Dynamic Programming - optimization technique.

- It is a general method for optimization that involves storing partial solution to problems so that a solution that has already been found can be retrieved rather than being recomputed.

- Fibonacci Series. - 0, 1, 1, 2, 3, 5 $f(n) = f(n-1) + f(n-2)$

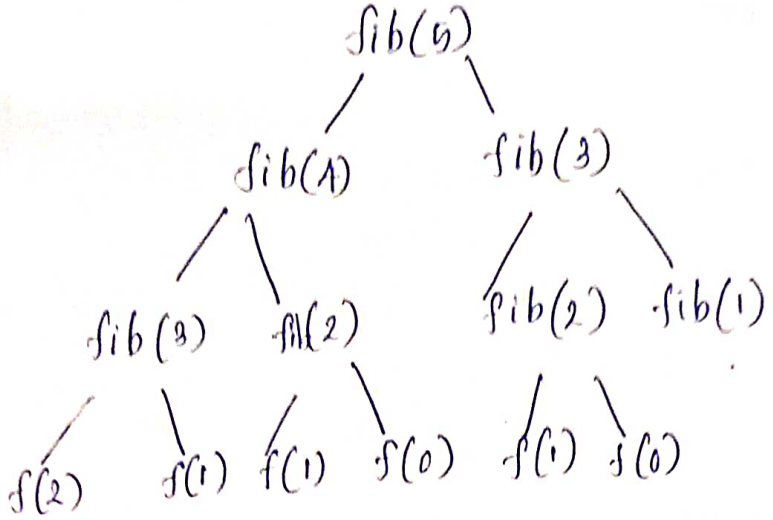
Recursive function to solve this.

```
public class Fibo {
public static void main (String args [])}.
int a=0;
int b=1;
int c;
S.o.p ("Enter the limit:");
DataInputStream dis = new DataInputStream (System.in).
```

---

n=4

```
#include<stdio.h>
{ int n;a=0; b=1;s;
  printf ("Enter the limit:");
  scanf ("%d" &n);
  int s = a+b;
```




```
#include <stdio.h>
int fibonacci (int)
int main() {
int terms;
printf ("Enter terms:");
scanf("%d", &terms);
for (int n=0; n<terms; n++)
{pf("%d", fibonacci(n));
 }
return 0;
}
int fibonacci (int num)
{
if (num==0 || num==1).
  {
   return num;
  }
else
{return fibonacci (num-1) +
    fibonacci (num-2);
  }
}
```

---

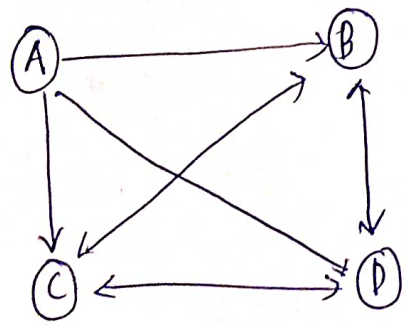
o/p → n=4.

0 1 1 2.

Write the program in dynamic programming.

```
#include <stdio.h>
int main()
{ int a[];
  int i,n
    int a[0] = 0; int a[1] = 1;
Pf("Enter the array elements: ");
limit
Sf ("%d", &n);
Pf("Enter the numbers: ");
Sf for (i=2; i<n; i++)
  {
     a[i] = a[i-1] + a[i-2]
  }.
```

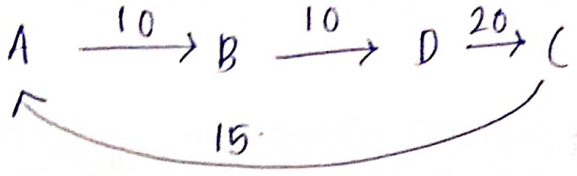
Travelling Sales Person Problem (TSP problem)



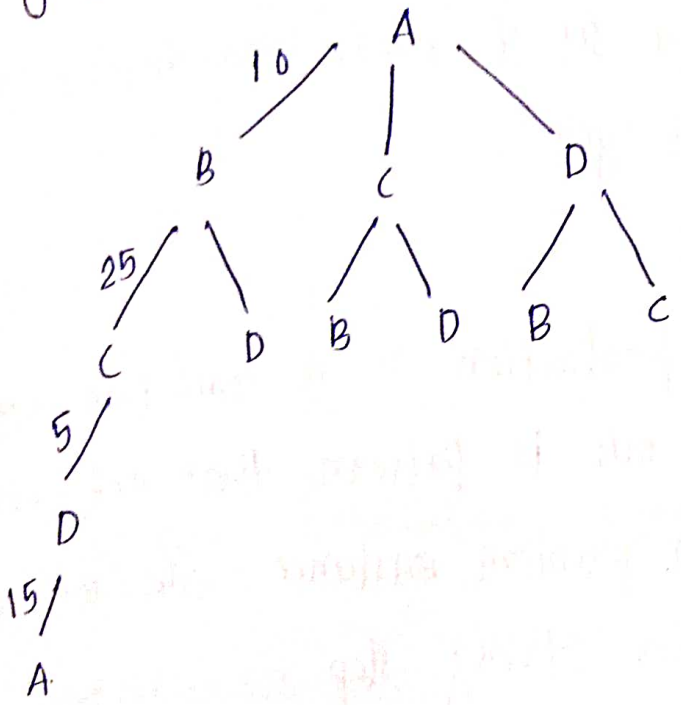
Objective: travelling from hometown to another city with min. travelling cost. with travelling all the cities exactly ones.

	A	B	C	D
A	0	10	15	20
B	5	0	25	10
C	15	30	0	5
D	15	10	20	0

Greedy Approach -



Group force search -



. Refer how to solve a TSP problem using the dynamic programming

Module-3

Production System

- is a model of computation that has proved particularly important in AI.
- The production system consist of a set of production rule, a working m/y and a recognize-act cycle.

defined by:

1) The set of production rules: A production is a condition-action pair. The condition part of the rule is pattern that determines when that rule maybe applied to a problem instance. The action part defines the associated problem-solving step.

2) Working m/y: Contains the current state of the problem.

3) Recognize-act cycle: Working m/y is initialized with the beginning problem description. The current state of the problem solving is maintained as a set of patterns in working m/y. This patterns are matched against the conditions of the production rules; this produces a subset of the production rules, called the conflict set.

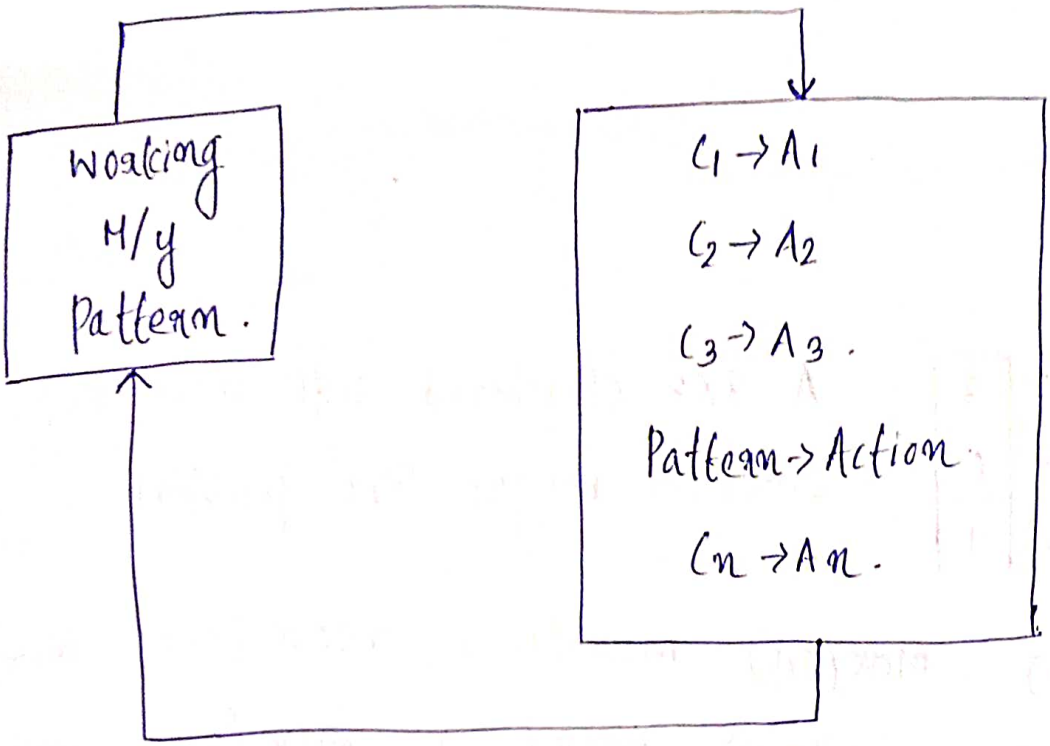


figure 1: Production System.

Production set:

1) ba → ab
2) ca → ac
3) cb → bc

Start state

2	8	3
1	6	4
7		5

Goal state

1	2	3
8		4
7	6	5

Condition — Action.

Goal state in working m/y → halt.
blank is not on the left edge →

1	2	3
4	5	6
7	8	9

A 3x3 chessboard with move rules for the simplified knight tour problem.

move (1,8) Move (6,1) Move (1,6) move (6,7) move (2,9)
move (7,2) move (2,7) move (7,6) move (3,4) move (8,3).
move (3,8) move (8,1) move (4,9) move (9,2) move (4,3)
move (9,4)

- Move the knight from 1 to 3.

Condition. Action.

knight on square 1 → move knight to square 8.
" " " " 6.

Iteration	Working m/y		Conflict set (rule #'s)	Fire rule.
	current state	goal square		
0	1	2	1,2	1
1	8	2	13,14	13
2	3	2	5,6	5
3	4	2	7,8	7
4	9	2	15,16	15
5	2	2		halt.

Trace of a simple production system.

Iteration #	Working M/y	Conflict set	Rule fired.
0	cbaca	1, 2, 3	1
1	cabac	2	2
2	acbca	2, 3	2
3	acbac	1, 3	1
4	acabc	2	2
5	aacbc	3	3
6	aabcc	ϕ	halt

Advantages of Production System for AI

The major advantages of production system include:

1) Seperation of knowledge and control: ease of modifying the knowledge base without requiring a change in the code for program control.

2) Natural mapping onto state space search: Simplifies the implementation, debugging, and documentation of search algorithm.

3) Modularity of production Rule:

4) Pattern-Directed Control: rules in a production system may fire in any sequence

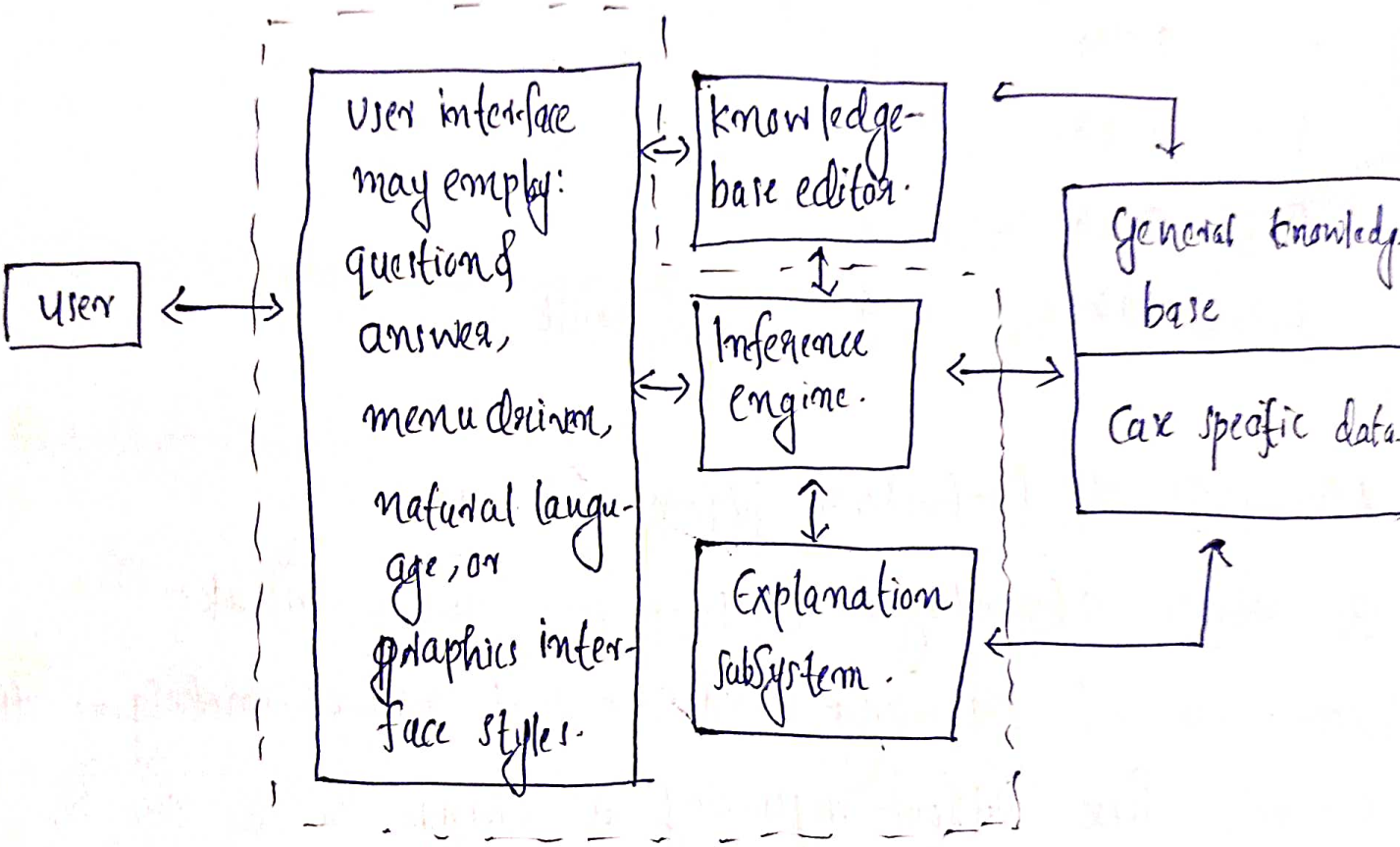
5) Opportunities for Heuristic Control of search.

6) Tracing & Explanation.

7) Language Independence.

Mycin- exp. sys. in healthcare

Expert System Technology.



The heart of the expert system is the knowledge base, which contains the knowledge of a particular application domain. In a rule-based expert system, this knowledge is expressed in the form of if-then rule. The knowledge base contain both general knowledge as well as the case specific information. The Inference engine applies the

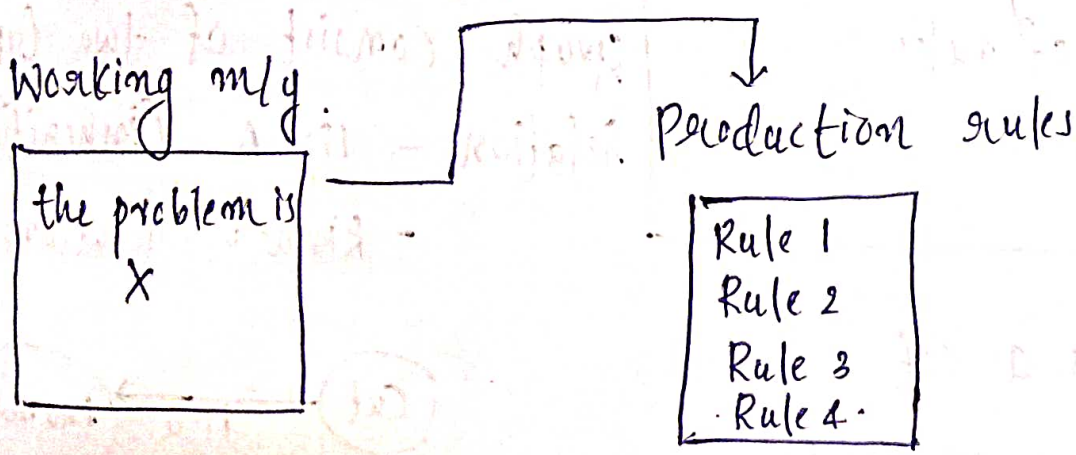
knowledge to the solⁿ of actual problem.

Eg: Rule 1:

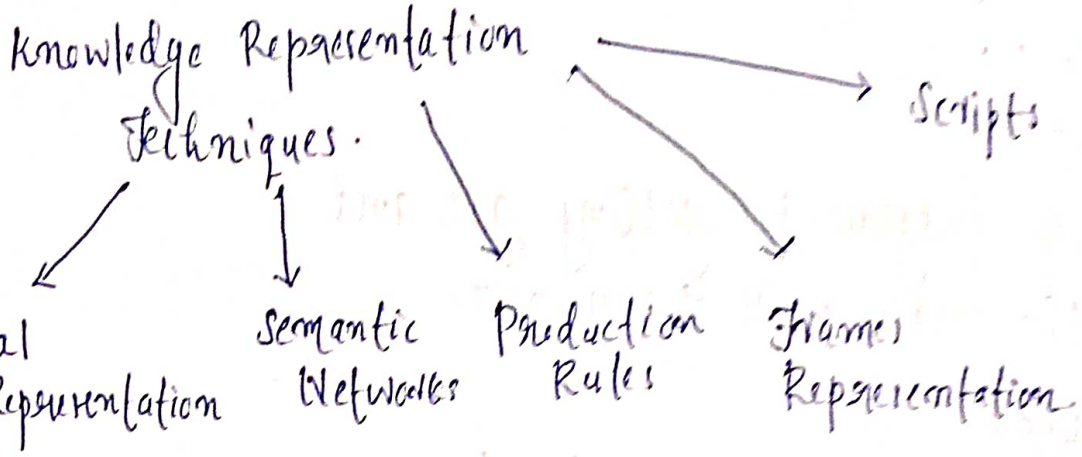
if
the engine is getting gas, and
the engin will turn over,
then
the problem is spark plugs.

Rule 2:

If
the engine does not turn over, and
the lights do not come on
then
the problem is battery or cables.



Knowledge Representation



Logical Representation.

- Predicate Calculus.
- Propositional Calculus.

Production Rule

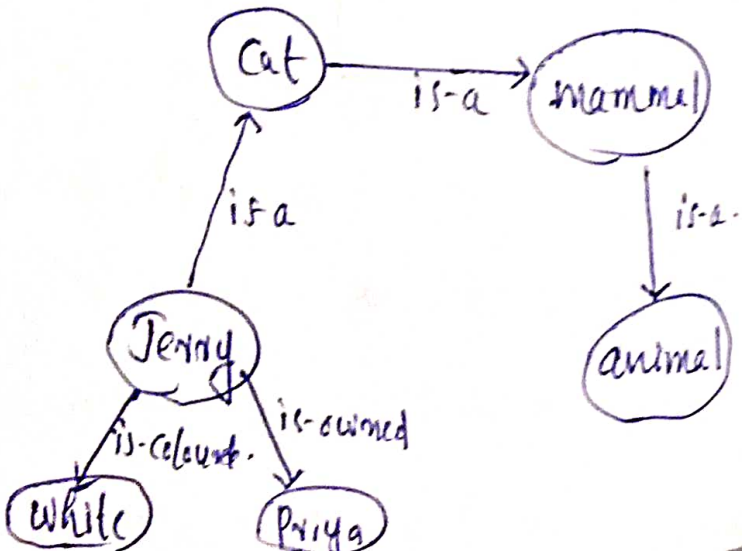
- If the knowledge is represented in the form of rules.
- Give examples.

Semantic Network

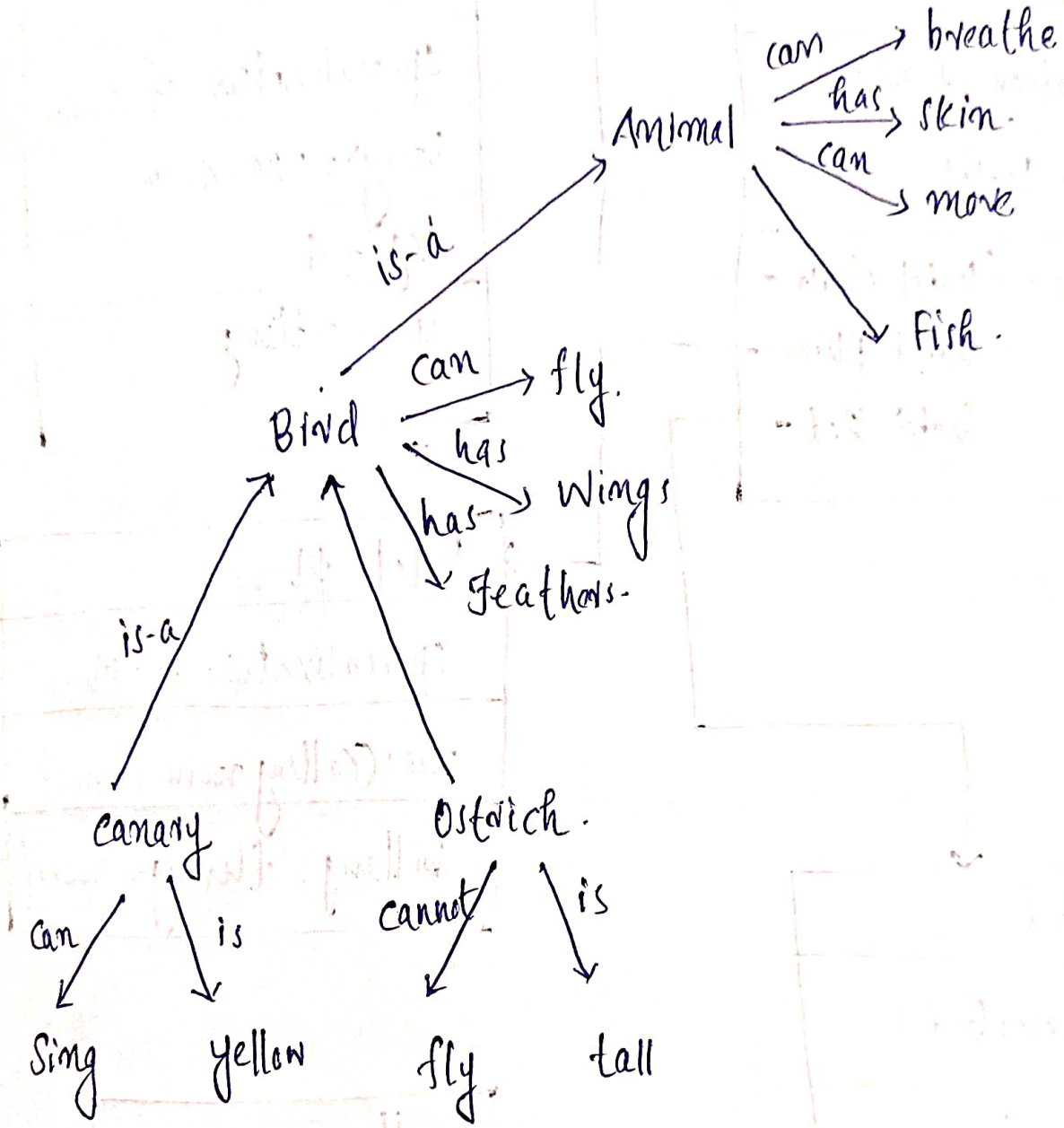
- In semantic network we can represent our knowledge in the form of graphical network. This network consist of nodes. representing object and arc which describe the relationship b/w those objects.

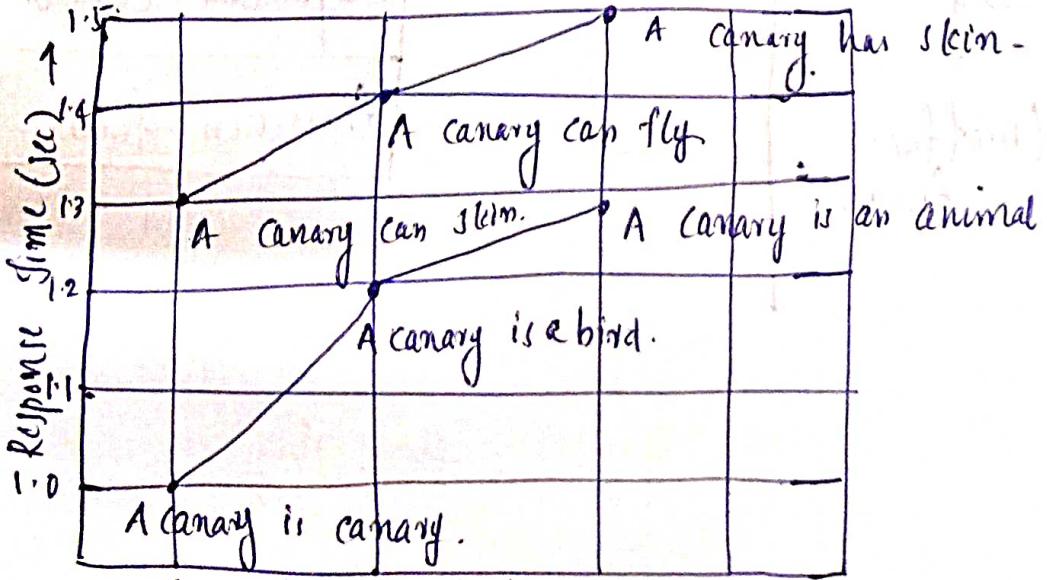
Graph consist of two types of relation - is-A (inheritance).
- kind-of-relation.

Eg:- Jerry is a cat
Jerry is a mammal
Jerry is owned by Priya
Jerry is ~~brown~~ white coloured
All mammals are animal.

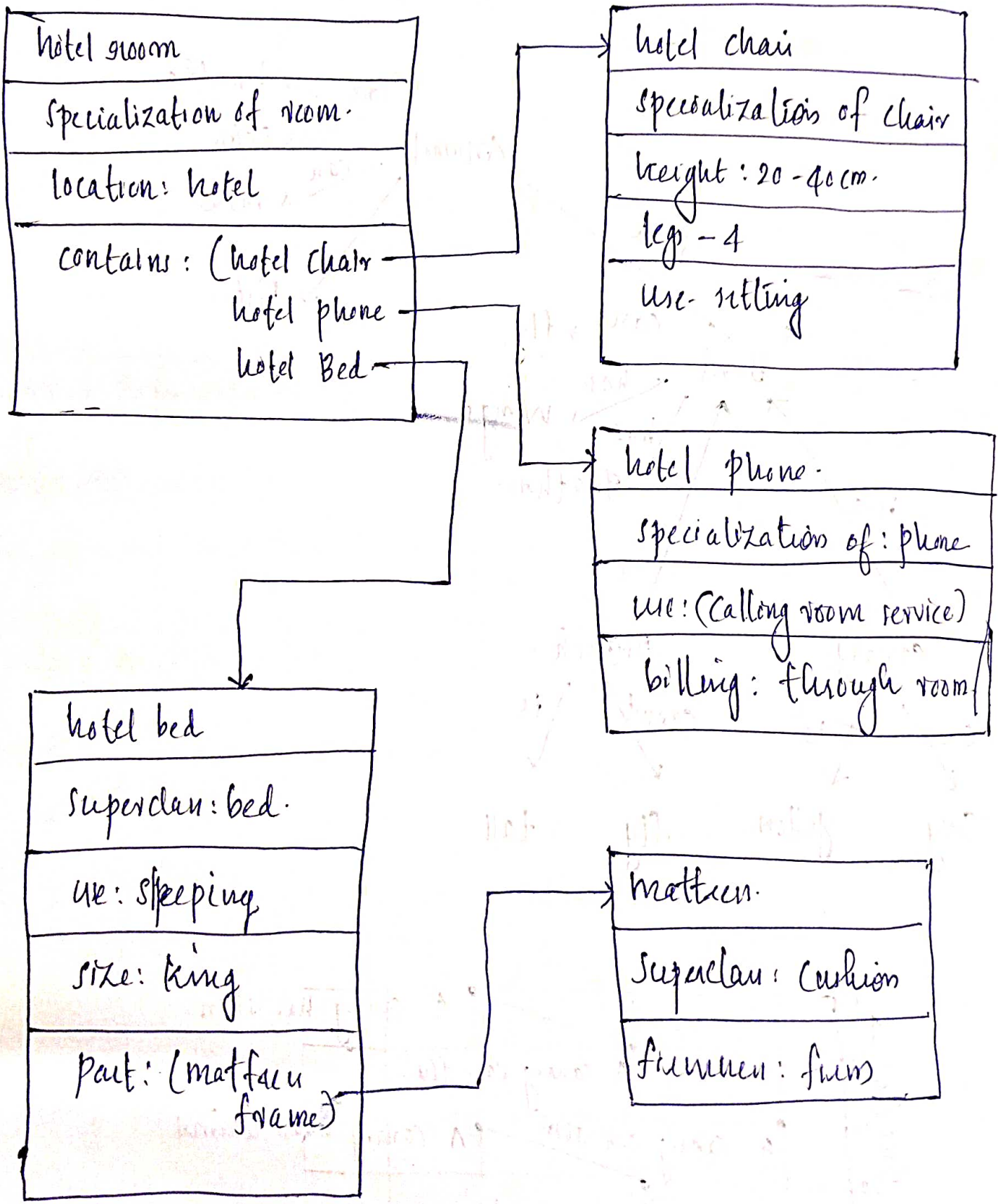


* Real life application of Semantic Network - Search Engine.





Frames Representation.



Scripts

A script is a structured representation describing a stereotyped sequence of events in a particular context.

- The components of scripts are:

Entry conditions or descriptors of the world that must be true for the script to be called.

Eg: An open restaurant and a hungry customer. the ~~restaurant owner has more money~~ has some money.

Result or fact that are true once the script has terminated.

- Things that support the content of the script.

Eg: tables, ~~using~~ waiters, menus.

- Roles are the actions that the individual participants perform.

- Scenes - Sequences of scene.

(Refer text for diagram)

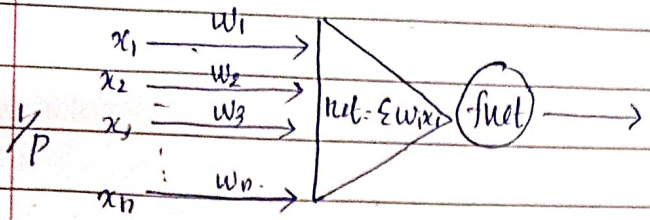
* Issues in Knowledge Representation (Essay)

* ID3 Algorithm problem.

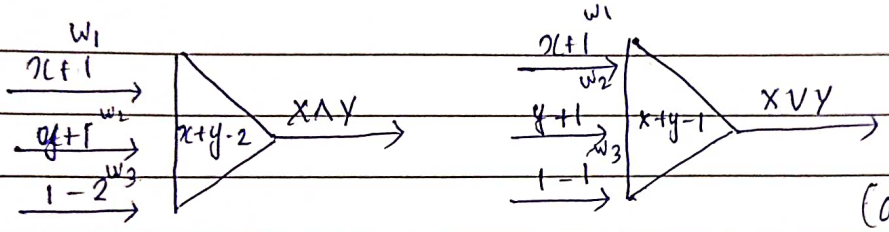
DOMS | Page No.
Date / /

Artificial Neural Network

NET/JRF



$x_1, x_2 \ldots x_n$ - represent the input of the neuron.
$w_1, w_2 \ldots w_n$ - weight / connection strength.
$\Sigma x_i w_i$ - activation level of neuron
function f - determine the final o/p of neuron.



(a) (b)

x	y	$\Sigma x_i w_i$	o/p. (AND GATE)
1	1	0	1
0	1	-1	0
1	0	-1	0
0	0	-2	0

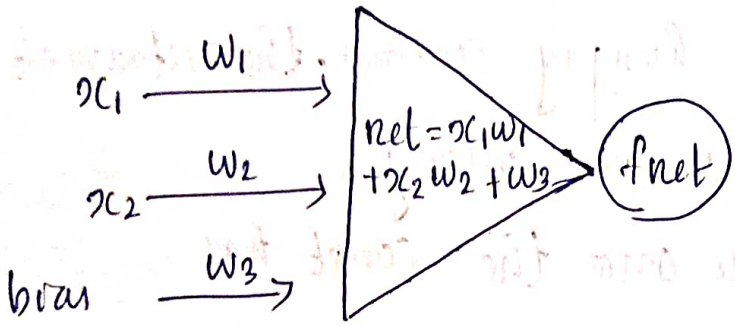
negative value, o/p = 0
+ve / zero, o/p = 1

> Training
> Testing

Perceptron

- Single layer network
- It comes under the supervised learning algorithm.

- if $\Sigma x_i w_i \geq t \rightarrow o/p = 1$
- if $\Sigma x_i w_i < t \rightarrow o/p = -1$



x_1	x_2	o/p
1.0	1.0	1
9.4	6.4	-1
2.5	2.1	1

$f(net)^1 = f(0.75 \times 1 + 0.5 \times 1 + 0.6 \times 1) = f(0.65) = 1$

$f((net)^2) = f(0.75 * 9.5 + 0.5 \times 6.4 + 0.6 \times 1) = f(9.65) = 1$ (wrong)

Learning Rule - How can we adjust the weight

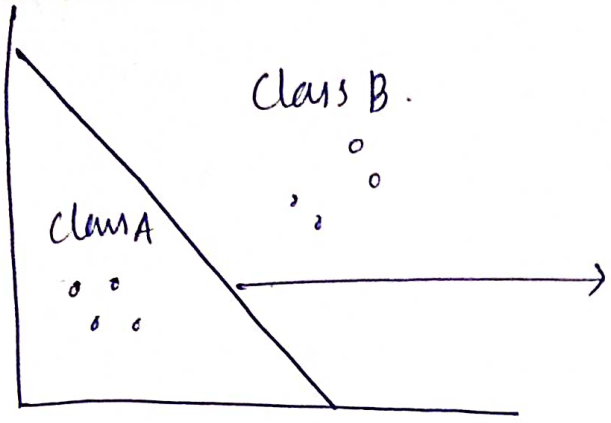
Eg:- $W_3 = w^2 + 0.2 \cdot (-1 \times -1) x^2$

in the above algorithm.

W_3 = weight in 3rd iteration

W_2 = weight " 2nd "

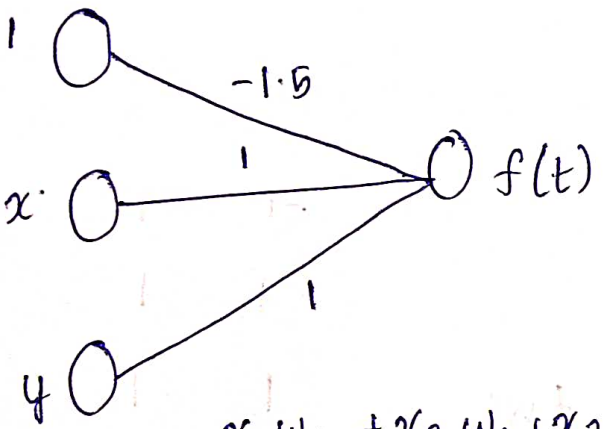
$f(net^3) = f(-3.01 \times 2.5 - 2.06 \times 2.1 - 1 \times 0.1) = f(-12.84) = -1$ (wrong)



This line seperation is by linear equation.

$(w_1 \times x_1) + (w_2 \times x_2) + (1 \times w_3) = 0$.

Q.



and $f(t) \begin{cases} 1, t > 0 \\ 0, t \leq 0 \end{cases}$

$x_1 w_1 + x_2 w_2 + x_3 w_3$

$(1 \times -1.5) + (x \times 1) + (y \times 1)$

$-1.5 + x + y$.

0	0
0	1
1	0
1	1.

0,0 = -1.5 , 0

0,1 = -0.5 0

1,0 = -0.5 0

1,1 = 0.5 1

* Typical Activation Function -- IRF

Q. Perceptron with sign activation function,

$w_1, w_2, w_3 = 0.1, -0.3, 0.1$ and a bias $\theta = 0$.

$x = [0.2, 0.6, 0.5]$

Q. Sigmoid fn $= \frac{1}{1+e^{-x}}$

Q. $C_1 : [-1,-1], [-1,1], [1,-1]$

$C_2 : [1,1]$

decision boundary b/w this two.

A. $x_1 - x_2 - 0.5 = 0$

B. $-x_1 - x_2 - 0.5 = 0$

C. $0.5(x_1 + x_2) - 1.5 = 0$

✓ D. $x_1 + x_2 - 0.5 = 0$

-1	-1	0.5
-1	1	2
1	-1	1
1	1	0

D) $w_1 = 1, w_2 = 1, w_3 = -0.5$

$1 \times 1 + 1 \times 1 = 0.5 = 1.5$ (C2)

Q. $C_1 : [1, 1.5]$ $[1, -1.5]$

$C_2 : [-2\ 2.5]$ $[-2\ -2.5]$.

Single perceptron.

$x_1 + x_2 + 1.5 = 0$

$x_1 + x_2 - 1.5 = 0$

$x_1 + 1.5 = 0$ ✓

$x_1 - 1.5 = 0$

~~$x_1 + x_2 + 1.5 = 0$~~

~~$w_1 = 1,\ w_2 = 1,\ w_3 = 1.5$~~

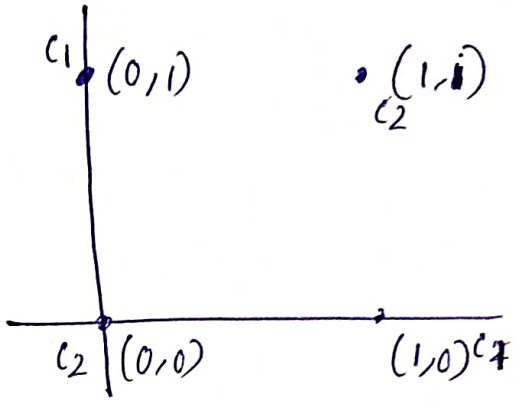
~~1 + 1 + 1.5 = 1.~~

$x_1 + 1.5 = 0$.

$1 + 1.5 = 2.5$.

$-2 + 1.5 = -0.5$.

XOR



not linearly seperable.
can't solve by perceptron.

→ Drawback of perceptron

- Can't solve non-linear seperable problem.
  Eg: XOR gate.

* → Backpropagation.

- can solve XOR problem.