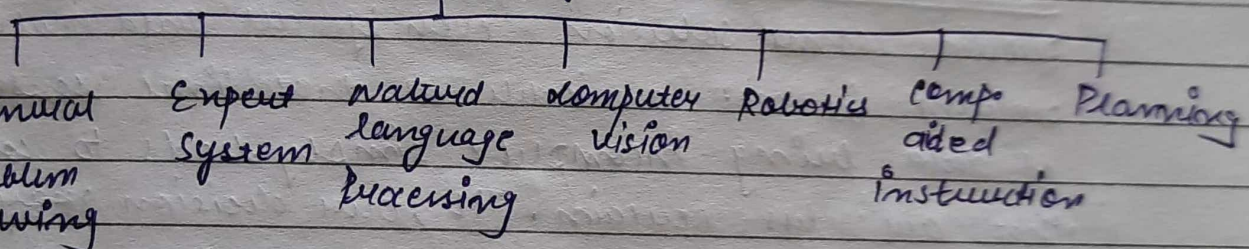
Date : 15/9/22
Page No.

Principle of AI

- Definition of AI :-
AI is the part of CS concern with designing intelligent system i.e., system that exhibit the characteristic associate with intelligence in human behaviour.
- AI contains two terms :-
1.) Intelligence 2.) Artificial devices.
- A system with intelligence is expected to behave as intelligence as a human being.
Secondly a system with intelligence is expected to behave in best possible manner.
- In 1956 AI is invented by John McCarthy.

Applications of AI



20/9/22

(a) Symbolic Representation -
- AI programs mainly deals with non-numerical symbols & contrast strongly with commonly accepted view that computers can deal.
- AI programs perform numerical calculations also when necessary but their significance as symbolic value with generally enter into reasoning process performed by program.
→ The importance of physical symbolic hypothesis are as follows :-
- It is significant theory of human intelligence and as such is of great ~~interest~~ interest to the psychologist.

- It forms basic philosophy that it is possible to built a prog which can perform intelligent task as performed by human.

→ Knowledge Representation:-

1. Knowledge can present as a modular (use can learn into supports) and there is a correspond b/w external world & symbolic representation system.

- This knowledge can be studied & understood in human terms bcoz symbols used for its representation are numerical.

2. Incomplete Data:- AI program has capability to provide some solution even if all data relevant to problem is not available at that time consequence of data being incomplete simply leads to conclusion that are less certain. IRL, our decision may be wrong in absence of relevant data's. It can be happen that absence of complete data is inherent in the problem as in game's bridge.

3. Conflicting Data:- Data items can even be contradicting to each other, this type of data is called conflicting data or data corrupted by error.

* Ability to learn:-

Intelligent computer have ability to learn from their mistake so that they can improve their performance by taking account of past errors. This is related to capacity

Date : / /
Page No.

for generalising to drawing analogy & discarding selective info.
The problem of given computers the learning ability similar to that of human being is that of stimulating the machine process the working of human minds never be programed in the machine so that no computer even will have as mind of its own.

History of AI

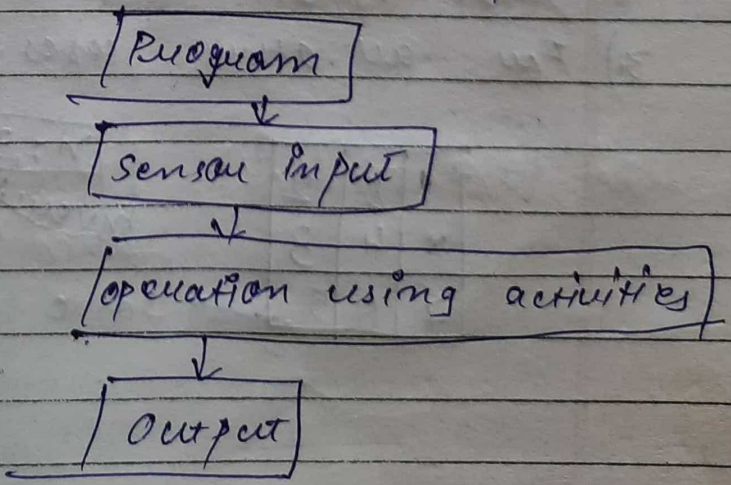
First Generation	Pre - 1950
second Generation	1950 - 1970
Third Generation	1970 - present.

Eg. (First Generation)

1. Can a self-repairing machine be built which locate repair.
2. Can a digital computer be programmed to program itself.
3. using mechanical concepts can be computer ~~prg~~ program.

22/9/22

Agent



Date : / /
Page No.

There are 3 parameters for an intelligent system.

1.) knowledge base.
2.) operators.
3.) search technique.

State space problems :-

A representation of all possible state for a given problem is called SSP.

Inital

2	1	3	4
5	6	7	8
9	10	11	12
13	14	15	

Goal state

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

1) If n is odd then puzzle is always solvable if no. of inversion is even in the input state.

2) If n is even puzzle is solvable in two condition the blank is on an even row counting from bottom like second last and fourth last and no. of inversion is odd.
- The Blank is on an odd row counting from bottom and no. of inversion is even.

3) For all other cases puzzles is not solvable.

1	8	2
x	4	3
7	6	5

→ solve this by Bubble sort.

n = 3
inversion count = 10

Date : / /
Page No.

Ques

3	9	1	15
14	11	4	6
13	x	10	12
y	7	8	5

Inversion count = 56

~~In~~ solve not solveable. (By 2(i) condition)

Ans

If Inversion count = 41 (solveable)

as n = even, inversion count = ~~even~~ Odd, even number row from bottom.

★ Architecture of AI Machine.

- Knowledge
- Interference process or control strategy
- Expert tools.
- Hardware & software.

26/09/22

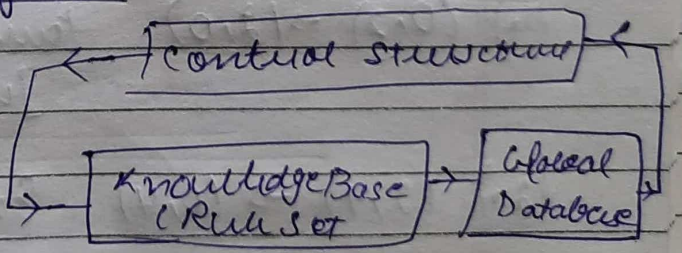
★ Production system.

→ Rules of Production.

- Powerfull knowledge representation.
- Bridge b/w AI Research to Expert System.
- Heuristic Model for Human Machine
- Strong data-driven nature.
- New Rules can easily added to system.

→ Architecture of Production system

- Knowledge Base
- Global Data Base
- Control Structure



Problems statement / representation

Transforming a problem into component of production System is called problem representation

27/9/22

Date : / /
Page No.

1.) → (x,y) if $x < 4 \rightarrow (4, y)$ → full 4 gallon jug.

2.) → (x,y) if $y < 3$ $(x, 3)$ → full 3 gallon jug.

3.) → (x,y) if $x = d$ & $d > 0 \rightarrow (x-d, y)$ Pour some water out of 4 gallon jug.

4.) → (x,y) if $y = d$ & $d > 0 \rightarrow (x, y-d)$ Pour some water out of 3 gallon jug

5.) (x,y) if $x > 0 \rightarrow (0, y)$ → Empty the 4 gallon water on the ground.

6.) (x,y) if $y > 0 \rightarrow (x, 0)$ → Empty the 3 gallon water on the ground.

7.) (x,y) if $x + y \leq 4$ and pour water from 3 gallon $y > 0 \rightarrow 4, (y+4-x)$ jug into it ~~x>0 → 4, (y+4-x)~~ 4 gallon jug is full.

8.) (x,y) if $x + y \geq 3$ Pour water from 4 gallon and $x > 0 \rightarrow [x-(3-y)]$ into 3 gallon jug until it is full.

9.) (x,y) if $x + y \leq 4$ and Pour all water of 3 gallon $y > 0 \rightarrow [0, x+y)$ into 4 gallon.

10.) (x,y) if $x + y \leq 3$ and Pull all water from 4 gallon $x > 0$ $(0, x+4)$ to 3 gallon.

11.) $(0,2) \rightarrow (2,0)$ Pour 2 gallon water from 3 gallon jug to 4 gallon jug.

12.) $(2,4) \rightarrow (0,4)$ Empty the 2-gallon in the 4-gallon jug on the ground.

Date: / /
Page No.

x	y	Rule
0	0	
0	3 →	2
3	0 →	9
3	3 →	2
4	2 →	7
0	2 →	5 or 12
2	0 →	9 or 11

x	y	
0	0	
4	0	→ [illegible]
1	3	→ 8.
1	0	→ 6
0	1	→ 10
4	1	→ [illegible]
2	3	→ 8..

★ Algorithm for Breadth first search. → Queue data structure.

1.) Create a variable called Node list and set it to inital state.

2.) Until a Goal State is found or NODE-list is empty do.

2.1) Remove first element from NODE-list and call it E. If NODE-list empty quit.

2.2) For each way that each rule can match the state described in E do.

2.2.1) Apply the rule to generate a new state.

2.2.2) If the new state is a goal state quit and return this state.

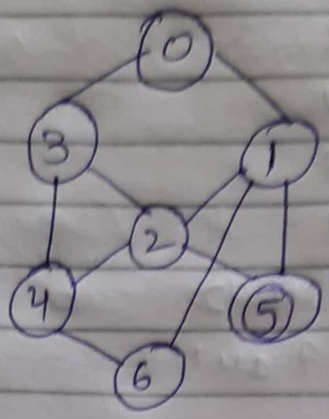
2.2.3) otherwise add new list to end of NODE-list.

Path is terminated.

→ Reaches dead end

→ Produces previous state

→ Becomes longer than limit.



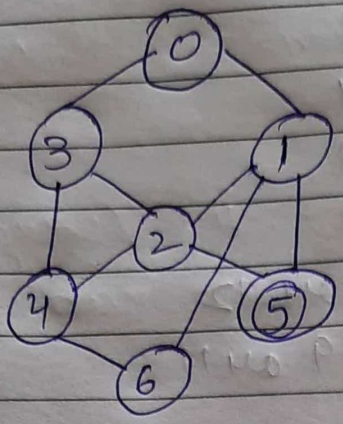
Back tracking :-

Path is terminated

→ Reaches dead end

→ Produces previous state

→ Becomes longer than limit.



Back tracking :-