Sub.: _____ SAT SUN MON TUE WED THU

DATE : / /

Reference Books

1. Artificial Intelligence

by Stuart Russell

Peter Norvig

Sub.: Lecture 1 (MAM)

SAT SUN MON TUE WED THU FRI

DATE: 03 / 08 / 2022

Computer Science (Domain)

→ Artificial Intelligence (sub-domain)

→ Machine Learning (sub-domain)

→ Data Mining

→ Big Data

.....

Artificial Intelligence

↙ man made

(not natural/regular)

↘ thinking power

- Man made thinking like as human;
- can be device or model

~~All~~ except AI, all → pre-programmed based

■ Turing test: is a method of inquiry in AI for determining whether or not a computer is capable of thinking like a human being.

(Alan Turing)

United

Sub.: SAT SUN MON TUE WED THU FRI

DATE: / /

* AI → It is a branch of computer science by which we can create intelligent machine which can behave like a human, think like human and able to make decision.

- thinking humanly
- acting humanly
- thinking rationally
- acting rationally

☐ How do human and animals think and act?

☐ How does language relate to thought?

* In AI, you do not need to pre-program machine to do some work.

* In AI, you have to create a machine with programmed algorithm which can work on own intelligent.

☐ Why AI?

→ With the help of AI, we can create such something/ solution/ software or device which can solve real-world-problems very easily and accurately such as — health issues, marketing, traffic issues.

Sub.: _____ SAT SUN MON TUE WED THU FRI

DATE : / /

* With the help of AI, you can create your own personal assistant such as - google assistant, siri, etc.

* You can build such robot that can work in a environment where survival of human can be at risk.

* AI opens a new path to new technologies, new devices, and new opportunities.

Sub: Lecture 2 (MAM)

SAT SUN MON TUE WED THU FRI

DATE: 04 / 08 / 2022

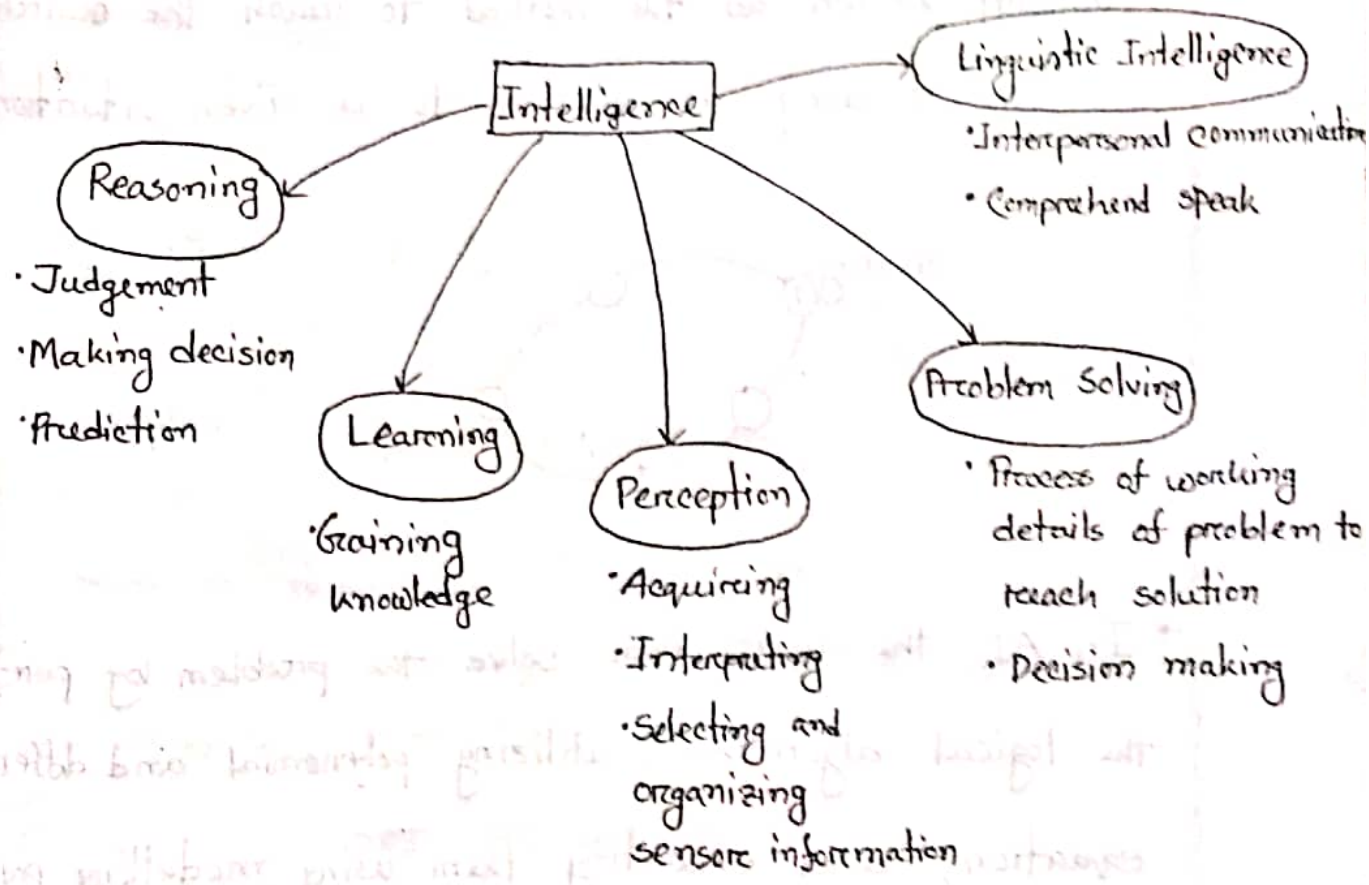
Goal of AI

1. Replicate human intelligence
2. Solve knowledge intensive task
3. An intelligent connection of perception and action
4. Build a machine which can perform tasks that require human intelligence, such as-
   - providing a theorem
   - playing games (chess)
   - performing surgical operation
   - driving a car in traffic
5. Creating some system, which can exhibit intelligent behaviors, learn new things by itself, demonstrate, explain and advise to its users.

Sub.: _____ SAT SUN MON TUE WED THU FRI

DATE : / /

Intelligence is composed of:



Basic →

I) Definition

II) Why AI

III) Goal of AI

IV) Intelligence is composed of

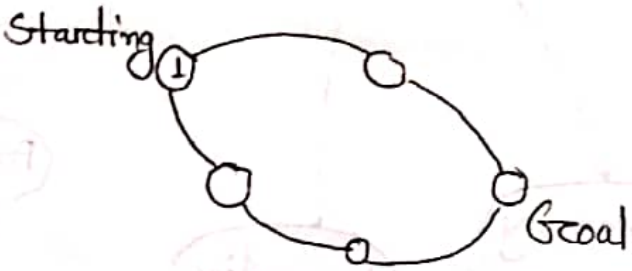
Sub.: _____ SAT SUN MON TUE WED THU FRI

DATE: / /

Part II

Problem Solving:

Commonly known as the method to reach the desired goal or finding a solution to a given situation.



* In AI, the useres can solve the problem by performing the logical algorithms, utilizing polynomial and differential equation, and executing them using modelling paradigms.

* To do this, one need to define problem statement and generate the solution.

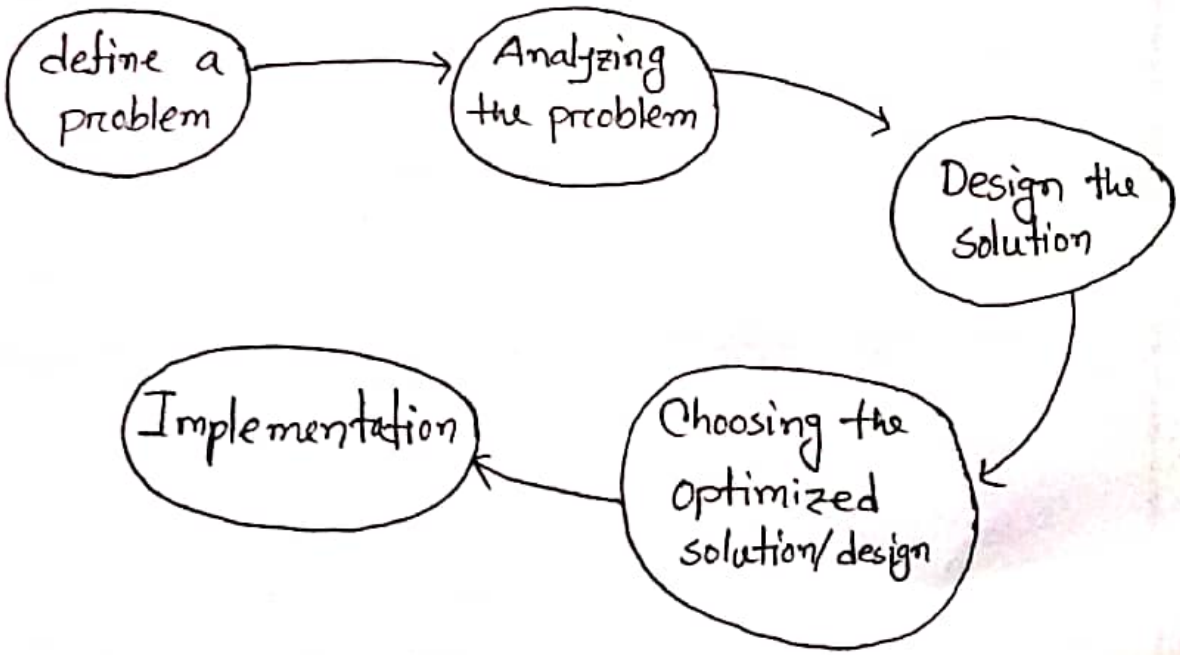
Sub.: _____ SAT SUN MON TUE WED THU FRI

DATE: / /

* Some of the most popularly used problems that are resolved by AI:
  - some kinds of games including chess
  - N-Queen problems (puzzle)
  - Travelling-Salesman problem
  - Towers of Hanoi problem

* Problem solution by searching
  - Finding information one need
  - Most commonly used technique of problem solving in AI

* The process of solving problem



United

Sub.: _____ DATE: / /

SAT SUN MON TUE WED THU FRI

Properties of Searching Algorithm:

1. Completeness: It is said to be complete if it is guaranteed to find the solution if at least any solution for any random input.

2. Optimality: If a solution is found to be best solution (lowest-path-cost) among all.

3. Time complexity: It is a measure of time for an algorithm to complete its task.

4. Space complexity: How much memory is needed to perform the task.

Sub.: Lecture 03 (MAM)

SAT SUN MON TUE WED THU FRI

DATE: 07 / 08 / 2022

Chapter 3

Types of Searching Algorithm:

- Uninformed Search (Blind Search)
- Informed Search (Heuristic Search)

Brute-Force search – traditional

Blind Search algorithm → Brute force Algorithm

Uninformed search: Some important information are not given (no additional info: no domain knowledge)

- BFS (Breadth-First-Search)
- DFS (Depth-First-Search)
- Uniform Cost Search
- Depth Limited search (updated)
- Iterative Deepening Search (updated)

Informed Search: Directed search; heuristic → extra knowledge

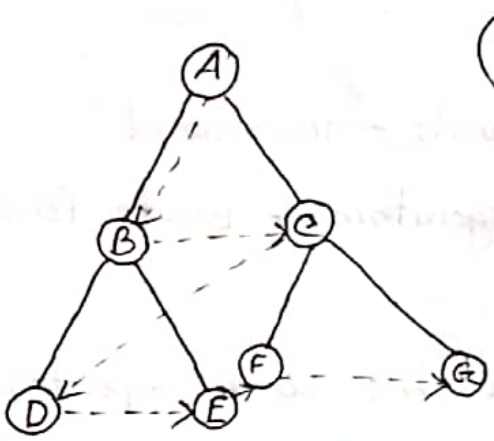
- Greedy Breadth-First Search

United

Sub.: _____ SAT SUN MON TUE WED THU FRI

DATE: / /

BFS: root node expanded first then successors respectively; Graraph based; FIFO queue, sequence - from left.



(Level wise)

Traverse : A → B → C → D → E → F → G

- Advantages of BFS:

- One of the simplest search strategies
- Complete. If there is a solution, BFS is guaranteed to find it
- If there are multiple solutions, then a minimal solution will be found
- The algorithm is optimal if all operators have the same cost. Otherwise, BFS finds a solution with the shortest path.
- Finds the path of minimal length to the goal

Sub: SAT SUN MON TUE WED THU FRI

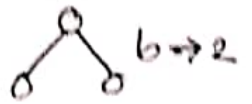
DATE: / /

* Disadvantages of BFS:

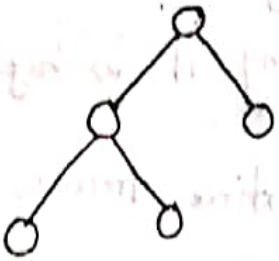
- Consumes large memory space. Its time complexity is more
- It has long pathways, when all paths to a destination are on approximately the same depth.

* Properties:

d - depth
b - branching factor



- Complete → Yes (If b is finite)
- Optimal → BFS, all action have the same cost.
  (cost = non-decreasing function)
- Time Complexity →



$b + b^2 + b^3 + \cdots$

Big O → $O(b^d)$

- Space Complexity →

  Same → $O(b^d)$

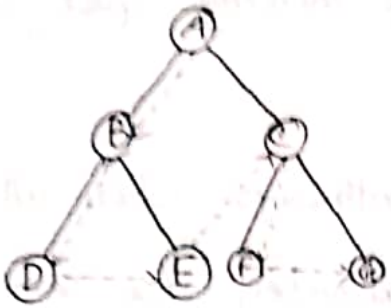
BFS / DFS

* Searching strategy
* Property
* Comparison

Sub: SAT SUN MON TUE WED THU FRI

DATE: / /

DFS: Traverses a graph in depthward motion; LIFO



Traverse: A→B→D→E→C→F→G

* Properties

- Completeness → DFS is complete if the search tree is finite

- Optimality → DFS is not optimal, meaning the no. of steps in reaching the solution, or the cost spent in reaching it, is high.

- Time complexity → If the entire tree is traversed, $O(V)$
  In case of a graph, $O(V+E)$
  (V = no. of vertex; E = no. of edges)

- Space complexity → $O(h)$; h = maximum height of the tree.

sub: Lecture - 4 (MAM)

Date: 10/09/2022

Problem Definition and Formulation

Agent - Perceiving information through sensors from environment.

Goal-based agent - A goal-based agent takes it a step further by using a goal in the future to help make decisions about how best to reach that outcome.

* A problem can be defined formally by 5 components:

1. Initial State: This stage state requires an initial state for the problem which starts the AI agent towards a specified goal. In this state new methods also initiate problem domain solving by a specific class.

2. Action: This stage of problem formulation works with function with a specific class taken from the initial state and all possible actions done in this stage.

Sub.: SAT SUN MON TUE WED THU FRI

DATE : / /

3. Transition : This stage of problem formulation integrates the actual action done by the previous action stage and collects the final stage to forward it to their next stage.

4. Goal Test: This stage determines that the specified goal achieved by the integrated transition model or not, whenever the goal achieves the stop the action and forward into the next stage to determine the cost to achieve the goal.

5. Path Costing: This component of problem solving numerical assigned what will be the cost to achieve the goal. It requires all hardware, software and human working cost.

Optimal solution: is one whiose measure of quality is close to the best that could theoretically be obtained.

Frontier → a set of leaf nodes.

Loopy path → infinite, unsolvable.

Sub: _____ SAT SUN MON TUE WED THU FRI

DATE / /

Explored set → keeps track of the nodes that have already been expanded.

Uniform Cost Search

UCS is an uninformed search algorithm that uses the lowest cumulative cost to find a path from the source to destination.

Nodes are expanded, starting from the root, according to the minimum cumulative cost. Then it is implemented using a Priority Queue.

✓ how many step - doesn't care

✓ cost should be less

* Properties

1. Completeness: completeness is guaranteed if only if the cost of every step is some positive number.

2. Optimal: UCS is optimal. This is because, at every step the path with the least cost is chosen, and paths never gets shorter as nodes are added, ensuring that

Sub:

Date: / /

the search expands nodes in the order of their optimal path cost.

3. Time complexity:

Total solution cost = C^*

Each step cost at least = ϵ

Steps = $1 + C^*/\epsilon$

Time complexity = $O(b^{[1 + C^*/\epsilon]})$

4. Space Complexity:

$O(b^{[1 + C^*/\epsilon]})$

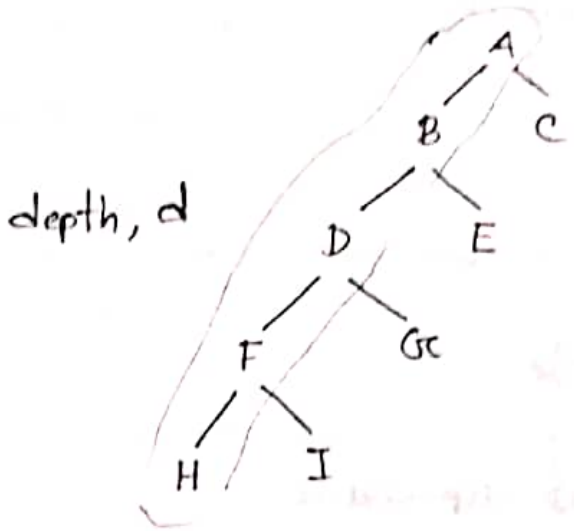
** Limitations of DFS and how can you address them?

Sub: Lecture - 05 (MAM)

SAT SUN MON TUE WED THU FRI

DATE 16 / 09 / 2022

* DFS Limitations



The failure of DFS in finite state spaces can be alleviated by supplying DFS with a predefined depth-limit l, i.e nodes at depth l have no successores.

Depth Limited Search

Unbounded length
Shallowest depth

- $l < d \longrightarrow$ incomplete
- $l > d \longrightarrow$ non optimal

- Time complexity: $O(b^l)$
- Space complexity: $O(bl)$

United

Sub: _____ SAT SUN MON TUE WED THU FRI

DATE: / /

Iterative Deepening Search

- repeated iteration ; towards depth;
- gradually increases the limit

* Properties :

1. Completeness : It is complete
2. Optimality : It is optimal if step cost = 1
3. Time complexity : $O(b^d)$
4. Space complexity : $O(bd)$

Bidirectional Search

Two simultaneous searches -

1) forward from the initial state
2) backward from the goal

Time complexity → $O(b^{d/2})$

Space complexity → $O(b^{d/2})$

Sub: _____ SAT SUN MON TUE WED THU FRI

DATE / /

* Comparing uninformed search strategies

Criterion	BFS	UCS	DFS	DLS	IDS	BS
Completeness	Yes[a]	Yes[a,b]	No	No	Yes[a]	Yes[a,d]
Time Comp.	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$	$O(b^l)$	$O(b^d)$	$O(b^{d/2})$
Space Comp.	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$	$O(bl)$	$O(bd)$	$O(b^{d/2})$
Optimality	Yes[c]	Yes	No	No	Yes[c]	Yes[c,d]

Informed (Heuristic) Search Strategies

- uses problem specific knowledge
- general approach best-first search

Searching strategy: defined by picking the order of node expansion.

Heuristic function $h(n)$: estimated cost of the cheapest path from node n to a goal node.

$g(n)$

Greedy Best first Search → incomplete, not optimal.

A* Search.

heuristic function: is a shortcut to solve a problem when there is no exact solution for it and or the time to find the solution is too long.

United

Sub: Lecture 6 (MAM)

SAT SUN MON TUE WED THU FRI

DATE: 23/08/2022

A* Search

minimizing the total estimated cost solution

The Cost:

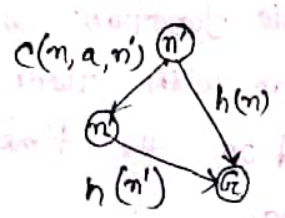
$f(n)$ = $g(n) + h(n)$ → n node to goal node

- $g(n)$ → path cost from start node
- $h(n)$ → estimated cost of the cheapest solution through n.

- form of BFS
- It evaluates nodes by combining $g(n)$, the cost to reach the node and $h(n)$, the cost to get from the node to the goal.
- complete and optimal

Admissible heuristic → never overestimates the cost to reach the goal

Consistent heuristic → $h(n) \leq c(n, a, n') + h(n')$



Sub:

DATE

Properties:

1. Completeness : Yes, it is complete
2. Optimality: It is optimal
3. Time complexity: $O(b^{Cd})$
4. Space complexity:

Memory Bounded Heuristic Search

Iterative Deepening A* algorithm

Recursive Best First Search

Sub.: Lecture-7 (MAM) [Online]

SAT SUN MON TUE WED THU

DATE: 25 / 08 / 2022

A* Implementation

* Heuristic function h(n) and how to make it

* Heuristic may be:

1. h(n) > actual cost (optimal solution is not possible)
2. h(n) = actual cost (best case scenario)
3. h(n) < actual cost (admissible, consistent heuristic)

Optimality of A* Search Algorithm

** Admissible হলে consistent হবে, কিন্তু consistent হলে admissible নাও হতে পারে।

Sub: Lecture - 8 (MAM)

SAT SUN MON TUE WED THU FRI

DATE: 30 / 08 / 2022

Local Search Algorithms

- Operate using a single current node
- generally moves only to neighbors of that node

A local search algorithm starts from a candidate solution and then iteratively moves to a neighbor solution.

* Advantages
  - use very little memory
  - can find reasonable solutions in large or finite state spaces

neighborhood → set of all potential solutions that differ from the current solution by the minimal possible extent

* Local Search for Optimization

$$\arg\min_{x} f_0(x)$$

$$s.t. \; f_i(x) \leq 0 \;, \quad i = \{1, \ldots, k\}$$

$h_j(x)$

local search is a heuristic method for solving computationally hard optimization problems.

** Beyond Classical Search

systematic solution/method → classical search

Sub: SAT SUN MON TUE WED THU FRI

DATE : / /

* Hill-Climbing Search (8-queens)

- local domain का knowledge चाहिए
- It is a greedy approach (best move का लेके step होते जाते)
- no back-tracking
- also called greedy local search
- if a state is better than current state then it is moved current s

Global maxima → target

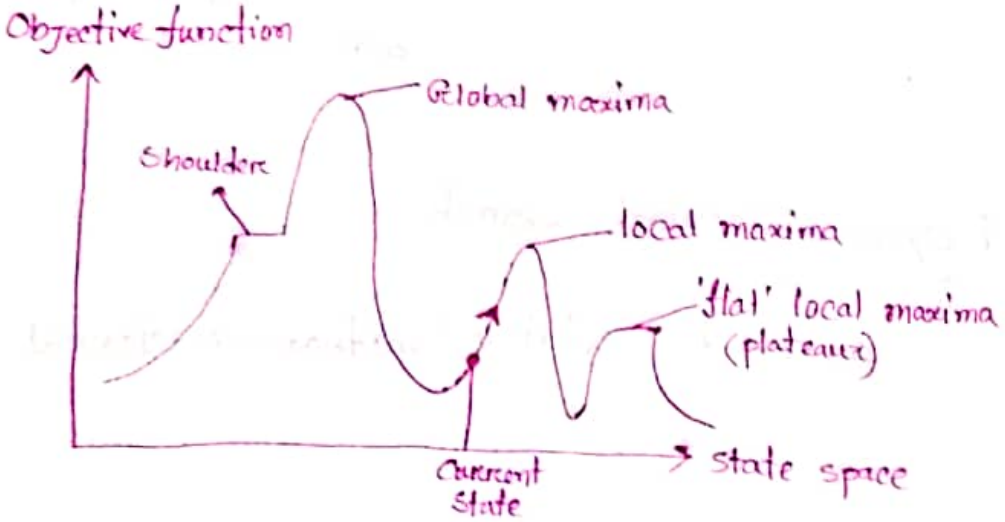
Local maxima → higher peak among the neighboring states

Shoulder - Plateaux

* Problems in hill climbing -

i) Local maxima
ii) Plateau/flat maxima } best move केके आगे जाते का
iii) Ridge - direction change करते का

Hill climbing is a heuristic search used for mathematical optimization problems in the field of AI. Given a large set of inputs and a good heuristic function, it tries to find a sufficiently good solution to the problem.



Sub.: Lecture-9 (MAM)

SAT SUN MON TUE WED THU FRI

DATE: 05 / 09 / 2022

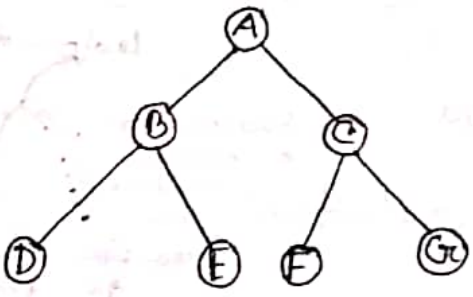
Simulated Annealing Search

* Challenges of hill-climbing

- never makes "downhill" moves toward states with lower value is guaranteed to be incomplete, because it can get stuck on a local maximum.
  - plateau
  - ridge

** Random walk - moving to a successor chosen uniformly at random from the set of successors.

- A pure random walk is complete but extremely inefficient.



A B D B E B A C F C G

It is complete but inefficient as there are Repeating nodes

Sub.: SAT SUN MON TUE WED THU FRI

DATE: / /

** Combining hill-climbing and random walk yeilds both efficiency and completeness. Simulated annealing is such an algorithm. (allows downward steps)

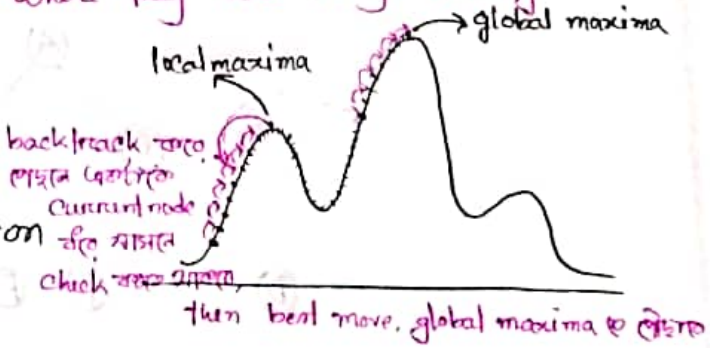
Annealing - high temperature to gradually low

- instead of picking the best move, it picks random move
- if the move improves the situation, it is accepted.
- otherwise accepts move with probability < 1

improve move

badness move

Annealing is a process where in metallurgy where metals are slowly cooled to make them reach a state of low energy where they are very strong.



T → temperature

ΔE → worsen evaluation

gradient ascent

gradient descent → minimizing cost

** How simulated annealing works?

** Properties.

Advantages
- checks all neighbours
- easy to code
- gives good solution
- statistically guarantees optimal solution

Disadvantages
- slow process
- can't tell whether an optimal solution is found.

United

Sub: _____ SAT SUN MON TUE WED THU FRI

DATE: / /

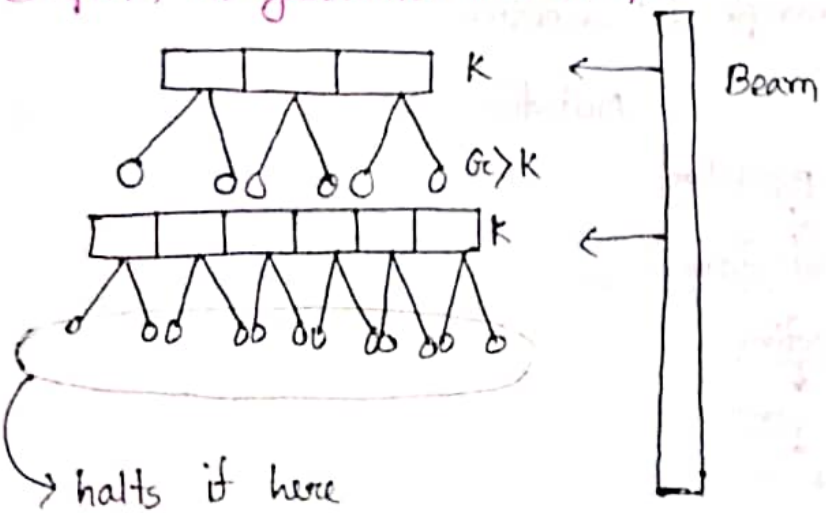
Annealing – Lowering temperature gradually

Quenching – Lowering temperature rapidly

* Local Search with Multiple Present States

Local Beam Search (space complexity related)

- keeps track of k states rather than just one.
- beam width (β)/n is given. (β/n মানে best value memory তে store)
- begins with k randomly generated states
- space complexity constant
- At each step, all the successors of k are generated
- β=1 হলে hill climbing search
- if anyone is a goal, the algorithm halts.
- not complete; no guarantee to find optimal solution



if β = 2

store করতে;
space কম লাগবে.
no. of nodes কম
হলে এই method কে
sort করা হোক না হোক
space/time কম লাগবে
So

Sub.: Lecture - 10 (MAM)

SAT SUN MON TUE WED THU FRI

DATE: 07 / 09 / 2022

Chapter - 6

Constraints Satisfaction Problem

Constraints → restriction, condition

- have alternate ways, but bound to do the task in a certain way.

CSP – to solve a wide variety of problems efficiently.

* Problems have these things-

set of ~~values~~, variables, values, constraints

** A ~~variable~~ problem in solved when each variable has a value that satisfies all the constraints on that variable is called CSP.

* CSP deals with general problems instead of specific problems; can solve complex problems.

* The main idea in to eliminate the large position of the search space all at once by identifying the variable combination that violate the constraints.

Sub: _____ SAT SUN MON TUE WED THU FRI

DATE : / /

* Defining CSP

A CSP consists of three components, X, D and C:

X is set of variables $\{X_1, \ldots, X_n\}$

D is set of domains $\{D_1, \ldots, D_n\}$, one for each variable

C is set of constraints that specify allowable combinations of values.

$D_i \rightarrow \{v_1, \ldots, v_k\}$

$C_i \rightarrow \{scope, rel\}$

scope : a tuple of variables that participate in the constraints

rel : a relation that defines the values that those variables can take on.

Assignment →

I) Complete assignment : every variable is assigned

II) Consistent assignment : doesn't violate any constraints

III) Partial assignment : assign values to some of the variables.

Sub.: _____ SAT SUN MON TUE WED THU FRI

DATE : / /

* CSP Example : Map coloring

- region গুলোকে color করা হবে; adjacent regions or neighbouring regions same color করা যাবে না।

** Why formulate a problem as a CSP?

Answer : I) yeilds natural representation of problems

II) faster than state-space searchers

* Job-shop scheduling

Mid Syllabus → Informed search & Uninformed search.

// Characteristics

// Complexity comparison

United

Sub.: Lecture - 11 (MAM)

SAT SUN MON TUE WED THU FRI

DATE: 14 / 09 / 2022

* Variations on the CSP formalism

* Discrete domain
* Finite domain
* Infinite domain
* Continuous domain
* Linear constraints – each variable appears only in linear form
* Non-linear constraints – nonlinear variable
  * Unary constraints – restricts value of a single variable
  * Binary constraints – relates two variables

Map coloring problem এ পাশাপাশি দুইটা region same color হতে পারবে না
SA ≠ NSW

Binary CSP → with only binary constraints

* Global constraints – A constraint involving an arbitrary no. of variables.

Alldiff → all the variables must have different values

Example : Sudoku (All row-columns are different)

Cryptarithmatic problems

United

DATE : / /

Chapter 7 (Book 2)

Uncertainty is defined as the lack of exact information or knowledge that hel us to find correct conclusion.

Probability → uncertainty ; came from random variable.

Random variable → value came from random experiment

* Why Pro Uncertainty ?
  - → Uncertain input (missing, noisy data)
  - → Uncertain knowledge (multiple causes leads to multiple effects; incomplete knowledge of causality in domain)
  - → Uncertain output (abduction-induction are uncertain)

Priore probability → যার কোনো হাত থাকে না
P(A), P(B), P(H), P(T) ...

Random experiment → Coin toss

Ω = Certain event
∅ = impossible event

(Omega) Ω = Random sample space → {T, H}

** Exact measurement এর জন্য probability দরকার

IId → Identically Independent

- যে সকল probability আগের পরের experiment এর উপ dependent না (priore probability)
- সকল random variable IId (নির্দিষ্ট distribution আছে)

Joint Probability → একাধিক ঘিলে

Example : 2 টি die roll করলে অনেক গুলো pair হতে পারে

$P(5, 6) = \frac{2}{6} = \frac{1}{3}$

Sub.: _____ SAT SUN MON TUE WED THU

DATE : / /

* Set Notations → $A \cap B = A \wedge B$, $A \cup B = A \vee B$, $\bar{A} = \neg A$, $\Omega = t$, $\phi = f$

* Definition 7.2

$$P(A) = \frac{|A|}{|\Omega|}$$

Summation of probability = 1

* Theorem 7.1

* Conditional Probability

Example 7.4

* Definition 7.3

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

'|' → Given sign; যে sign এর পরের টা given থাকে.

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

[Probability of B when A is true.]

* Independent Event

P(A) & P(B) যদি Independent হয় তাহলে

$$P(A \wedge B) = P(A) \cdot P(B)$$

Example 7.5

United

Sub: _____ SAT SUN MON TUE WED THU FRI

DATE : / /

** Chain Rule

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

$$\Rightarrow P(A \wedge B) = P(A|B) \cdot P(B) \quad \text{——— (1)}$$

for n variables

$$P(X_1, X_2, --, X_n) = P(X_n | X_1, X_2, --, X_{n-1}) \cdot P(X_1, X_2, --, X_{n-1})$$

$$= P(X_n \wedge (X_1, X_2, --, X_{n-1})) \cdot P(X_{n-1} | X_1, X_2, --, X_n$$

$\prod$ → multiplication কে denote করে

$$= \prod_{i=1}^{n} P(X_n | X_1, X_2, --, X_{i-1})$$

* Marginalization

* Baye's Theorem

Conditional probability থেকে Baye's Rule আসছে।

Baye's Rule থেকে ~~Baye's~~ Bayesian Classifier আসছে।

↳ classification Algorithm

$$P(A \wedge B) = P(A|B) \cdot P(B) \quad \text{——— (1)}$$

$$P(B \wedge A) = P(B|A) \cdot P(A) \quad \text{——— (11)}$$

Sub:

SAT SUN MON TUE WED THU

DATE / /

$P(A \wedge B) = P(B \wedge A)$

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

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

↳ Baye's Rule.

Posterior ← (Prob. of B when A is true)

$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$ → marginal probability (Prob. of evidence)

→ Prior Prob (Prob. of hypothesis)

Likelihood

(Prob. of evidence)

SUB: Lecture-12 (MAM)

SAT SUN MON TUE WED THU

DATE: 19 / 09 / 2022

Chapter 13 (Russel)

Quantifying Uncertainty

$0 \leq P(\omega) \leq 1$ for ω $\quad \sum_{\omega \in \Omega} P(\omega) = 1$

* Probability with a Proposition

ϕ: $\quad P(\phi) = \sum_{\omega \in \phi} P(\omega)$

$$P(\text{total} = 11) = P((5,6)) + P((6,5))$$

$$= \frac{1}{36} + \frac{1}{36}$$

$$= \frac{1}{18}$$

* Conditional Probability

$P(\text{Cavity} \mid \text{toothache}) = 0.6$

↳ given that

$P(\text{cavity}) = ?$

$P(\text{covid-19} \mid \text{fever}) = 0.7$

↳ given

$P(\text{covid-19}) = ?$

Sub: SAT SUN MON TUE WED THU FRI

DATE: / /

$$P(a|b) = \frac{P(a \wedge b)}{P(b)}$$

$$\Rightarrow P(a \wedge b) = P(a|b) \cdot P(b) \quad \text{[Product Rule]}$$

Example

$T = \{\text{cavity}, \text{toothache}, \text{catch}\}$

$\text{Die}_1 = \{1, 2, 3, 4, 5, 6\}$

$\text{Age} = \{\text{Juvenile}, \text{teen}, \text{adult}\}$

$\text{Weather} = \{\text{sunny}, \text{rainy}, \text{snow}, \text{cloudy}\}$

$P(\text{cavity} \mid \neg\text{toothache} \wedge \text{teen}) = 0.1$

$P(w = \text{sunny}) = 0.6$

* Joint Probability (calculates the probability of two events occurring together and at the same point in time)

Marginalization or Marginal Probability

Summing out

$$P(Y) = \sum_{z \in Z} P(Y, z)$$

$Z \rightarrow$ another sample space

$$P(Y) = \sum_{z} P(Y|z) \cdot P(z) \quad \text{[conditioning rule]}$$

Sub: Lecture - 13 (MAM)

SAT SUN MON TUE WED THU FRI

DATE: 17 / 10 / 2022

Marginalization (summing out) - task of marginal probability

↳ tells us just to add up some probabilities to get to the desired probabilistic quantity.

$$P(a|b) = \frac{P(b|a) \cdot P(a) \longrightarrow}{P(b) \longrightarrow} \Big\} \text{marginalization}$$

Observable value

Unobservable value

$$P(Y) = \sum_{z \in Z} P(Y, z)$$

$$= \sum_{z \in Z} P(Y|z) \cdot P(z) \quad \text{(rules of conditioning)}$$

$X, Y \rightarrow$ joint probability

$X \wedge Y \rightarrow$ joint probability

(',' , '$\wedge$' → similar sign)

Normalization → (The goal of normalization is to transform features to be on a similar scale. This improves the performance & training stability of model)

$$P(X|e) = \alpha P(X, e) = \alpha \sum_{y} (X, e, y)$$

$$\alpha = \frac{1}{P(e)}$$

SAT SUN MON TUE WED THU

Sub:

DATE : / /

Independence Probability

$$P(a|b) = P(a)$$

$$P(b|a) = P(b)$$

$$P(a \wedge b) = P(a) \cdot P(b)$$

$$P(a \wedge b) = P(a|b)\,P(b)$$
$$P(b \wedge a) = P(b|a)\,P(a)$$
} Product rule

from product rules,

$$P(b|a) = \frac{P(a|b) \cdot P(b)}{P(a)} \rightarrow \text{Bayes rule}$$

Unconditional facts →

Example

Bayesian classifier

13.16

13.17

$$\prod_{i=1}^{n} P(\text{effect}|\text{cause})\,P(\text{cause}) \rightarrow \text{Naive Bayesian Classifier}$$

$$V_{NB} = P(\text{cause}) \prod_{i=1}^{n} P(\text{effect}|\text{cause})$$

United

Sub: Lecture - 14 (MAM)

SAT SUN MON TUE WED THU FRI

DATE: 19 / 10 / 2022

Example of Naive Bayesian Classifier

$$\prod_{i=1}^{n} \underbrace{P(\text{effect}_i|\text{cause})}_{\text{Conditional Probability}} \cdot \underbrace{P(\text{cause})}_{\text{Prior Probability}}$$

	X		Y
Name	H	W	
	~~V~~	~~W~~	M
	—	—	F
			M
			F
			F

Prior probability of M, $P(M) = \frac{2}{5}$

" " " F, $P(F) = \frac{3}{5}$

$$P(M) + P(F) = \frac{2}{5} + \frac{3}{5}$$

$$= \frac{5}{5} = 1$$

Learning (Machine Learning) - 3 types

1. Supervised learning - output, level info, supervisor info
2. Unsupervised learning

Sub: _____ SAT SUN MON TUE WED THU

DATE : / /

* Example 1

* Example 2

$$V_{NB} = \underset{v_j \in (yes, no)}{\arg\max} \; P(v_j) \prod_i P(a_i | v_j)$$

(argument)

a_i = attribute

v_j = Y/N

** Normalization করে value যোগ করলে 1 হবে

*** অন্য example দিয়ে implement করতে হবে for 3 instances

Lab mid → 23/10/2022

Sub.: _____ SAT SUN MON TUE WED THU FRI

DATE : / /

* Example 1

Days	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Instance :

Outlook = Sunny

Temperature = Cool

Humidity = High

Wind = Strong

Sub.: SAT SUN MON TUE WED THU FRI

DATE: / /

$P(\text{Play Tennis} = yes) = 9/14 = 0.64$

$P(\text{Play Tennis} = no) = 5/14 = 0.36$

Outlook	Y	N
Sunny	2/9	3/5
Overcast	4/9	0
Rain	3/9	2/5

Humidity	Y	N
High	3/9	4/5
Normal	6/9	1/5

Temperature	Y	N
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Windy	Y	N
Strong	3/9	3/5
Weak	6/9	2/5

$$V_{NB} = \underset{v_j \in \{yes, no\}}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$$

$$= \underset{v_j \in \{yes, no\}}{\operatorname{arg\,max}} P(v_j)$$

$P(\text{Outlook} = \text{Sunny} | v_j) P(\text{Temperature} = \text{Cool} | v_j)$

$P(\text{Humidity} = \text{High} | v_j) P(\text{Wind} = \text{Strong} | v_j)$

$V_{NB} = P(yes) P(\text{sunny}|yes) P(\text{cool}|yes) P(\text{high}|yes) P(\text{strong}|yes)$

$= 0.0053$

$V_{NB} = P(no) P(\text{sunny}|no) P(\text{cool}|no) P(\text{high}|no) P(\text{strong}|no)$

$= 0.0206$

$$V_{NB}(yes) = \frac{V_{NB}(yes)}{V_{NB}(yes) + V_{NB}(no)}$$

$$= 0.205$$

$$V_{NB}(no) = \frac{V_{NB}(no)}{V_{NB}(yes) + V_{NB}(no)}$$

$$= 0.795$$

Example-2

No.	Colour	Legs	Height	Smelly	Species
1	White	3	Short	Yes	M
2	Green	2	Tall	No	M
3	Green	3	Short	Yes	M
4	White	3	Short	Yes	M
5	Green	2	Short	No	H
6	White	2	Tall	No	H
7	White	2	Tall	No	H
8	White	2	Short	Yes	H

Unknown : Colour = Green, Legs = 2, Height = Tall, Smelly = No

$$P(M) = \frac{4}{8} = 0.5$$

$$P(H) = \frac{4}{8} = 0.5$$

Sub.: _____ SAT SUN MON TUE WED THU FRI

DATE: / /

Color	M	H
White	2/4	3/4
Green	2/4	1/4

Legs	M	H
2	1/4	4/4
3	3/4	0/4

Height	M	H
Tall	3/4	2/4
Short	1/4	2/4

Smelly	M	H
Yes	3/4	1/4
No	1/4	3/4

$P(M|Instance1) = P(M)\,P(Color=Green|M)\,P(Legs=2|M)\,P(Height=Tall|M)\,P(Smelly=No|M)$

$= 0.5 * \frac{2}{4} * \frac{1}{4} * \frac{3}{4} * \frac{1}{4}$

$= 0.0117$

$P(H|Instance1) = P(H)\cdot P(Color=Green|H)\,P(Legs=2|H)\,P(Height=Tall|H)\,P(Smelly=No|H)$

$P(H|Instance1) > P(M|Instance1)$

∴ The instance belongs to species H.

Sub: Lecture-1 (MMR)

SAT SUN MON TUE WED THU FRI

DATE: 26 / 09 / 2022

Chapter 7

Logical Agents

Logical agents - in which we design agents that can form representations of a complex world, use a process of inference to derive new representations about the world, and use these new representations to deduce what to do.

Reasoning - The logical process of drawing conclusions, making predictions or constructing approaches towards a particular thought with the help of existing knowledge.

Logic - a general class as a general of representations to support knowledge-based agent.

- Propositional Logic
- First-order Logic

* Knowledge-based agent - are those agents who have the capability of maintaining an internal state of knowledge, reason over that knowledge, update their knowledge after observations and take action.

Sub:

DATE: / /

Knowledge-based agents are composed of -

I) Knowledge base

II) Inference system

Knowledge-base - a set of sentences (technical terms) expressed in a language called knowledge representation language.

Axiom - when the sentence is taken as given without being derived from other sentences.

Inference - deriving new sentences from old.

** Algorithm of generic knowledge based agent

function KB-AGENT (percept) returns an action

persistent: KB, a knowledge-base

t a counter, initially 0, indicating time

TELL (KB MAKE-PERCEPT-SENTENCE)(percept, t))

action ← ASK (KB MAKE-ACTION-QUERY (t))

TELL (KB MAKE-ACTION-SENTENCE, (action, t))

t ← t+1

return action

Sub: _____ SAT SUN MON TUE WED THU FRI

DATE: / /

Knowledge-based agent - 3 levels:

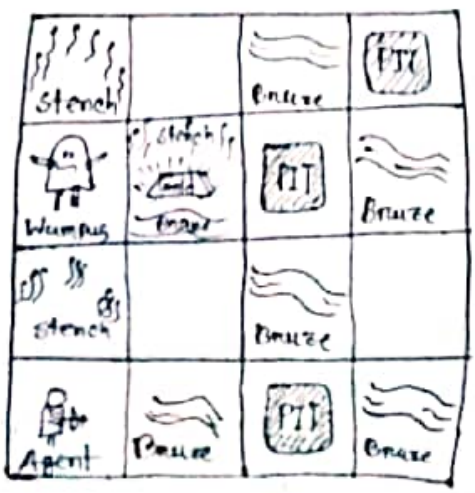
1) Knowledge level
2) Logical level
3) Implementation level

Declarative approach - TELL sentences one by one until the agent knows how to operate in its environment.

Procedural approach - encodes desired behaviors directly as a program code.

* The Wumpus World

The Wumpus world is a simple world example to illustrate the worth of a knowledge-based agent and to represent knowledge representation.



United

Sub:

SAT SUN MON TUE WED THU FRI

DATE: / /

Components of Wumpus:

1. Rooms adjacent to wumpus are smelly, it have stench
2. The rooms adjacent to PITs have breeze
3. There will be glitter in the room iff it has gold.
4. Wumpus can be killed by the agent and it would emit a horrible scream which can be heard anywhere in the cave

* PEAS Description of the Wumpus world:

Performance measure -

- +1000 points if agent comes out with gold
- -1000 points for being eaten by the wumpus or get killed by falling into the pit.
- -1 for each action, -10 for using arrow
- The game ends if the agents dies or came out of the cave.

Environment -

- A 4*4 grid of rooms
- The agent initially in room square [1, 1], facing right
- Location of gold and wumpus are random except [1,1]
- Each square except [1,1] can be pit with probability 0.2.

United

Sub.: _____ SAT SUN MON TUE WED THU FRI

DATE : / /

Actuators -

- Left turn
- Right turn
- Move forward
- Grab
- Release
- Shoot

Sensors -

- Agent will receive stench in room adjacent to wumpus.
- Agent will perceive breeze in room adjacent to pit.
- Agent will perceive glitter in room where the gold is present.
- Agent will perceive bump if he walks into the wall
- When the wumpus is shot, it emits a horrible scream which can be heard anywhere in the cave.

* The Wumpus world Properties

- Partially observable
- Deterministic
- Sequential
- Static
- Discrete
- One agent

* Exploring the Wumpus World.

Sub: Lecture-2 (MMP) Online

SAT SUN MON TUE WED THU FRI

DATE: 27 / 09 / 2022

* Some Propositional Rules for Wumpus world:

- for a 4*4 square board, there will be 7*4*4 = 122 propositional variables.

R1) $\neg S_{11} \rightarrow \neg W_{11} \wedge \neg W_{12} \wedge \neg W_{13}$

R2) $\neg S_{121} \rightarrow \neg W_{11} \wedge \neg W_{21} \wedge \neg W_{22} \wedge \neg W_{31}$

R3) $\neg S_{12} \rightarrow \neg W_{11} \wedge \neg W_{12} \wedge \neg W_{21} \wedge \neg W_{13}$

R4) $S_{12} \rightarrow W_{13} \vee W_{12} \vee W_{22} \vee W_{11}$

* Representation of Knowledge base for Wumpus World

* Prove that Wumpus is in room [1,3]

** Derive a propositional logic for wumpus world

* Propositional Logic - is a declarative statement which is either true or false but can't be both.

Conjunction	$\wedge$
Disjunction	$\vee$
implication	$\rightarrow$
biconditional	$\leftrightarrow$
negation	$\neg$

United

Sub: _____ SAT SUN MON TUE WED THU FRI

DATE: / /

* A BNF grammar of sentences in propositional logic

Sentence -

i) Atomic → True / false

ii) Compound → sentence "connective" sentence

Connective → AND, OR, NOT, implies, equivalent

Tautology → universal truth / valid sentence

Contradiction → inconsistent sentence

* Truth Table : five logical connectives

P	Q	¬P	P ∧ Q	P ∨ Q	P → Q	P ↔ Q
F	F	T	F	F	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
T	T	F	T	T	T	T

* Proving things

Proof - sequences of sentences, where each sentence is either a premise or a sentence derived from earlier sentences in the proof by one of the rules of inference.

* Horn Sentences

$(P \rightarrow Q) = (\neg P \vee Q)$ [Truth table for Proof]

Sub:

SAT SUN MON TUE WED THU FRI

DATE: / /

- Propositional Logic is a weak language
- Rules Propositional rules for wumpus world

7 marks; 2 questions

Sub: Lecture - 3 (MMR) Online

SAT SUN MON TUE WED THU FRI

DATE 01 / 11 / 2022

* Propositional logic

- quantification express করতে পারে না

* First-Order logic / Predicate logic

- Covers quantification & predicates

- extended part of propositional logic

Chapter - 8

First-Order Logic

* Representation Revisited

** Programming Language lackings

* The Language of Thought

Communication এর জন্য context থাকতে হবে

Ambiguity - problem for a representation language

Whorf hypothesis =

United

Sub.: _____ SAT SUN MON TUE WED THU FRI

DATE : / /

* Combining the best of formal and natural languages

- Objects - people, houses, something
- Relations - unary relations are properties
- Functions - father of.

यहे 3 for निरूप natural language

Quantification → some, all

Figure 8.1

* Syntax & semantics of FOL

Domain

Domain elements

Tuples

First Order Logic

Objects - things with individual identities

Properties - that distinguishes objects from other objects

Relations - that holds among sets of objects

Functions - subjects of relations

User Provides

- Constant symbols
- Functional symbols
- Predicate symbols

FOL Provides

- variable symbols
- Connectives
- Quantifiers

**sentences are built from terms and atoms

* Quantifiers
  - Universal Quantification
  - Existential Quantification

Quantifier Scope

Sub:

SAT SUN MON TUE WED THU FRI

DATE / /

- Connections between All and Exists
- Quantified Inference Rules

United

Sub: Lecture - 9 (MMR) Online

SAT SUN MON TUE WED THU FRI

DATE 04 / 11 / 2022

Final Examination Lab

Chatbot তৈরি করতে হবে Using Python

* Models for FOL

Domain — set of objects

non-empty set হতে পারে না, তবে না থাকলে object থাকতে হবে

Figure 8.2

* Symbols and Interpretations

Figure 8.3

¬ , = , ∧ , ∨ , ⇒ , ⇔ } → Precedence

* Terms
* Atomic Sentences
* Complex Sentences
* Quantifiers
  - Universal (∀)
  - Existensial (∃)
  - Nested
* Equality

Sub: Lecture-5 (MMR) Online

SAT SUN MON TUE WED THU FRI

DATE: 06 / 11 / 2022

* Using FOL

- System কে TELL/ASK করতে হবে

Assertions and Queries in FOL

- Sentence কে TELL এর সাহায্যে Knowledge base এ add করা হয়
  এবং assertion বলে
- ASK এর সাহায্যে → Query; always positive answer পাওয়া যায়

The Kinship Domain

- parenthood, brotherhood, marriage - etc

* The Wumpus World

* Knowledge Engineering in FOL

Knowledge engineer - someone who investigates a popular domain, learns what concepts are important in

United

Sub: _____ SAT SUN MON TUE WED THU FRI

DATE / /

Knowledge Engineering Steps:

1. Identifying the task
2. Assemble the relevant knowledge

   Knowledge aquisition
3. Decide a vocabulary
4. Encode general knowledge of the domain
5. Encode the specific problem
6. Pose queries to the inference procedure
7. Debug the knowledge base

Exercises

Sub:

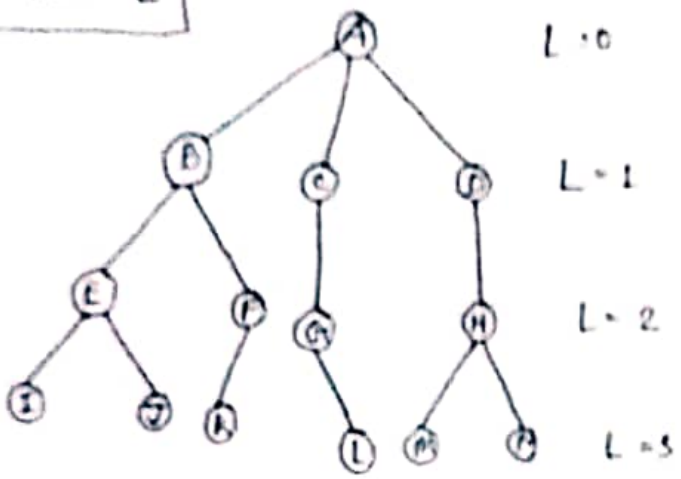
SAT SUN MON TUE WED THU FRI

DATE / /

* Iterative Deepening Search

IDS is a state space search strategy in which a depth-limited version of DFS is run repeatedly with increasing depth limits until the goal is found.

Goal node - L



first iteration: A

2nd iteration : ABCD

3rd iteration : ABEFCGcDH

4th iteration : ABEIJFKCGcLDHMN

Here we found and came goal node L after 4th iteration.

TC → $O(b^d)$

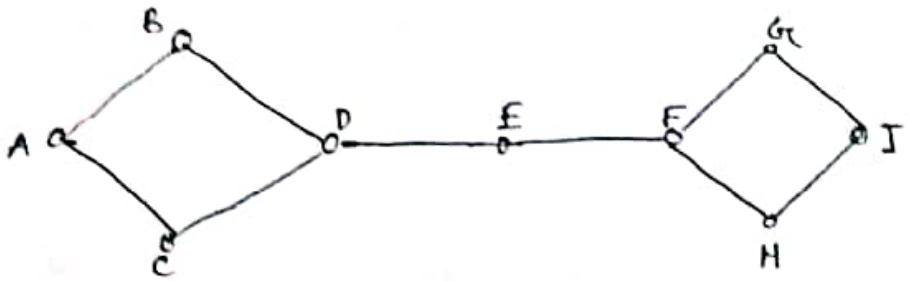
S.C → $O(bd)$

Sub: _____ SAT SUN MON TUE WED THU FRI

DATE : / /

* Bidirectional Search

Bidirectional search is a graph search algorithm that finds a shortest path from an initial vertex to a goal vertex/ node. It runs two simultaneous searches - one forward from the initial state, and one backward from the goal, stopping when the two meets.



We want to find if there is a path from A to I. Here we can execute two searches, one from the starting node A and other from goal node I. When both forward & backward search meet at node 'E', we know that we've found a way from node A to I. Hence search can be terminated now.

* Why ⇒ $O(b^{d/2} + b^{d/2})$ is less than $O(b^d)$

* When ⇒ 1) branching factor same for A and I
2) when BFS is used

United

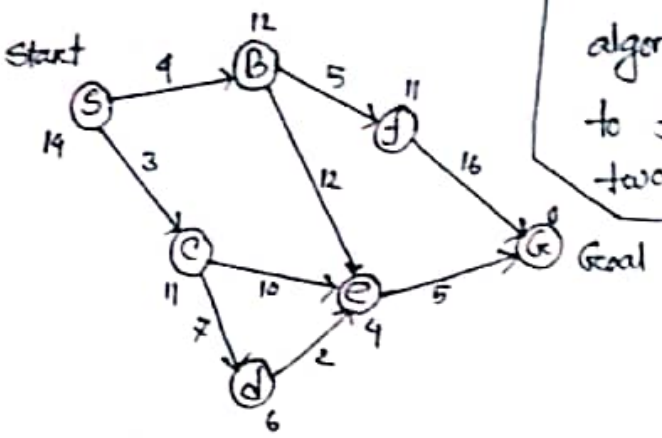
Sub: _____

SAT SUN MON TUE WED THU FRI

DATE : / /

* A* Search

A* search algorithm is a simple and efficient search algorithm that can be used to find the optimal path between two nodes in a graph. It is used for the shortest path finding.



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

- $g(n)$ → actual cost from start node to n
- $h(n)$ → Estimated cost from n to goal node

$f(S) = 0 + 14 = 14$

$S \rightarrow B = 4 + 12 = 16$

$S \rightarrow C = 3 + 11 = 14$

$SC \rightarrow BE = (3+10) + 4 = 17$

$SC \rightarrow B = (3+7+6) + 6 = 16$

$SCD \rightarrow E = (3+7+2) + 4 = 16$

$SCDE \rightarrow G = (3+7+2+5) + 0 = 17$

$T.C = O(b^d)$

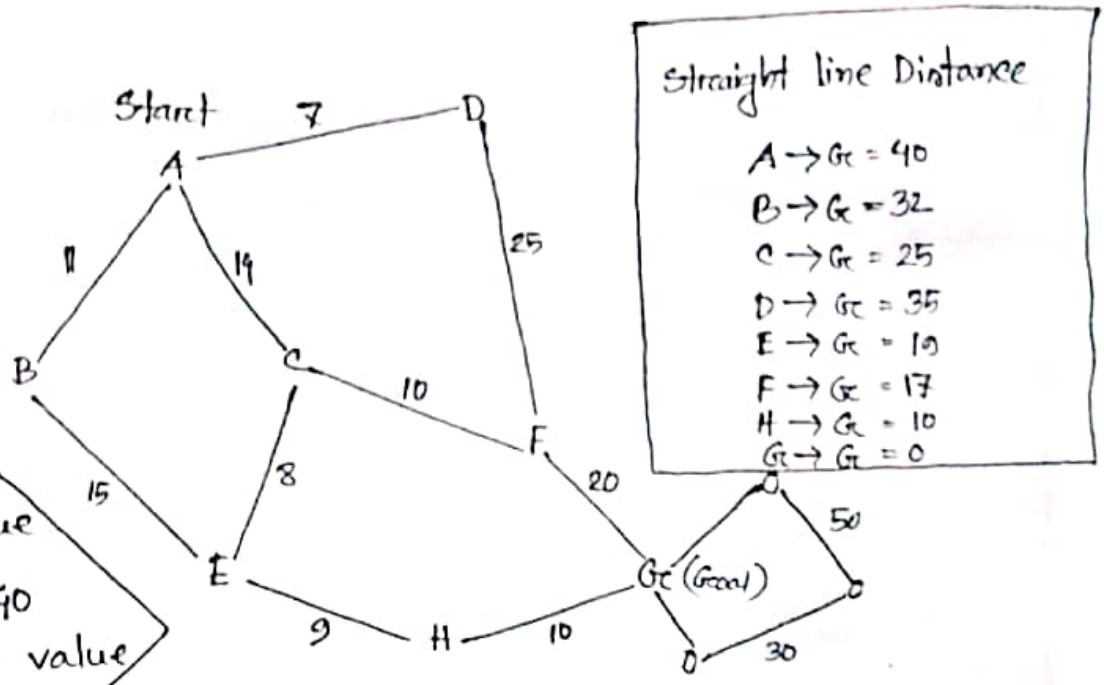
$S.C = O(b^d)$

United

Sub.: SAT SUN MON TUE WED THU FRI

DATE: / /

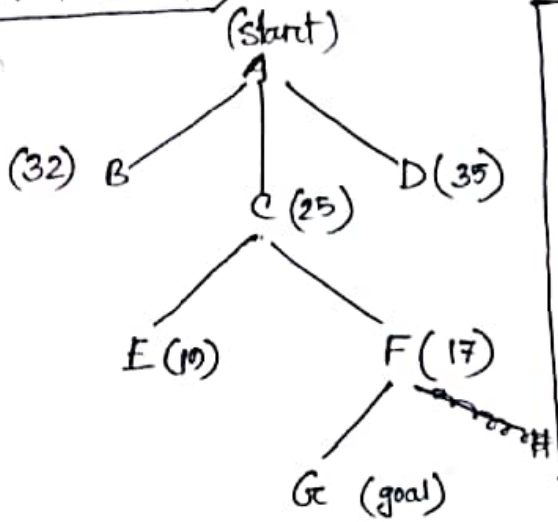
* Best First Search (Informed, Heuristic)



Straight line Distance

A → Gt = 40
B → Gt = 32
C → Gt = 25
D → Gt = 35
E → Gt = 19
F → Gt = 17
H → Gt = 10
Gt → Gt = 0

with heuristic value
A → C → F → Gt = 40
with out heuristic value
A → C → F → Gt = 44



Definition: Best first search algorithm is a traversal technique that decides which node is to be visited next by checking which node is the most promising one and then check it.

Based on heuristic value
A → C → F → Gt

It gives good solution. Optimal solution is not guaranteed.

T.C → $O(b^m)$ / $O(b^d)$

In worst case it performs same but in normal/average case it performs better than uninformed search.

Sub.: _____ SAT SUN MON TUE WED THU FRI

DATE: / /

* How to minimize the total estimated solution cost using the best-first search, A* search algorithm?

→ By making A* admissible, we can minimize the total estimated solution cost.

For A* search algorithm, the equation we use is

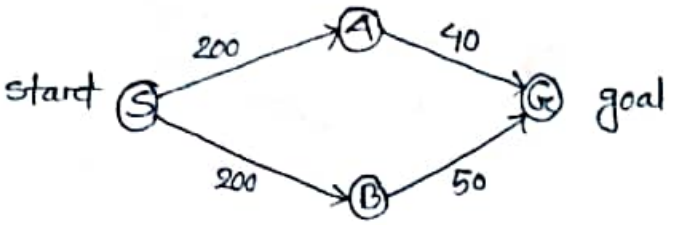
$$f(n) = g(n) + h(n)$$

where $g(n)$ = actual cost from start node to n.

$h(n)$ = estimated cost from n to goal node.

We know, when.

$h(n) \geq$ actual cost, then it is overestimation and when $h(n) \leq$ actual cost, then it is underestimation.



here,

$g(A) = 200$

$g(B) = 200$

actual cost, $A \rightarrow G = 40$

actual cost, $B \rightarrow G = 50$

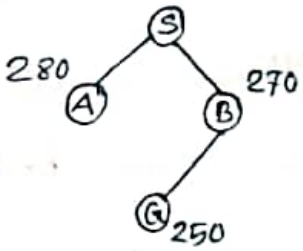
United

Sub.: _____ SAT SUN MON TUE WED THU FRI

DATE : / /

Case I : Overestimation

Let, $h(A) = 80$, $h(B) = 70$ } > actual cost



Now, $f(A) = 200 + 80 = 280$

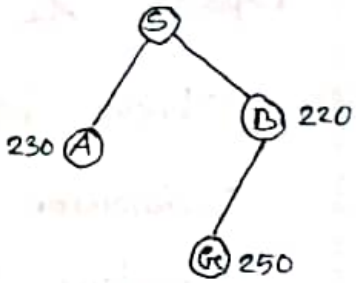
$f(B) = 200 + 70 = 270$

$f(G) = g(G) + h(G) = 250 + 0 = 250$

As A is greater so the search is stopped.

Case II : Underestimation

Let, $h(A) = 30$, $h(B) = 20$ } < actual cost



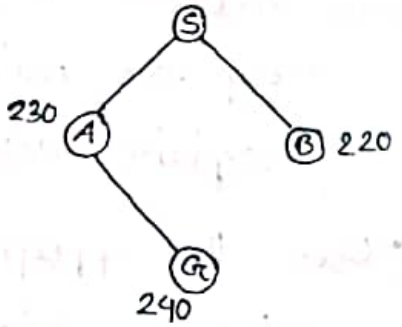
$f(A) = 200 + 30 = 230$

$f(B) = 200 + 20 = 220$

$f(G) = 250$

here A* doesn't stop but as A is less than G. So it explores A then G.

$f(G) = g(G) + h(G) = 240 + 0 = 240$



Underestimation gives optimal solution

here 240 is less than 250, so the estimated cost is minimized.

Sub.: _____ SAT SUN MON TUE WED THU FRI

DATE: / /

* What are the learning outcomes of Artificial Intelligence?

⇒ There are several learning outcomes of AI. Some of them are given below-

- Distinguishes between a conventional system and intelligent system
  - describes data, information, knowledge
  - explains algorithmic versus heuristic methods
  - defines structured and unstructured.
- Explains AI concepts and its applications
  - knows how to represent knowledge
  - describe searching methods
  - explains approximate reasoning
- Represents knowledge using various different techniques.
  - explains first order logic and predicate calculus
  - explains semantic network
  - explains rule-based system
  - explains case-based system
- Uses the appropriate searching techniques to reach desired goals
  - knows state space search
  - distinguishes between data driven and goal driven search
  - uses BFS, DFS; - uses some heuristic search method

United

Sub: SAT SUN MON TUE WED THU FRI

DATE: / /

* What is Uninformed Search?

⇒ Uninformed search is a class of general-purpose search algorithms which operates in brute force-way. Uninformed search algorithms do not have additional information about state or search space other than how to traverse the tree, so it is also called a blind search.

* What is Informed Search?

⇒ Informed search algorithms have information on the goal state which helps in more efficient searching. This information is obtained by a function that estimates how close a state is to the goal state.

* What is the heuristic function of informed search strategy?

⇒ Heuristic is a function which is used in informed search. It is a way to inform the seach about the direction of the goal. It provides an informed way to guess which neighbor of a node will lead to a goal. It finds the most promising path. It takes the current state of the agent as its input and produces the estimation of how close the agent is from the goal.

Sub.: _____ SAT SUN MON TUE WED THU FRI

DATE: / /

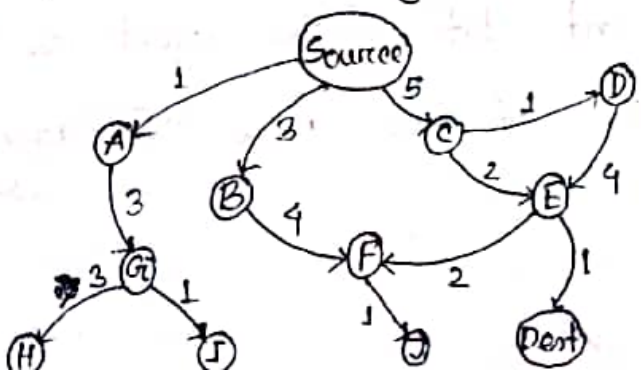
Mid Question (2017-18)

1. What in blind search technique?

⇒ Blind search technique applies a way in which search tree in searched without any information about the search space like initial state operators and test for the goal, so it is also called uninformed search. It examines each node of the tree until it achieves the goal node.

2. Describe Uniform-Cost Search method.

⇒ Uniform-cost search in an uninformed search algorithm that uses the lowest cumulative cost to find a path from the source to the destination. In this method, the nodes are expanded, starting from the root, according to the minimum cumulative cost. The uniform-cost search is then implemented using a priority queue.



Time complexity needed to run ucs in:

$O(b^{[1+c^*/\epsilon]})$ [steps = $1+c^*/\epsilon$]

b = branching factor.

c^* = total solution cost

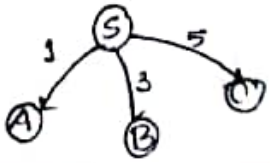
ϵ = each step cost

United

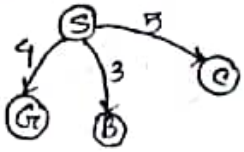
Sub.: _____ SAT SUN MON TUE WED THU FRI

DATE : / /

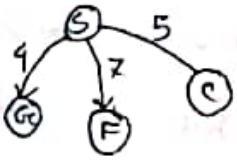
First we have the source node in the priority queue and we add their children with their cumulative distance as priority



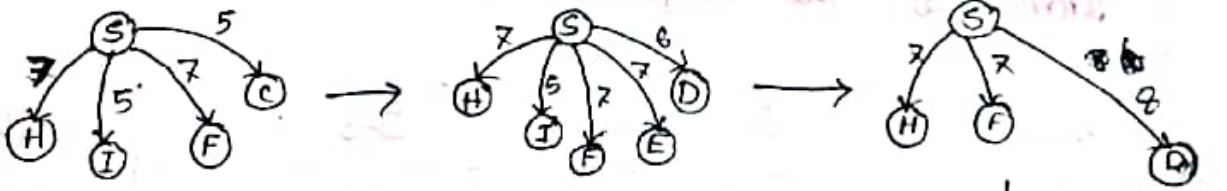
here a 'A' has the minimum distance i.e maximum priority. So it is extracted from the ~~dertis~~ list and its children are added to the PQ.



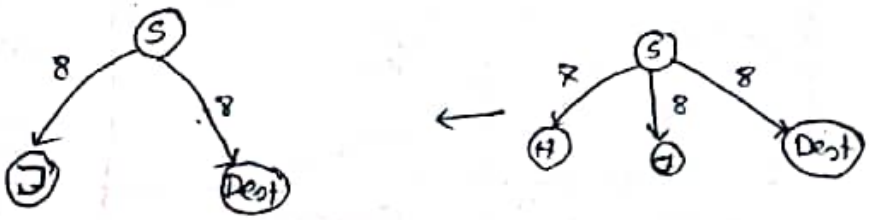
Similarly removing B and adding its children to the PQ



Following this procedure we get,



The minimum distance between the source & destination has been found i.e 8.



Sub: _____ SAT SUN MON TUE WED THU FRI

DATE : / /

3. How to evaluate an algorithm's performance?

⇒ An algorithm's performance can be evaluated based on the four properties -

1. Completeness : An algorithm is said to be complete if it is guaranteed to find the solution if at least for any random solution.
2. Optimality: If a solution is found to be best among all the solutions for an algorithm
3. Time Complexity: It is a measure of time for an algorithm to complete its task.
4. Space Complexity: It defines ~~m~~ how much memory is needed to complete the task for an algorithm.

4. Compare BFS, DFS & Iterative deepening search strategies in terms of the evaluation criteria.

⇒

Criterion	BFS	DFS	IDS
Completeness	Yes	No	Yes
Optimality	Yes	No	Yes
Time Complexity	$O(b^d)$	$O(b^m)$	$O(b^d)$
Space Complexity	$O(b^d)$	$O(bm)$	$O(bd)$

Sub: _____

SAT SUN MON TUE WED THU FRI

DATE / /

Masud Sire

1. Chapter 3
2. Chapter 4
3. Chapter 6
4. Chapter 13

Mahbub Sire

1. Chapter 7
2. Chapter 8

Unit 2