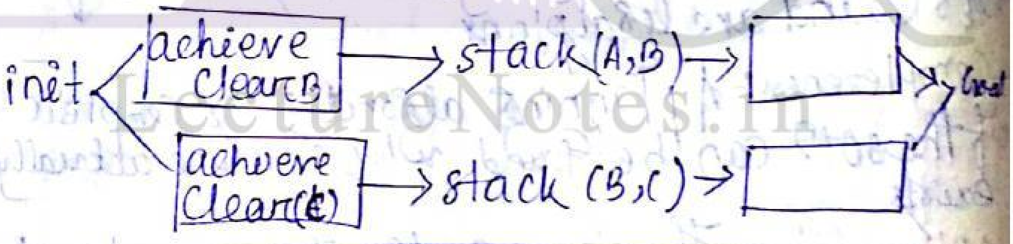
→ For difficult problems operators use to solve 1 Subproblem may interfere with the soln of a previous solved problem.
(involved)

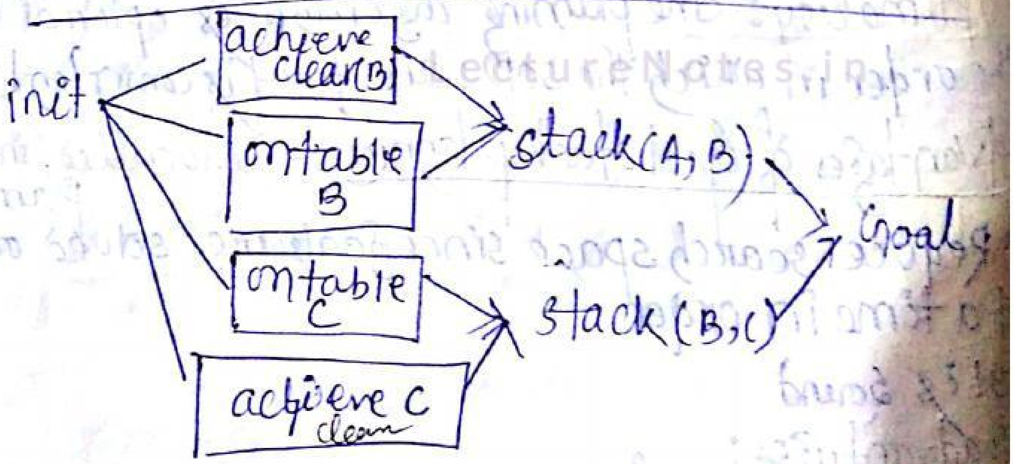
→ It is non linear planning because subproblems are planned simultaniously.

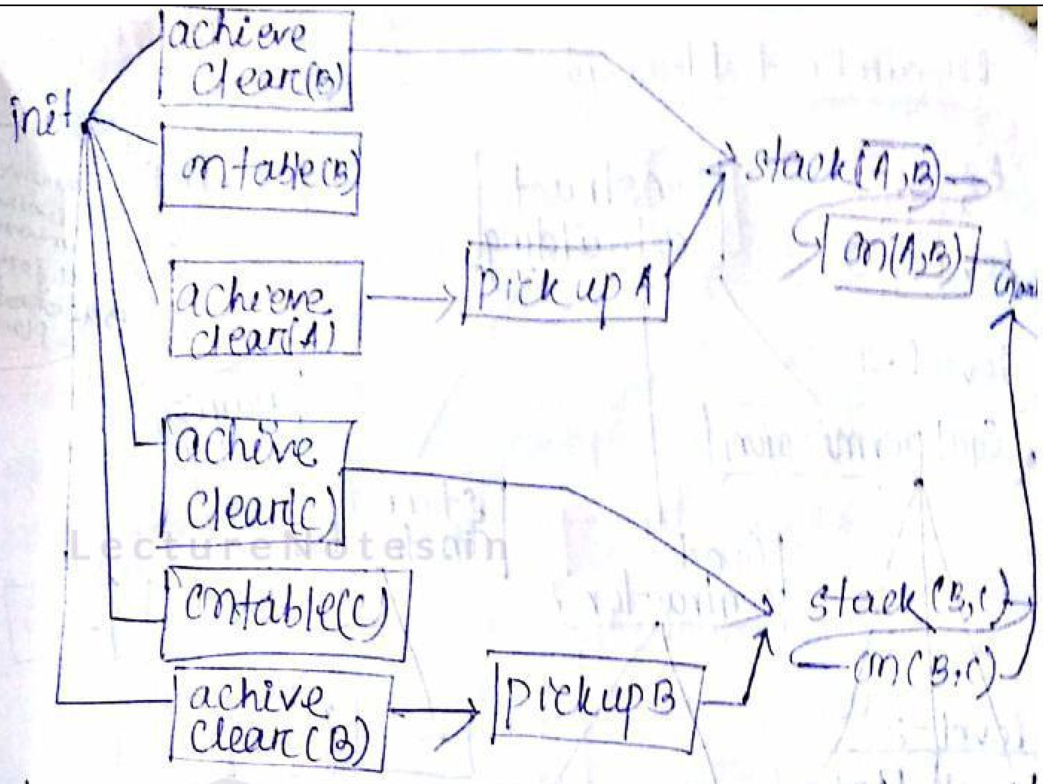
→ Eg:-











Advantages

→ It is sound
→ It is complete
→ It may be optimal wrt plan length

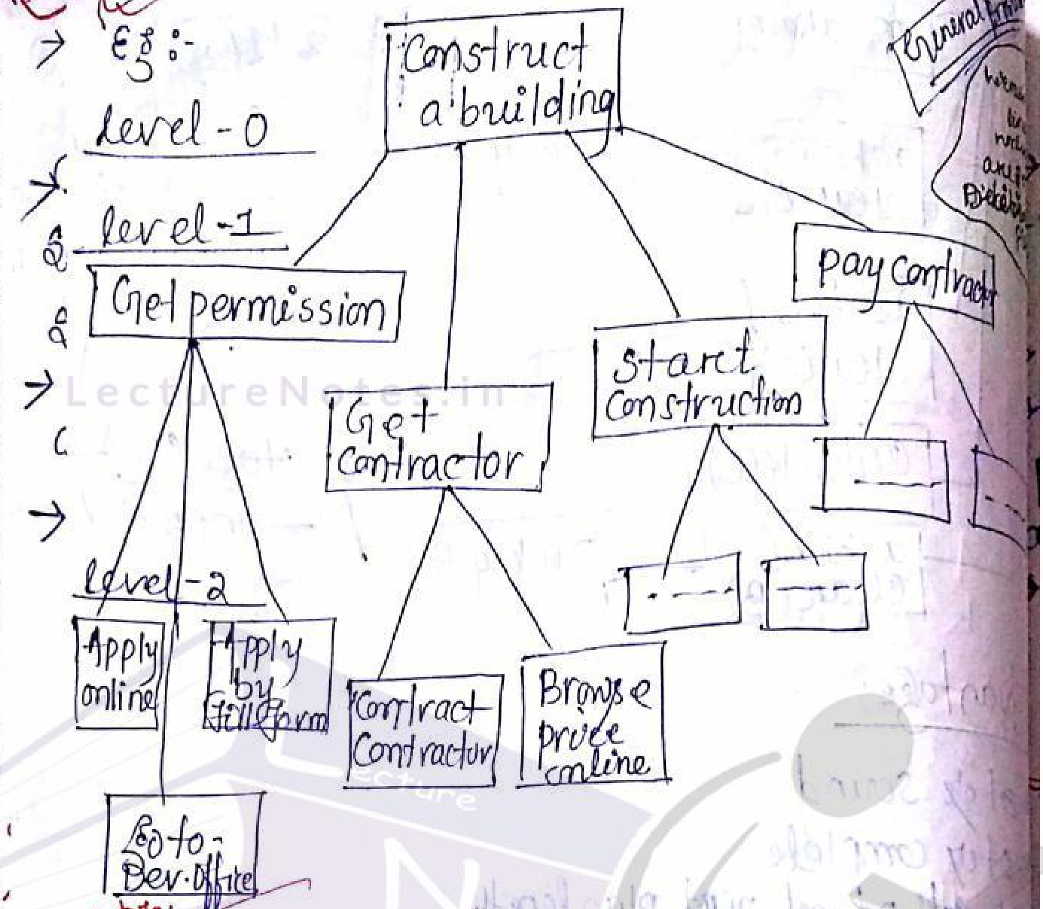
Dis advantage

→ It requires Large search space
→ It more complex than linear planning.

Hierarchical Planning

8/11/2016

→ Eg:-



→It is a planning technique to express the dependen among actions using a hierarchical structure calli hierarchical tasknetwork (HTN)

→HTN planning uses a refinement of action thr decomposition.

→It is similar to and-or graph

→It is used by ABSTRIPS problem solver. (Abstraction base stanfore Research Institute Problem solver)

Advantage:-

→It more efficient than goal stack planning.

→It sound & complete.

→HTN planning gives more expressivity.

Disadvantage:-

→Time Consuming →It is not optimal (best)

→requre memory

Deliberative Planning

→ If a plan for completing an entire task is constructed prior to action then it is called as Deliberated planning

→ Linear, nonlinear, & hierarchical are deliberative planning

→ Reactive planning is used in Reactive system which avoid planning all together and use simple pair of situation action rule.

Other Planning Technique

•) Metaplanning :-

It is a technique for reasoning not just about the Problem being solved but also about the planning process itself.

•) Macro-Operator:-

It allow a planner to build new operator (actions) that represents commonly used sequences of operators.

•) CaseBased planning:

It reuses code plan to make new one.

~ End ~

Artificial Intelligence

Topic:

MEANS END ANALYSIS

Contributed By:

Sankarsan Sahoo

chapt ③
wvamp

10/11/16

and also see constraint satisfication problem

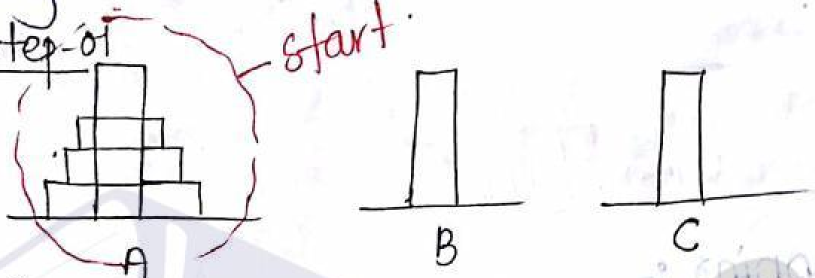
- After heuristic search technique
Refer → { Plz see the (text book) ② & ③ because it's vvamp. for semester exam

Means-Ends Analysis mechanique

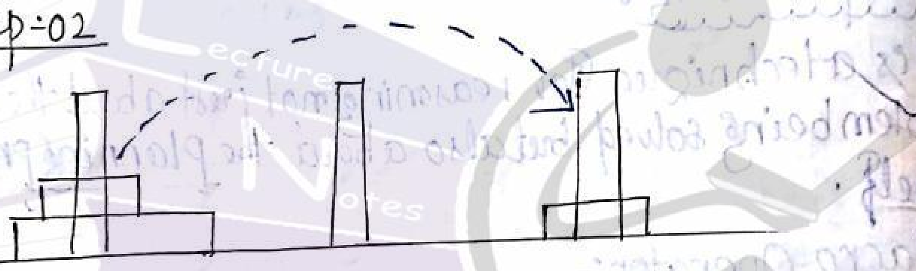
→ It's a problem solving technique used to limit search in problem space.

→ It combines features of both forward & Backward reasoning.

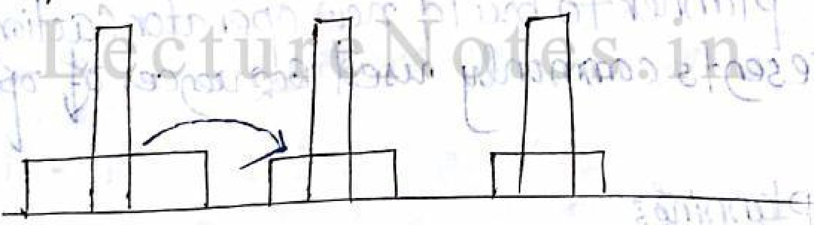
step-01



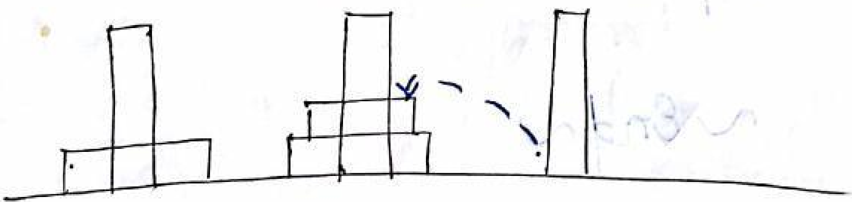
step-02



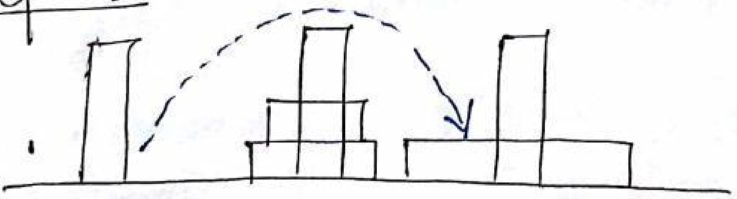
step-03



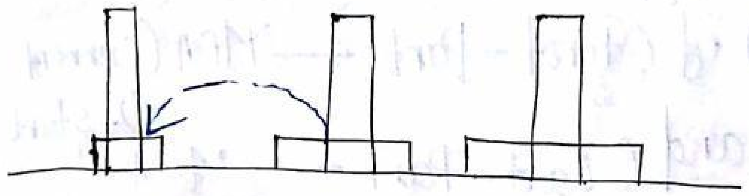
step-04



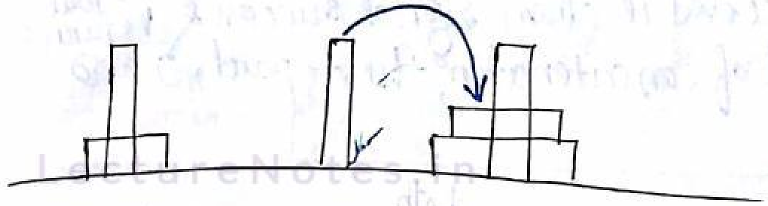
step-05



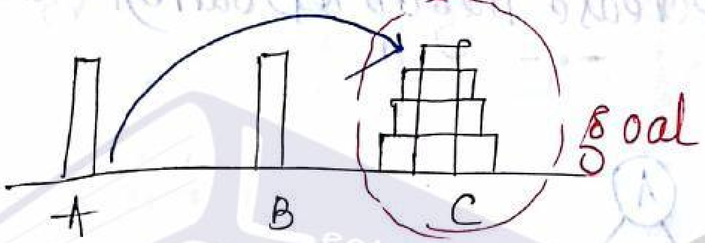
Step-06



Step-07



Step-08



Towers of Honor problem (Q3)

Algorithm:

1) → compair current to goal; if there are no differences bet^n them then return.

2) → other wise select the most imp difference and reduce it by doing the following until success or failure is found.

a) select an untried operator O, i.e applicable to the current difference; if there are no such operator (action) then return failure.

b) attempt to apply 'O' to current. generate description of 2 states as

(i) O start: A state in which O's precond^n are satisfied

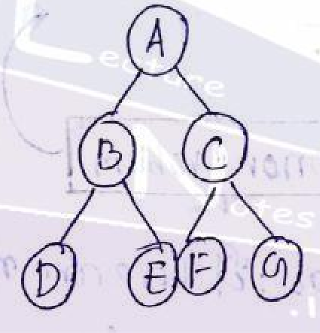
(ii) O result: The state that would resu if 'O' were applied in O start

c) if (First-Part ← MEA (Current, O-Start)

and (Last-Part ← MEA (O-Result, goal)

are successful then signal success & return the result of concatenating First-part, O and last-part.

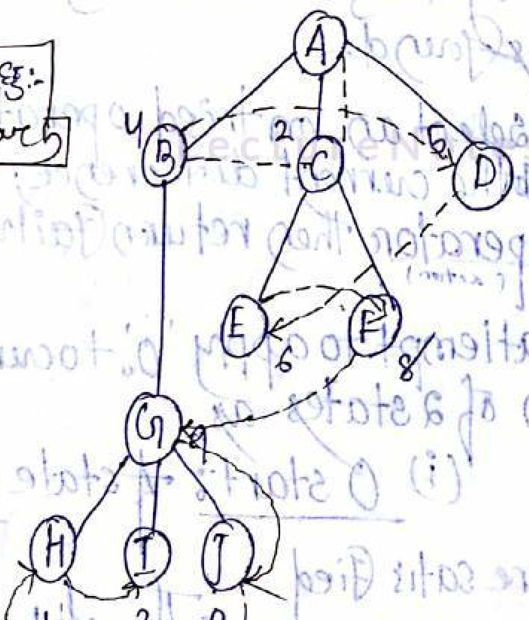
The 1st & last part are both recursively exicu in order to decrease the gap bet^n current & goal.



BFS = A, B, C, D, E, F, G

DFS = A B D E C F G

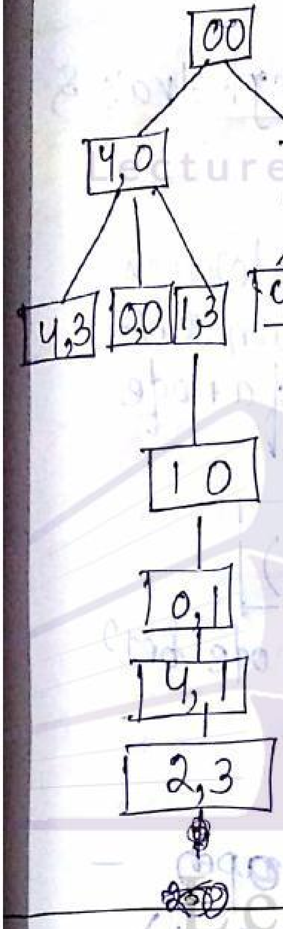
BFS + DFS :- eg:-
Best 1st search



Best 1st Search = BFS + DFS

so, (A - C - B - D - E - F - G - J - H - I)

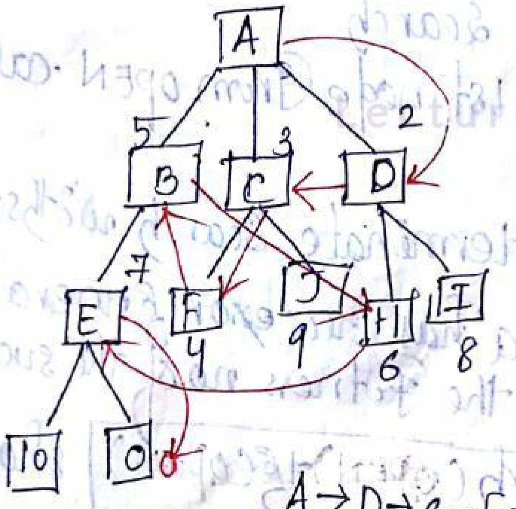
* waterjug problem



4 lit , 3 lit

2, n goal state

n = 0,1,2...



A→D→C–F→B→H→E–N.

V.V.V imp

A* Search Algorithm

→ It is an informed search algorithm use

→ For path find & graph traversal.

→ It combines features of both dijkstra's → Best-First-search algorithm.

→ For A* algorithm we need to calculate fitness number which is the sum of the evalution fun value (h)' and the path cost value (g) of a node from the start node.

So, $f(n) = h(n) + g(n)$

where, n is any node betn start and goal'

Algorithm

1 → Put the inital noad on a list open

2 → if (OPEN is empty or OPEN is goal) the terminate search

3 → Remove the 1st node from OPEN. call this node as a

4 → If a = goal (terminate search with succes)

5 → else if noad 'a' has successor generate them all estimate the fitness no of the successors

i.e, $f(n) = h(n) + g(n)$ short the list by the fitness no.

6)-> name the newlist as closed. ①

7)-> Replace OPEN with CLOSED

8)-> GO TO STEP-②

Advantages of A *

-> It provides optimal path to goal.

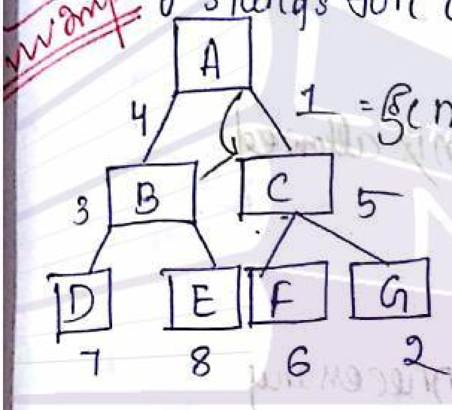
Dis advantages

It requires high memory

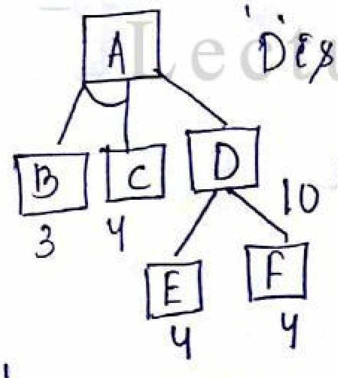
5/11/16

AO* Search Algorithm :- ①

imp O stands for ordered



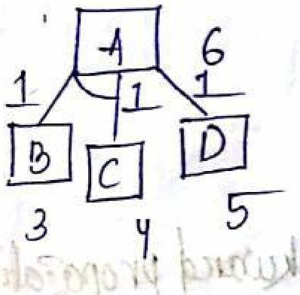
1 = g(n) (path cost)



D is expanded

10 (4+4+1+1)

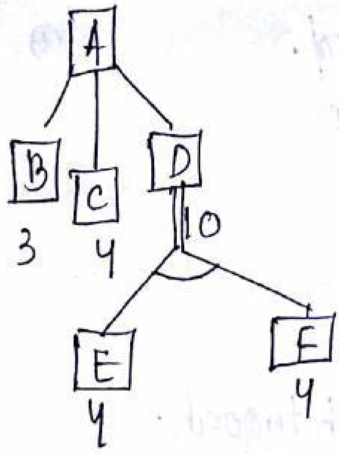
1st



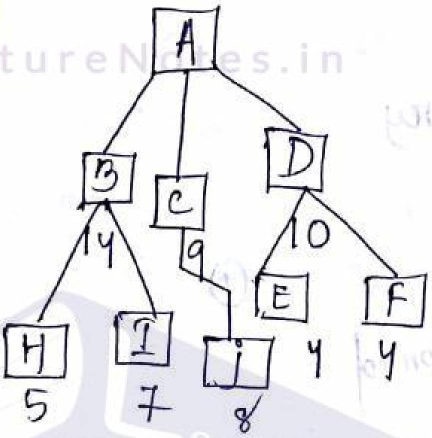
(and-OR graph)

2nd

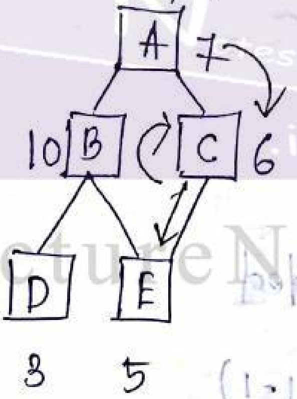
2nd



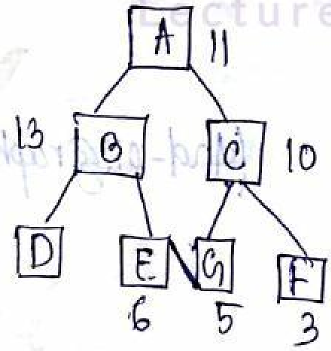
→ 3rd



⇒ If Backward propagation is allowed



← unnecessary



no backward propagation

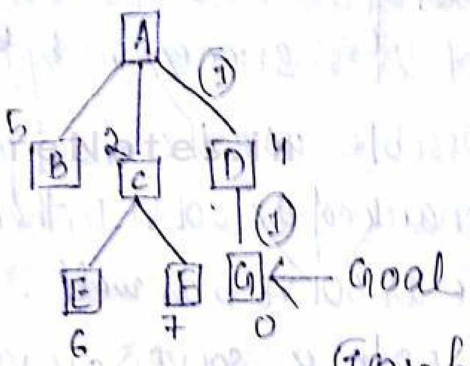
18/11/16

$f(n) = h(n) + g(n)$

$g(n)$ = path cost (actual)

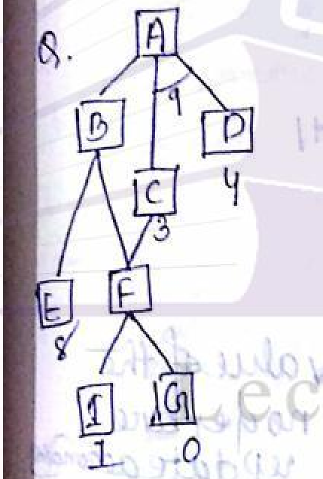
$h(n)$ = heuristic/evaluation (fun) value (best)

$f(n)$ = fitness no.

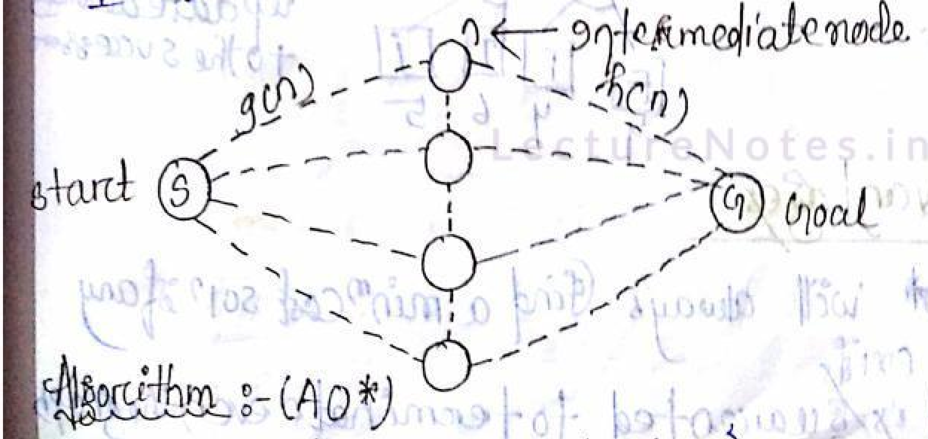


$f(n) = h(n) + g(n)$
$= 0 + 2\{1 + (1)\}$
$= 2$

Q.



$1+1+3+4 = 9$



Algorithm :- (AO*)

1) Initialize the graph to start node.

2) Traverse the graph following the current path accumulating nodes that have not yet been expanded or solved

3) pick any of these nodes and expand it & if it has

no successor then mark it as FUTILITY

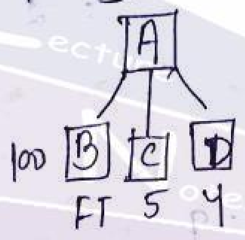
Otherwise calculate only f(n) for each of the successor.

4) → if f(n) = 0, then mark the node as SOLVED

5) → Change the values of f(n) for the newly created node to reflect it's successor by back propaga

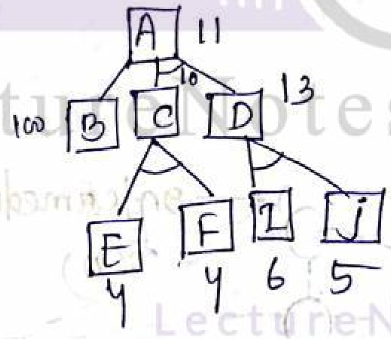
6) → where ever possible use the most promising ro if a node is marked as SOLVED, then mark the parent node as SOLVED as well.

7) → if the starting node is SOLVED or value greater than the futility then stop else repeat from step



5+4+1+1

FUTILITY = 100



value of the nodes are update according to the successor

Advantages

→ AO* will always find a minm cost soln if any exists

→ It is guarenteed to terminate even the graph contains any cycle.

Disadvantage :-

→ Not optimal

→ It is not complete.

Q. Difference bet^n A* and AO*

A* Search	AO* Search
→ Applicable for OR graph	→ Applicable for AND-OR graph
→ No back-propagation	→ Back propagation possible
→ optimal	→ Not optimal
→ Fitness no is not updated	→ Fitness no. can be updated

Q. What do u mean by (isa) & (instance) rel^nship in knowledge representation.

isa

It is used to show class inclusion

Eg: is a (mega-star, Rich)

instance

It is used to show class membership

Eg: instance (Amitabh, megastar)

End

(BEST UP LUCK)

COMET	61078
SATURN	298354
URANUS	359432

COMET	80692
URANUS	431745
SATURN	512437

EARTH	94761
PHOBOS.	81252
EUROPA	