Machine Learning & Artificial Intelligence

Unit I
Introduction to artificial Intelligence

What is Artifical Intelligence?

→ Artifical intelligence is the study of systems that act in a way that to any observer would appear to be intelligent.

→ Artifical intelligence involves using methods based on the intelligent behavior of humans and other animals to solve complex problems.

Strong AI Vs Weak AI

→ Strong AI - The followers of strong AI belive that by giving a computer program sufficient processing power, and by providing it with enough intelligence, one can create a computer that can literally think and is conscious in the same way that a human is conscious.

→ The ~~dir~~ questions to be answered by in case of strong AI are

i) How much processing power is sufficient for creating strong AI

ii) How much intelligence is needed to create human like AI.

Weak AI - This is simply the view that intelligent behavior can be modeled and used by computers to solve complex problems. The point of view argues that just because a computer behaves intelligently does not prove that it is actually intelligent in the way that a human is.

→ Weak AI refers to any AI ~~todl~~ tool that focuses on doing one task really well. That is, it has a narrow scope in terms of what it can do. The idea behind weak AI isn't to mimic or replicate human intelligence, rather, its to simulate human behavior.

Strong Methods and Weak Methods

(1) Strong Methods

Strong methods problem solving depends on a system being given a great deal of knowledge about its world and the problems that it might encounter. ~~Strong~~

-> Strong methods problem solving depends on the weak methods because a system with knowledge is useless without some methodology for handling that knowledge.

Weak Methods

Weak methods in AI use systems such as logic, automated reasoning and other general structures that can be applied to a wide range of problems but that do not necessarily incorporate any real knowledge about the world of the problem that is being solved.

Knowledge Representation

- A computer needs a representation of a problem in order to solve it.
- In AI, both problem representation and knowledge representation are needed.

A representation must be

1) Efficient - not wasteful in time or resources.
2) Useful - allows the computer to solve the problem
3) Meaningful - really relates to the problem.

Knowledge Representation

Need for good representation:

The representation that is used to represent a problem is very important. The way in which the computer represents a problem the variable it uses, and the operators it applies to those variables can make the difference between an efficient algorithm and an algorithm that doesn't work at all.

When applying artificial intelligence to search problems, a useful efficient and meaningful representation is essential. The representation should be such that the computer does not waste too much time on pointless computations, it should be such that the representation really does relate to the problem that is being solved, and it should provide a means by which the computer can actually solve the problem.

* Example :- Search contact lenses on a football feild.

→ Assumption :- Computer has access to an omniscient oracle that will answer questions about the feild and can accurately identify whether the contact lens is in a particular spot.

Rep 1 :- Divide the whole field into 4 equal parts (squares) and ask the oracle to search each square.
- The computer will only need to perform 4 searches and thus the time taken for search will be very small.
- Yet this approach will not be very helpful as despite of yeilding the correct result, it will still be very difficult to locate the lenses, given the large size of the squares.

→ Thus in this approach we have, ~~high~~ low ~~ese~~ search time but also give us highly unaccurate result.

Rep 2: In exactly opposite approach, we can divide the football field into individual pixels / atoms.

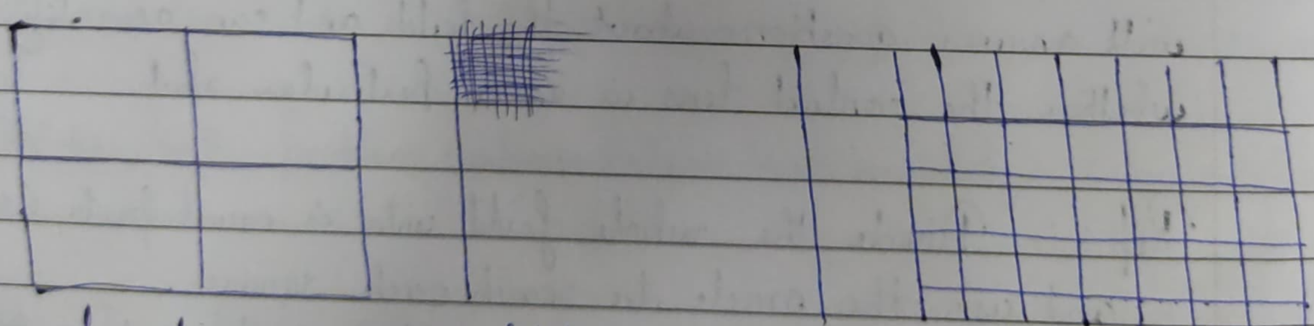
→ We run the algorithm and have the system scan every pixel for the lenses. The number of pixels will eclipse billions.

→ This approach will return highly accurate result but the time taken to perform the search will make this impractical.

→ This method is highly inefficient way of search and even an extremly powerful computer will take a very long time to locate the lenses.

Rep 3:- The best approach would be to divide the field up into a grid where each square is 1ft × 1ft and to eliminate all the squares from the grid that you know are nowhere near where your lost the lens.

R1 R2 R3



low time
less accuracy.

high accuracy
~~low~~ very high
search time

optimal "good enough"
time and
accuracy.

* Knowledge ~~is~~ may be represented as symbol structures representing bits of knowledge. Intelligent behavior can be achived through manipulation of symbol structures.

Date / /

Semantic Nets

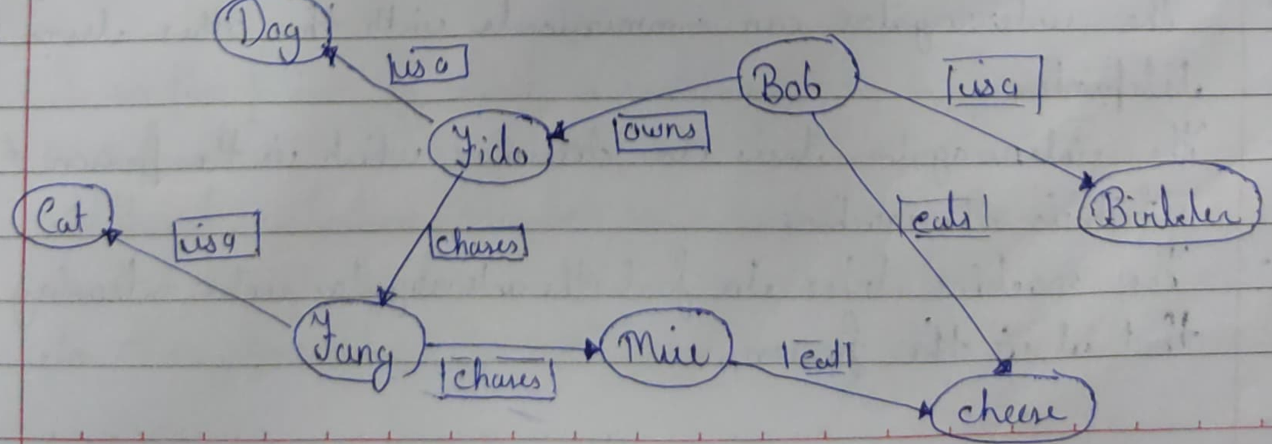
* A semantic Net is a graph consisting of nodes that are connected by edges. The nodes represent objects, and the links between nodes represent relationship between those objects.
→ The links are usually labeled to indicate the nature of the relationship.

* Semantic Nets provide a very intuitive way to represent knowledge about objects and the relationships that exist between those objects.
* The data in semantic nets can be reasoned about in order to produce systems that have knowledge about a particular domain.

* The specific objects are generally referred to as instances of a particular class.

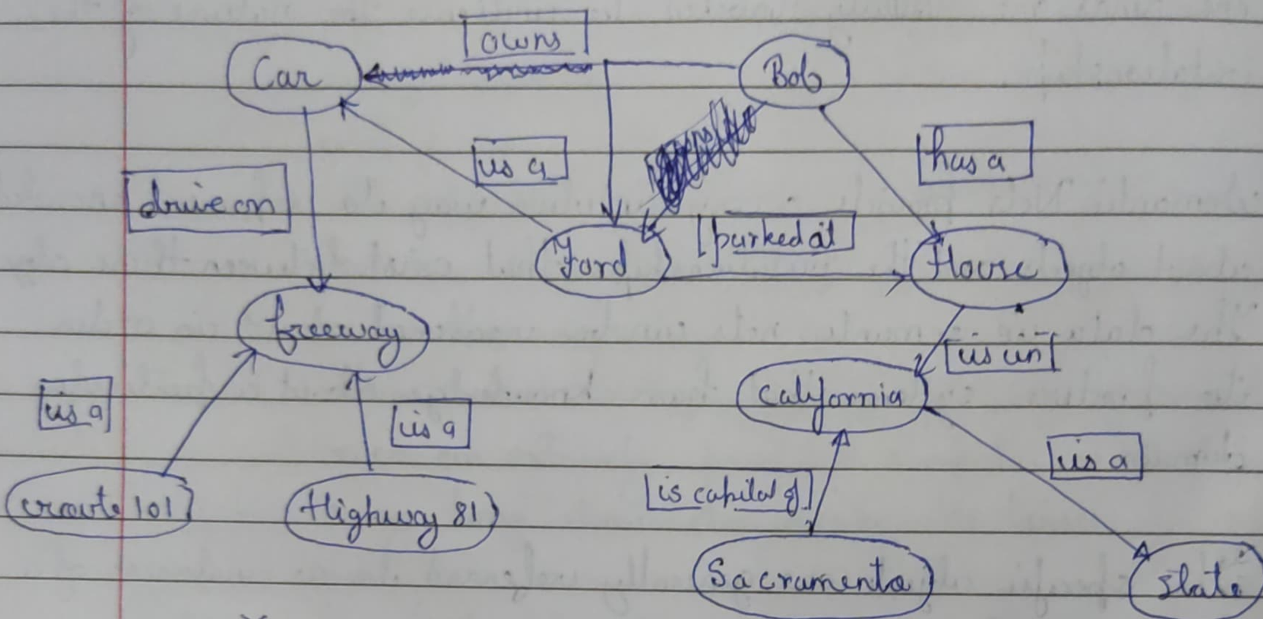
Disadvantages / drawbacks.
* Semantic nets ~~do not~~ cannot be used to represent negations
* Eg. If implemented for searching a word in a dictionary any attempts to search the graph will be fairly inefficient because it means visiting each node in turn until the desired node is found. (The words are nodes arranged in a chain with one connection from first node to the 2nd, & from 2nd to 3rd)

eg.



eg. 2.

A ford is a type of Car. Bob owns two cars. Bob parks his car at home. ~~The~~ His house is in california which is a state. Sacramento is the capital of California. Cars drive on the freeway, such as route 101 & highway 81



— x — x — x — x — x — x —

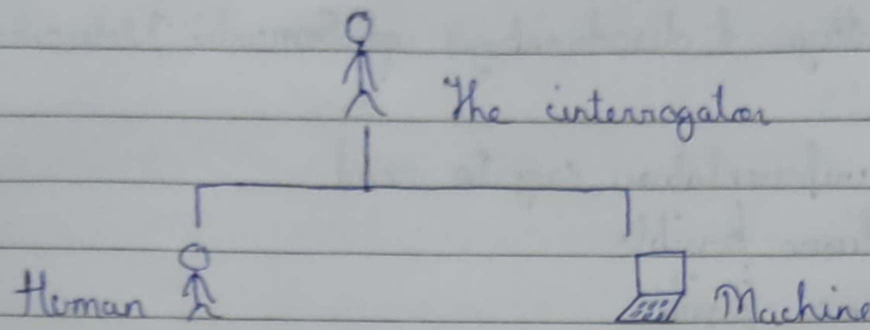
Turing Test

Alan turing invented the turing test, designed to determine if a computer system can be called an artificial intelligence or not based on whether it can fool a human into thinking it is human too.

Setup :-

- Three rooms contain a person, a computer and an interrogator
- The interrogator can communicate with the other two by teleprinter
- The interrogator tries to determine which is the person & which is the machine
- The machine tries to fool the interrogator into believing that it is the person.

- If the machine succeeds, then we conclude that the machine can think.



The turing test provides a basis for many of the schemes used to evaluate AI programs.

* Searle's "Chinese Room"

- A thought experiment used to argue against strong AI
- A non-chinese speaker is in a room with a set of cards with chinese characters, and a set of instructions in English.
- Questions in chinese are fed into the room, and by following the instructions, the human is able to produce answers.
- The room appears to understand chinese - it can answer questions in the language - but the human inside cannot

Conclusion Turing test works for "weak" AI, not strong AI.

—— x —— x —— x —— x —— x —— x —— x —— x -

* ~~Frames~~ Inheritance

It is the process by which a subclass inherits properties from a superclass.
Although inheritance maybe is a useful way to express generalities about a class of objects, in some cases we need to express exceptions to those generalities. In such cases we say that the default value has

been overridden in the subclass

Advantages & disadvantages of Semantic Networks

→ Adv
i) Simple representation, easy to read
ii) Associations possible
iii) Inheritance possible

→ Disadvantages
i) A seperate inference procedure (interpreter) must be build
ii) The validity of the inferences is not guaranteed
iii) For large networks the processing is inefficient.

* Frames

Frame based representation is a development of semantic nets and allows us to express the idea of inheritance.

As with Semantic nets, a frame system consists of a set of frames which are connected together by relations. Each frame describes either an instance or a class. Frames are thus object oriented representation that can be used to build expert systems.

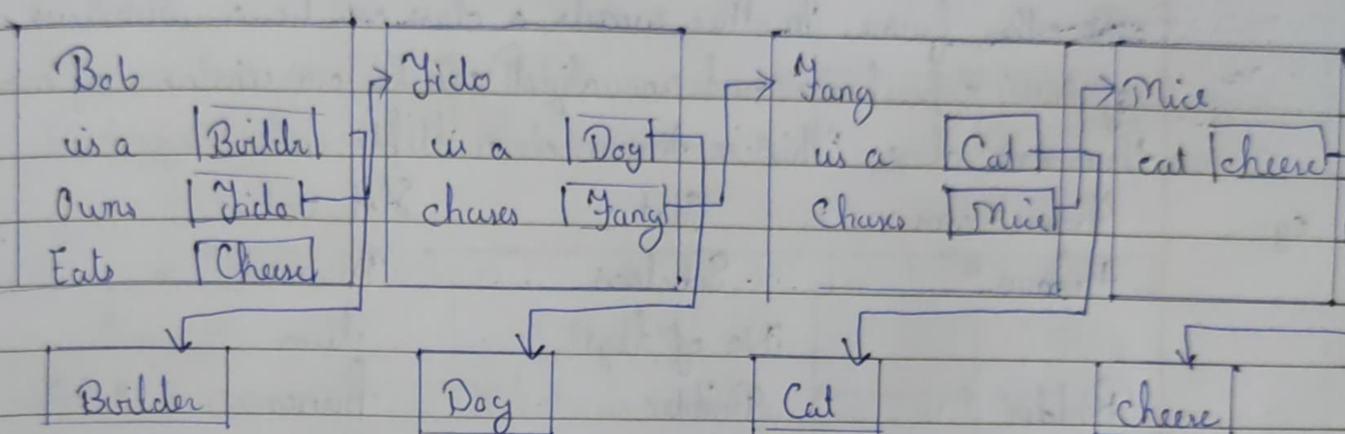
→ Each frame has one slot, which are assigned slot values.
→ Rather than simply having links between frames, each relationship is expressed by a value being placed in a slot.

Example

Frame name	Slot	Slot value
Bob	is a	Builder
	owns	Fido
	eats	Cheese
Fido	is a	Dog

	chases	Fang
Fang	is a	Cat
	chases	Mice
Mice	eat	Cheese
cheese		
Builder		
Dog		
Cat		

Representation in digramatic form



Types of Relationship

i) Generalization ii) Aggregation iii) Association

<u>Inheritance</u>

Example:

Frame Name	Slot	Slot value
Mammal	* number of legs	four
Dog	subclass	Mammal
Cat	subclass	Mammal
Fido	is a	Dog
	number of legs	three
Fang	is a	Cat

* - can be overridden

Slots as frames:

It is also possible to express a range of values that can take. A way to express this kind of restriction is by allowing slots to be frames

Eg

Frame Name	Slot	Slot Value
Number of legs	minimum value	1
	maximum value	4

Multiple Inheritance

It is possible for a frame to inherit properties from more than one other frame. In other words, a class can be a subclass of two superclasses and an object can be an instance of more than one class. This is known as multiple inheritance.

eg.

Frame Name	Slot	Slot value
Human	Subclass	Mammal
	No of legs	two
Builder	Builds	houses
Bob	is a	Human

In some cases, we will encounter ~~confits~~ conflicts, where multiple Inheritance leads us to contradictory info about a frame

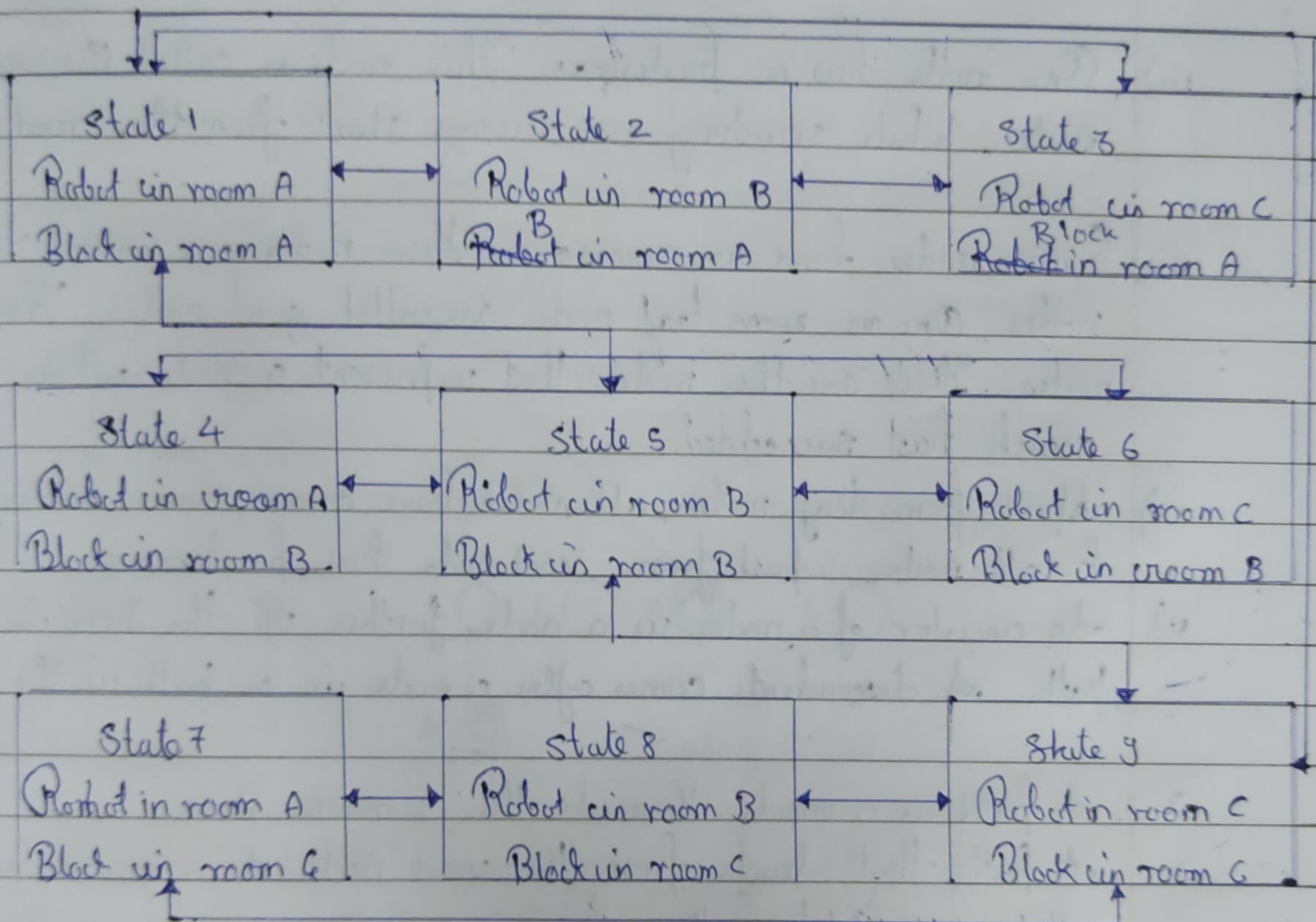
eg.

Frame name	Slot	Slot Value
Cheese	is	smelly
Thing wrapped in foil	is	not smelly
cheddar.	is a	cheese
	is a	Thing wrapped in foil

Search Spaces

- Many problems in AI can be represented as search spaces.
- A search space is a representation of the set of possibilities in a given choice, one or more of which are the solution to the problem.

Eg. Consider a Robot that lives in an environment with 3 rooms (room A, room B & room C) and with a block that he can move from one room to another.



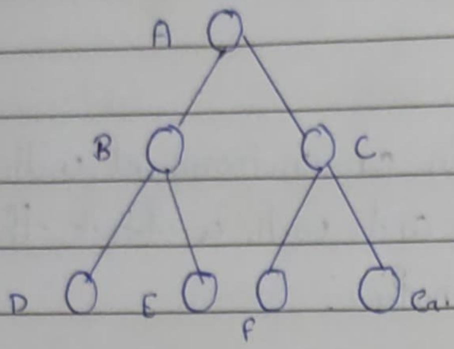
- The aim of most search procedures is to identify one or more goals and usually, to identify one or more paths to these goals.
- Because a search space consists of a set of states, connected by paths that represent actions, they are also known as state spaces.

Semantic Trees

Semantic trees are kind of semantic net with following prop.

i) Each node except for the root node has exactly one predecessor and one or more successor.

eg.



The non-symmetric nature of the relationship in semantic tree means that a semantic tree is directed graph.

ii) One node has no predecessors. This node is called the root node. While searching we always start from the root node.

iii) Some nodes have no successors. These nodes are called leaf nodes. One or more leaf nodes are called goal nodes. These nodes These are the nodes that represent a state where the search has succeeded.

iv) Apart from leaf nodes, all nodes have one or more successors & all nodes apart from root node has predecessor.

v) An ancestor of a node is a node further up the tree in some path. A descendent comes after a node in a path in the tree.

→ A path is a route through the semantic tree.

→ A path that leads from the root node to a goal node is called a complete path.

→ A path that leads from the root node to a leaf node that is not a goal node is called a partial path.

→ A cycle is a path through the net that visits the same node more than once.

→ An edge connecting two nodes is called a branch.

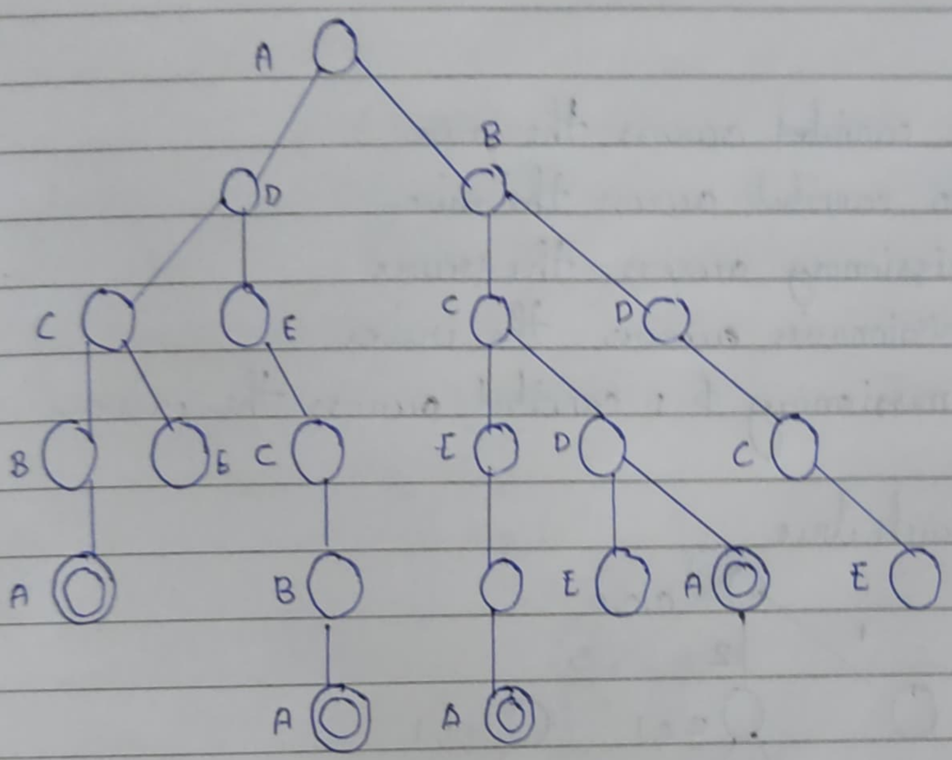
→ If a node has n successors, the node is said to have n branching

Factor

-> Successors of nodes of level n are at level n+1

Search Trees

* Used to represent search spaces.
* Root node has no predecessor.
* Leaf node has no successor.
* Goal nodes represents solution
* Type of a search tree that eliminates cycles.



Example Missionaries & Cannibals.

-> 3 missionaries & 3 cannibals

-> Need to cross the river using one canoe.

-> Canoe can hold upto 3 ppl.

-> Can never have more than cannibals than missionaries on either side of the river.

-> Aim : To get all safely across the river without any missionaries being eaten

A representation

→ Choose an appropriate representation.

→ Start State (cannibals, missionaries, canoes)

On side 1   On side 2

3, 3, 1   0, 0, 0

Since its a closed system, we just need to represent one side of the river.

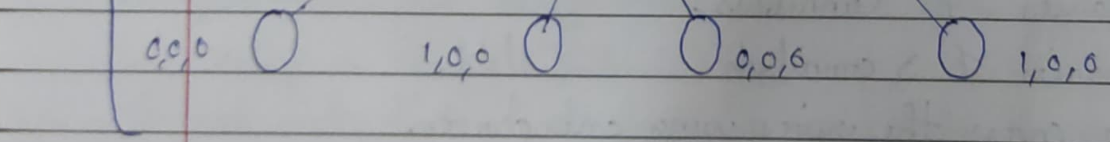
→ finishing side of the river, omit start side.

∴ Start side is 0,0,0.

Operators.

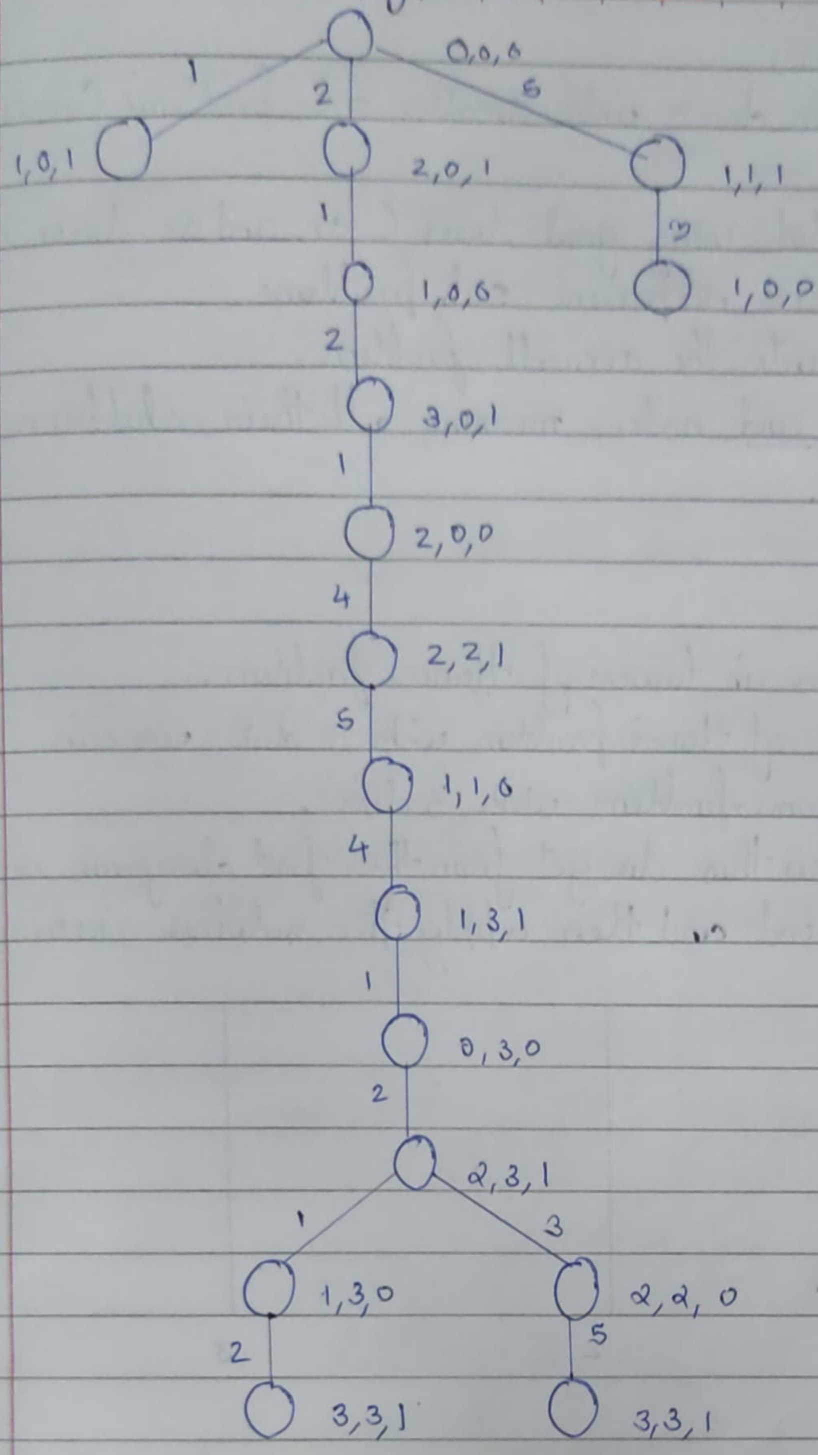
1. Move one cannibal across the river
2. Move two cannibals across the river
3. Move 1 missionary across the river
4. Move 2 missionaries across the river
5. Move 1 missionary & 1 cannibal across the river

→ Partial Search tree



A more effective representation for the problem is the one without any cycles. In this tree, we omit most repeated states. As well as avoiding cycles, we also removed suboptimal paths from the tree.

Search tree w/o cycles:



Combinatorial Explosion

- Problems that involve assigning values to a set of variables that can grow exponentially with the number of variables.
- This is the problem of Combinatorial Explosion.
- Some such problems can be extremely hard to solve (NP-Complete, NP-Hard).
- Selecting the correct representation can help to reduce this

Problem Reduction

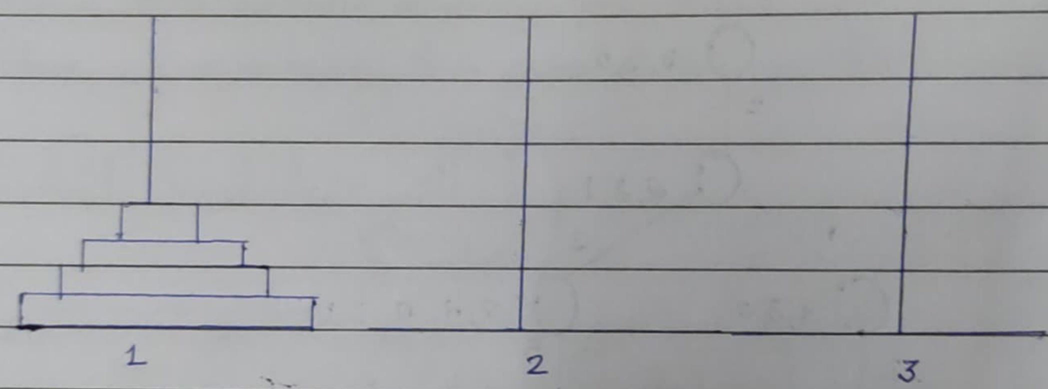
- Breaking a problem down into smaller sub-problems (or sub-goals.
- Can be represented using goal trees (or and or tree).
- Node in the tree represent sub problems.
- The root represents the overall problem.
- Some nodes are and nodes, meaning all their children must be solved.

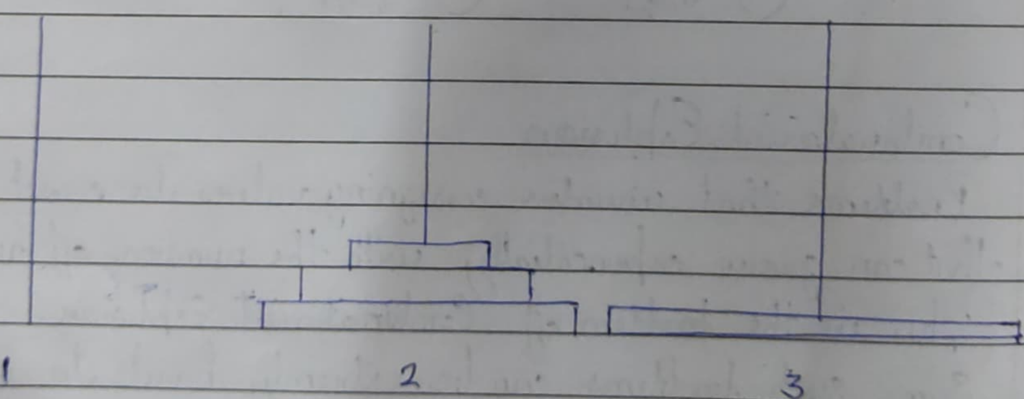
Example.

Problem Reduction in towers of Hanoi problem.

→ To solve tower of Hanoi problem with 4 disks, we can first solve the same problem with 3 disks.

→ The solution is thus to get from the first diagram on the left, to the second and then apply the solution recursively





Search tree Examples.

1) Design a suitable representation & draw the complete search tree for the following problem.

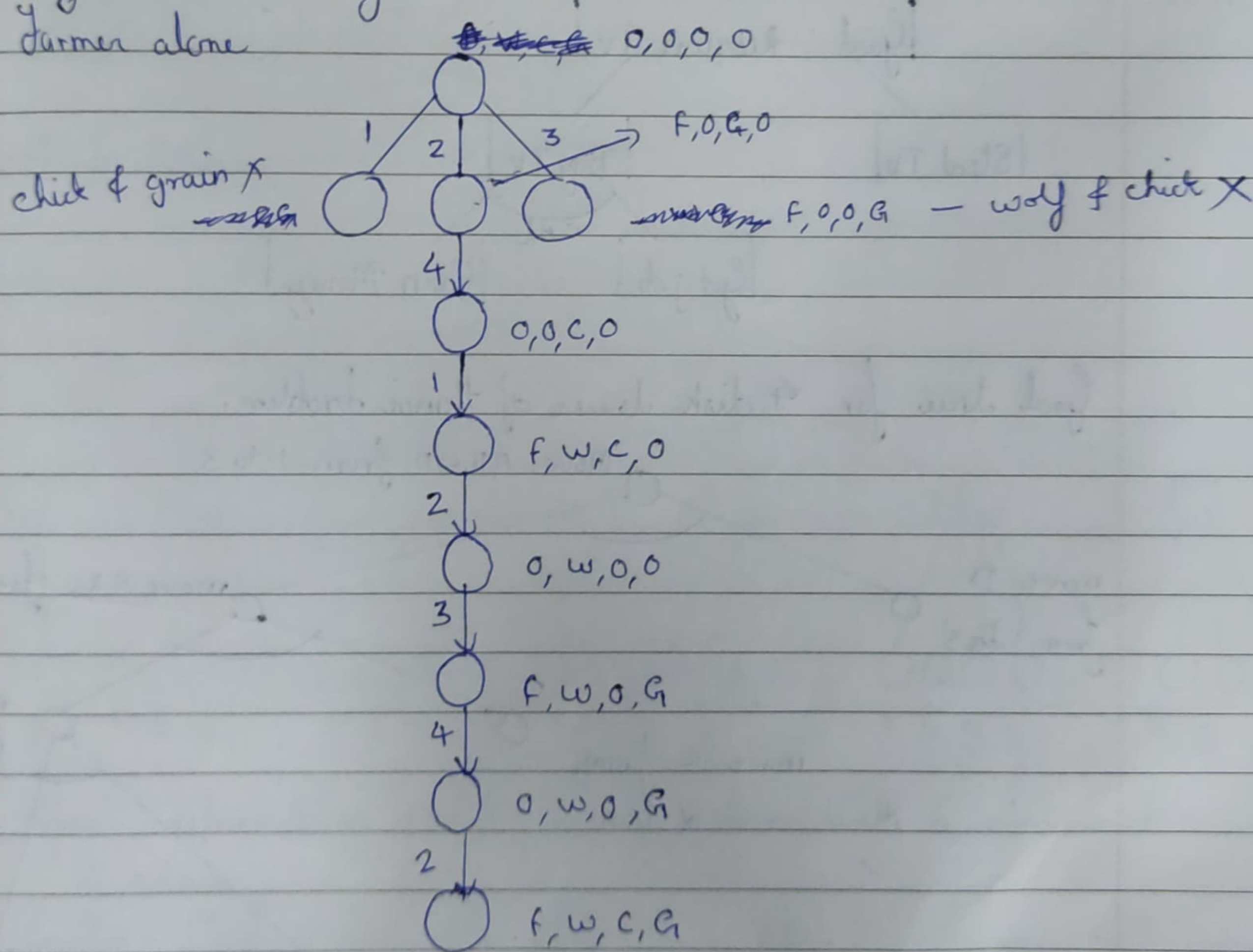
→ The farmer, wolf, chicken & bag of grain problem.

Side A				Side B			
Farmer	wolf	chick	Grain	Farmer	wolf	chick	Grain
1	1	1	1	0	0	0	0

→ ignore farmer, he'll have to move everytime.

Since the boat can hold only two items, we have only 3 operation

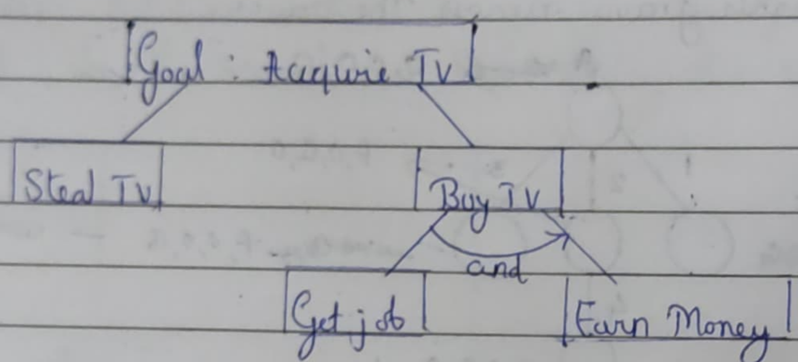
1) Farmer takes wolf accross the rive
2) Farmer take goat accross the river
3) & Farmer takes grains accross the river.
4) Farmer alone



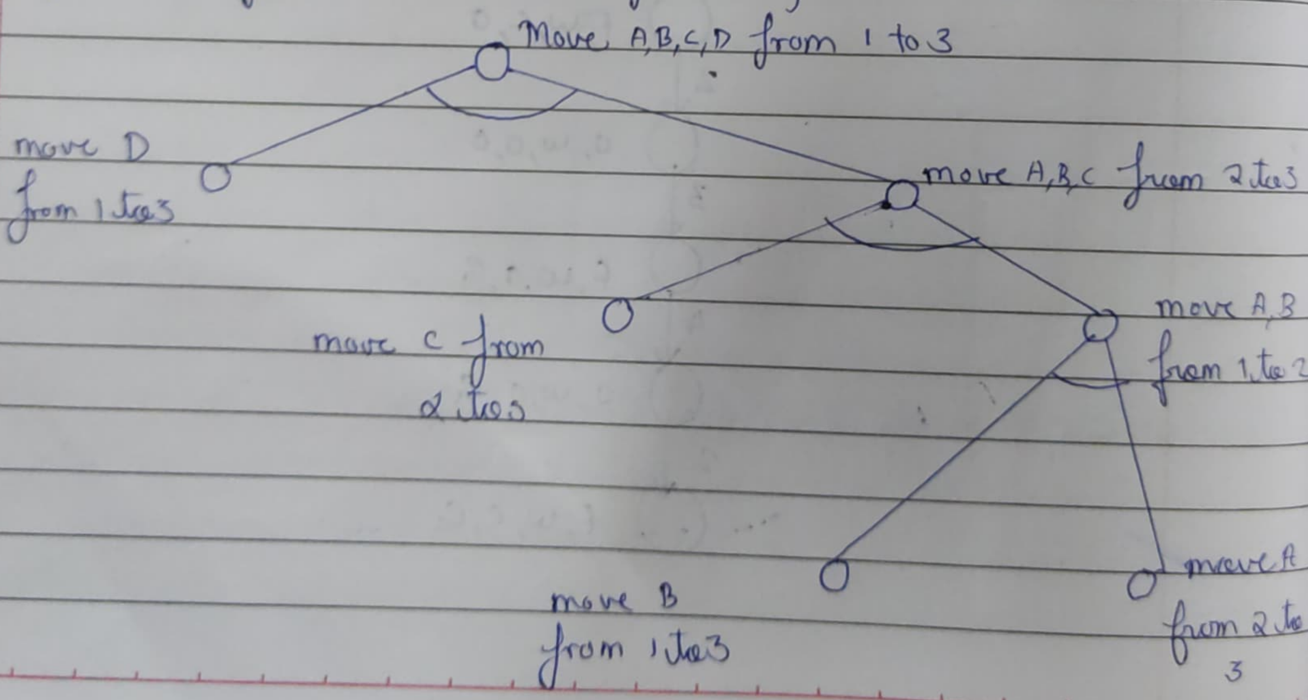
Goal Trees

- Also called and or trees
- Nodes in the tree represent sub problem
- The root node represents the overall problem.
- Some nodes are and nodes, meaning all their children must be solved
- An and node is shown by drawing an arc across the arcs that join it to its subgoals (children). Or nodes are not marked in this way.

And /or Tree



Goal tree for 4 disk tower of Hanoi problem.



⇒ Top down & Bottom Down.

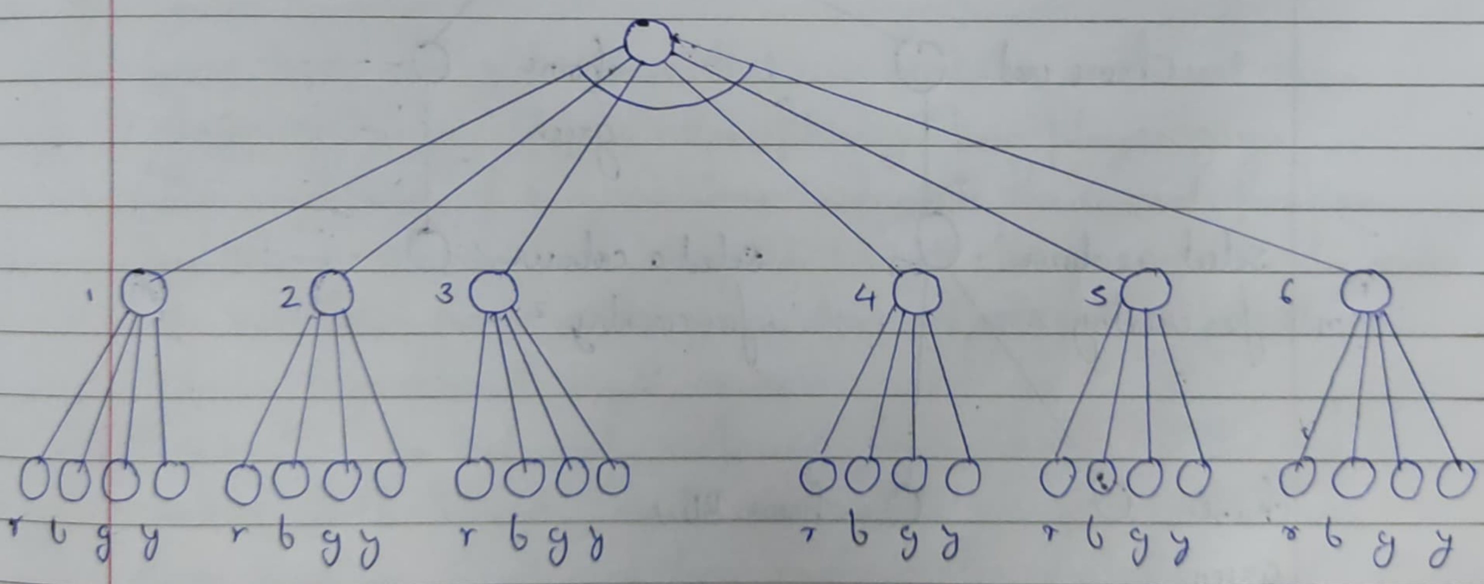
- There are two main approaches to breaking down a problem into subgoals - top down & bottom up.

- A top-down approach involves first breaking down the main problem into smaller goals and then recursively breaking down those goals into smaller goals, & so on, until leaf nodes, or success nodes, are reached, which can be solved.

- A bottom up approach involves first determining all of the sub-goals that are necessary to solve the entire problem, and then starting solve the success nodes, and working up until the complete solution is found.

* Applications of Goal Trees

1) Map Coloring



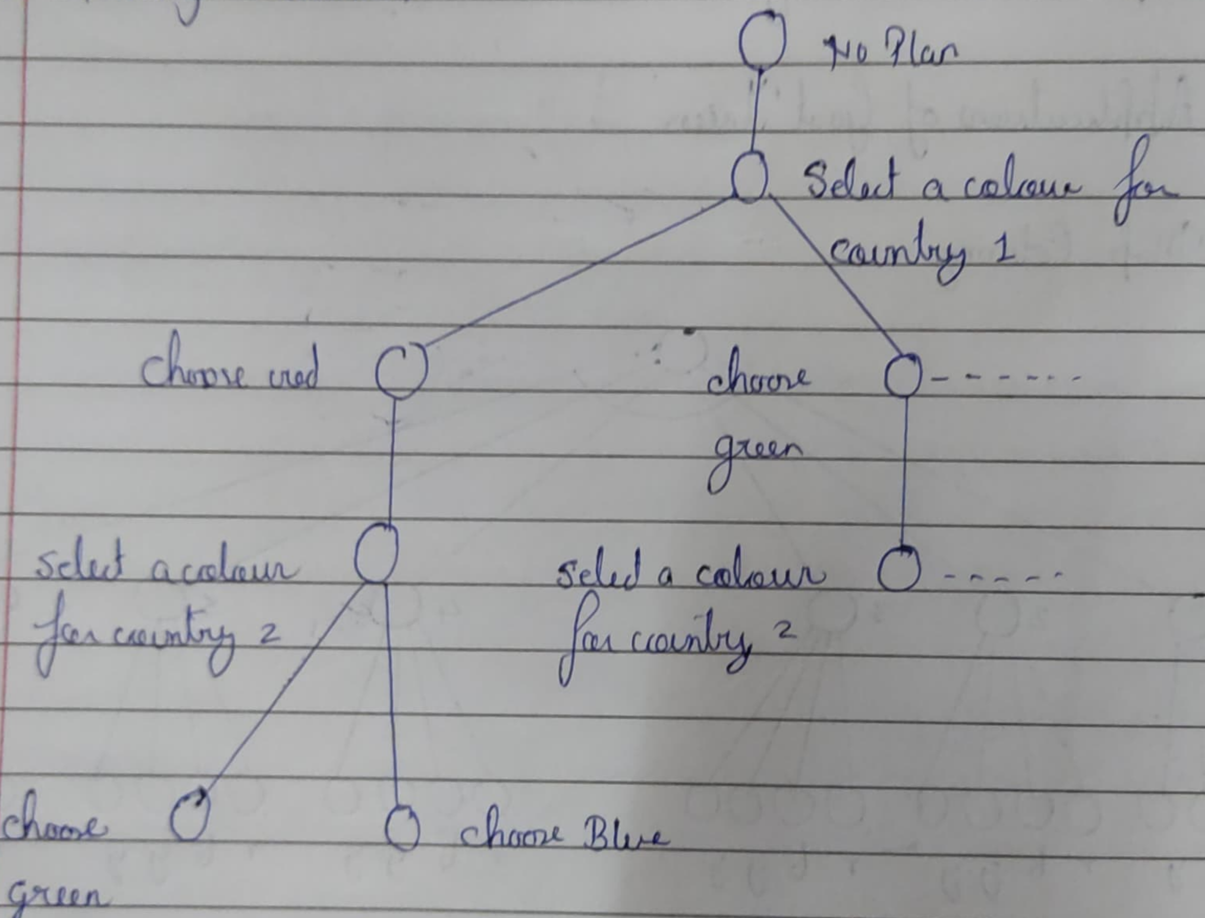
Goal representing a map coloring problem with 6 countries & 4 colours

→ Problem → Map colouring for 6 countries using 4 colours.

Constraints
No two adjacent may have the same colours.

→ To apply these constraints and apply a search method to the problem, the goal tree must be redrawn as a search tree because search methods are generally not able to deal with and nodes.
→ This can be achieved by redrawing the search tree as a search tree; where paths through the tree represent plans rather than goals

Proving Theorems.



partial Search tree for map colouring of 6 countries

2) Proving Theorms
→ It is an or node because there may be several ways to prove the theorm
→ The next level down consists of and node, which are leemmas that are to be proven.
→ Each of these leemmas again may have several ways to be proved so, therefore is an or node.
→ The leaf nodes of the tree represent axioms that do not need to be proved.

3) Parsing Sentences
→ A parser is a tool that can be used to analyze the structure of a sentence in the english language.
→ Sentences can be broken down into phrases and phrases can be broken down into nouns, verbs, adjectives and so on.

4) Games
→ Game trees are goal trees that are used to represent the choices made by players when playing two player game.
→ The root node of a game tree represents the current position and this is an or-node because I must choose one move to make.
→ The next level down in the game tree represents the possible choices my opponent might make.
→ Eventually the leaf nodes represent final positions in the game and a path through the tree represents a sequence of moves from start to finish resulting in a win, loss or a draw.
→ This kind of tree is a pure and or tree because it has an or node at the top, each or node has and node as its direct sucessors and each and node has or nodes as it direct sucessors.
→ A pure tree shouldn't have constraints.

Page No.
Date / /

Stuff I missed.

1. Procedures (right after multiple inheritance)
→ A procedure is a set of instructions associated with a frame
- for example, a slot reader procedure might return the value of a particular slot within the frame
- Another procedure might insert a value into a slot (a slot writer)
- Another important procedure is the instance constructor, which creates an instance of a class
- Such procedures are called when and so are called WHEN NEEDED procedure
• The procedure can be run upon request.
~~Demons~~

Demons
• A demon is a procedure that is run automatically, usually triggered by an event such as when a value is:
- Read: when a particular value is read. (WHEN READ procedures)
- Written: when the value of a slot is changed. (~~WHEN~~ written
- Created
- Changed. WHEN - CHANGED

Self Study BS
1) ~~OOP~~ Object Oriented Programming
→ done in OO ~~lan~~ programming languages like Java; C++.
→ Uses ideas such as:
→ Inheritance
→ Multiple inheritance
→ Overriding default values.
→ procedures & demons.

· Languages such as IBM's APL 2 use a frame based data structure.

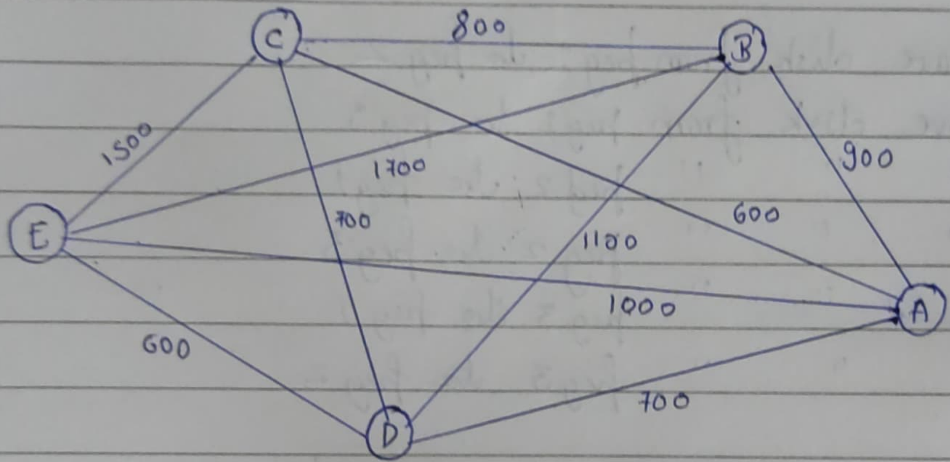
Some More search tree Examples

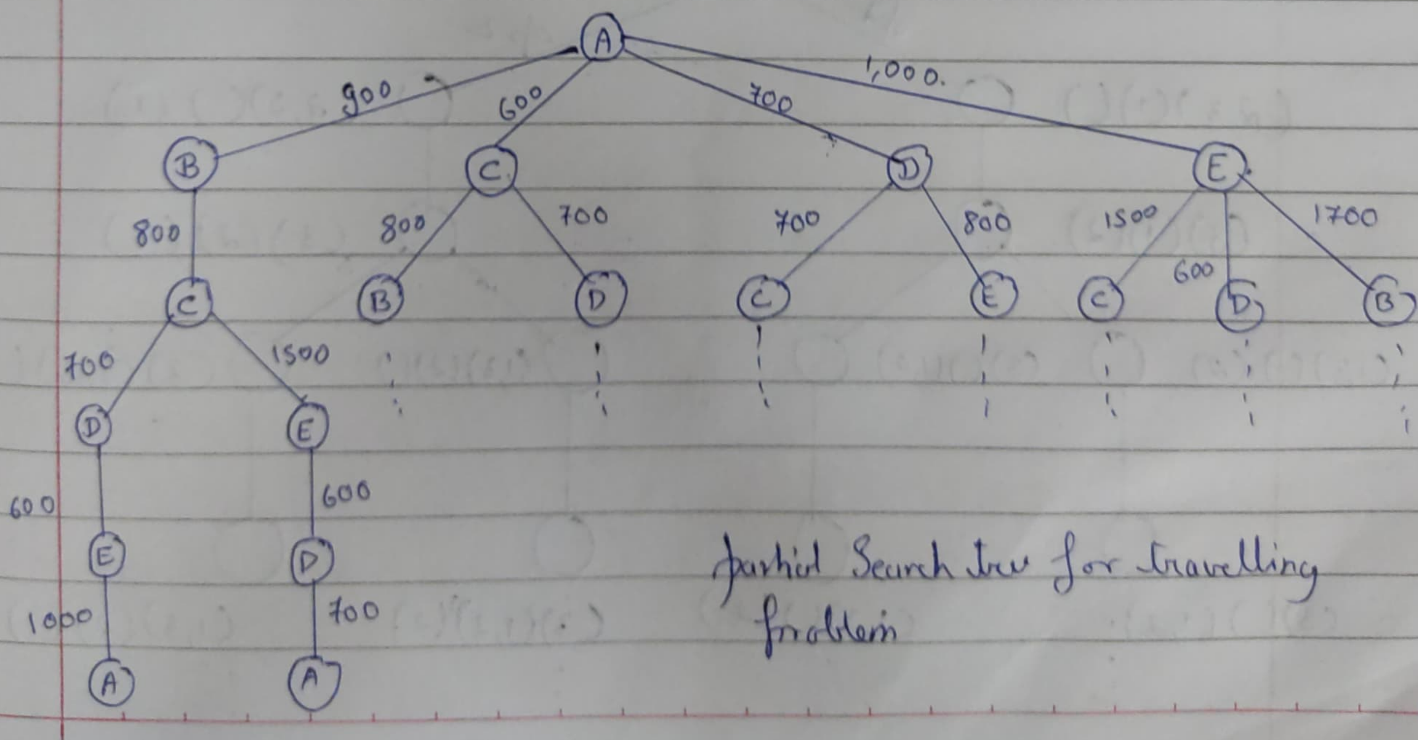
1) Travelling Salesman

A salesman must visit each of a set of cities and then return home. The aim of the problem is to find the shortest path that lets the salesman visit each city

eg A → Atlanta B Boston C Chicago
D. Dallas E - El Paso

Start → Atlanta, visit the remaining 4 cities





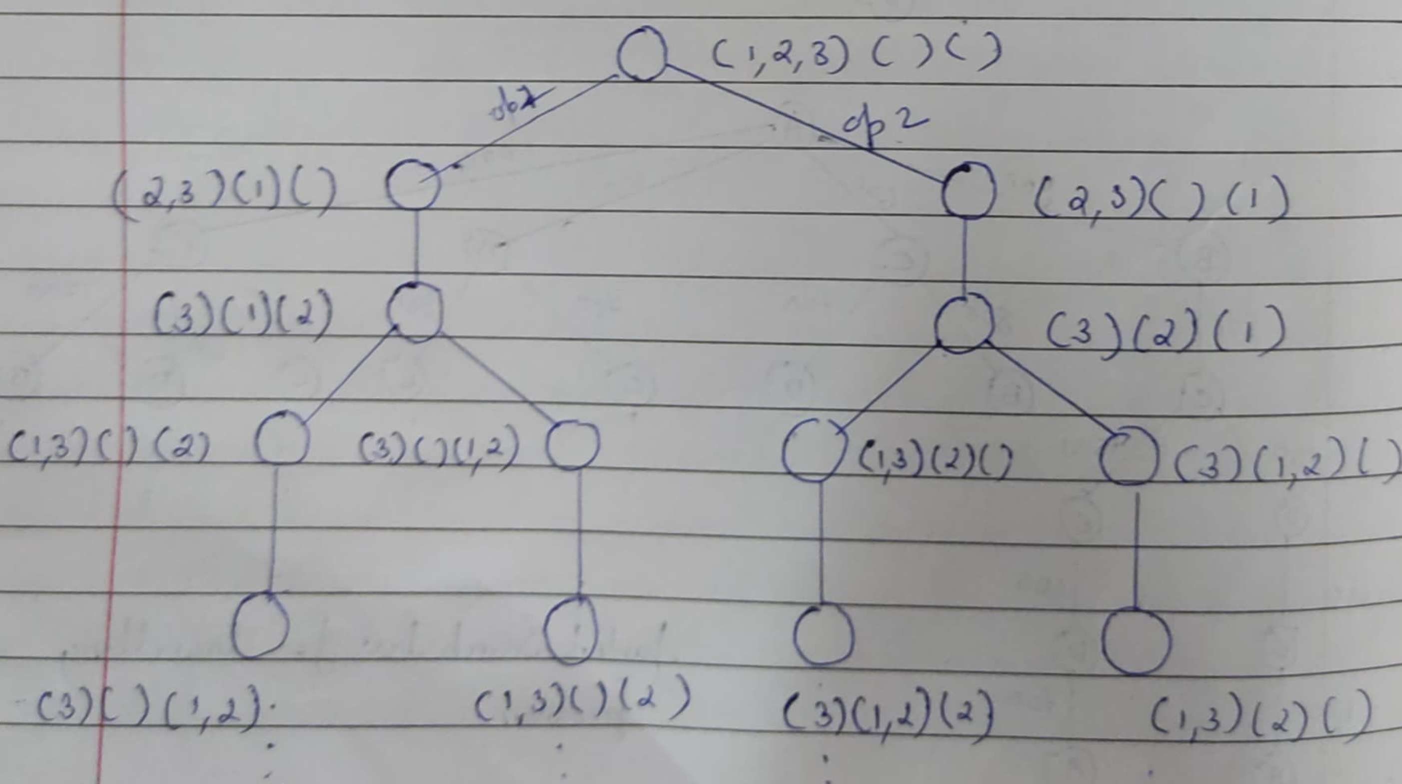
partial Search tree for travelling problem

→ In total there will be (n-1) possible paths for a TSP with n cities.
→ But for large n, the brute force search method
→ To solve search problem with large trees, we use heuristics
→ Here we just follow the closest path from each city, called nearest neighbor heuristic.
→ It gives efficient but not the ~~most~~ best result

2) <u>Towers of Hanoi</u>

Ok we know the problem duh!!!

op 1 Move disk from peg 1 to peg 2
op 2 Move disk from peg 1 to peg 3
op 3 " " " peg 2 to peg 1
op 4 " " " peg 2 to peg 3
op 5 " " " peg 3 to peg 1
op 6 " " " peg 3 to peg 2



& so on.