② B

Infer: $A \wedge B$

8. TRANSPOSITION:

Given: $A \rightarrow B$

Infer: $\neg B \rightarrow \neg A$

24/2/21 PRINCIPLE OF RESOLUTION

① Given: 1. $A \vee \neg B$

2. B

Infer: A

② Given: ① $A \vee B$

② $\neg A$

Infer: B

"Principle of resolution can only be applied if knowledge is in "CLAUSE FORM""

→ To apply principle of resolution we have to convert ~~large~~ logic expression into clauses.

CONVERSION OF PROP. LOGIC EXPRESSION INTO CLAUSE:

1. Remove all implications and equivalente operators using suitable laws.
2. ~~Correct~~ Convert the expr. into CNF.
3. Partition the expr. from conjunction operator into clauses.

Ex $(A \vee B) \rightarrow C$

Law of implic.: $\neg(A \vee B) \vee C$

$(\neg A \wedge \neg B) \vee C$

CNF: $(\neg A \vee C) \wedge (\neg B \vee C)$

Clauses: ① $\neg A \vee C$

② $\neg B \vee C$

Sabyasachi
DATE / /

Sol: Convert following statements into Prop logic exp. and by principle of resolution prove that "car is wet".

W = Car is wet

1. If it rains and car is in open it will wet.
2. It is raining
3. Car is open.

$R \wedge O \rightarrow W$

$\neg(R \wedge O) \vee W$

$(\neg R \vee \neg O) \vee W$

R: It is raining
O: car is in open.

1. $R \wedge O \rightarrow W$ Rule
2. R } Facts
3. O

To apply principle of resolution all the expression must be converted into clause form.

1. $R \wedge O \rightarrow W$ {Law of implication}

$\Rightarrow \neg(R \wedge O) \vee W$

$\Rightarrow (\neg R \vee \neg O) \vee W$ {Demorgan Law}

$\neg R \vee \neg O \vee W$ ← clause form

Now following clauses are there for Principle of Reso.

1. $\neg R \vee \neg O \vee W$
2. R
3. O

Apply resolution b/w clause 1 and 2

4. $\neg O \vee W$

Apply resolution b/w clause 4 and 3

5. W

Sabyasachi
DATE / /

Ex: Convert following statements into prop. logic expression.

1. If it rains and I do not stay home I will be wet.
2. If I wet & not change the clothes I will be in fever.
3. It is raining
4. I don't stay home.
5. I don't change clothes.

Prove by resolution than I am in fever.

R: It rains
H: I stay home
W: I am wet
C: I change clothes
F: I am in fever

① $R \wedge \neg H \rightarrow W$
② $W \wedge \neg C \rightarrow F$
③ R
④ $\neg H$
⑤ $\neg C$

To apply principle of resolution all the expression must be converted into clause form.

① $R \wedge \neg H \rightarrow W$
$\neg(R \wedge \neg H) \vee W$
$\neg R \vee H \vee W$ → clause form

② $W \wedge \neg C \rightarrow F$
$\neg(W \wedge \neg C) \vee F$
$\neg W \vee C \vee F$ → clause form

Apply resolution
b/w ① & ③ ⑥ $H \vee W$
⑥ & ④ ⑦ W
② & ⑤ ⑧ $\neg W \vee F$
⑦ & 8 F

Sabyasachi
DATE / /

26/2/21

Que.1 All IIIT Bhopal students are intelligent
2. Yatharth is IIIT student? Is Yatharth intelligent

I: IIIT Bhopal students are intelligent
Y:

PREDICATE LOGIC (1st order logic)

- Knowledge is represented in the form of logical exp.
- Operand of exp.: Predicates.
- Predicate: these are like functions of high level language, but they are always true/false.

→ Here functions can also be used but not as operand rather these are used within predicate.

- Function: These are like predicate but they return non boolean values.

- Operation of logical expression are same as of proposition logic expression.

- Here we can used Quantifiers:
  * Universal Quantifiers ∀ : for all
  * Existential Quantifiers ∃ : there exist

Predicate:

F(x, y) : "x is father of y".
M(x,y) : "x is more than y".
F(x) : "x is a father".

Sabyasachi
DATE / /

Function:

age (x) : age of 'x'

$f(x)$: father of 'x'.

$S(x)$: Salary of x.

sum (x,y) : summation of x and y

If age of student is more than 20 years.

$M(x,y)$: x is more the y.

$M($ age $\overset{\text{student}}{(x)}, 20)$ T

Following predicates are used:

$S(x)$: x is IIIT bhopal student

$I(x)$: x is intelligent

① $(\forall x)\ S(x) \rightarrow I(x)$ ① $\neg S(x) \vee I(x)$

② $S(Yatharth)$ ② $S(yatharth)$

Prove: $I(Yatharth)$

Clauses:

① $\neg S(x) \vee I(x)$

② $S(Yatharth)$

Apply resolution b/w clauses ① and ②

a) By binding x to yatharth

b) By unifying $S(x)$ and $S(yatharth)$

1. $\neg S(yatharth) \vee I(yatharth)$
2. $S(Yatharth)$

obtain: $I(Yatharth)$

Sabyasachi
DATE / /

Que: Represent following sentences in predicate logic expression.

1. If x is on top of y, y supports x.
2. If x is above y and they touching each other, x is on top of y.
3. A cup is above a book.
4. A cup is touching a book.

(b) Show that book supports cup using resolution.

$T(x,y)$ = x is on top of y

$S(y,x)$ = y supports x

$A(x,y)$ = x is above y

$CT(x,y)$: x touches y

To prove { ~~A (cup, book)~~ cup above a book. / CT (cup, book) = cup touching book

① $T(x,y) \rightarrow S(y,x)$

② $A(x,y) \wedge CT(x,y) \rightarrow T(x,y)$

① ¬ T(x ③) A (cup, book)

¬(A ④, y CT (cup, y book) (x, y)

② $\neg A(x,y) \vee \neg CT(x,y) \vee T(x,y)$

Clauses:

① $\neg A(x,y) \vee S(y,x) \vee \neg CT(x,y)$

② $\neg A(x,y) \vee \neg CT(x,y) \vee T(x,y)$

③ $A(cup, book)$

④ $CT(cup, book)$

⑤ $\neg A(x,y) \vee \neg CT(x,y) \vee S(y,x)$

binding x with cup and y with book.

apply clause b/w ⑤ & ③ & ④

$S(book, cup)$

⟹ book supports cup.

Ex: By resolution answer " what course would steve like"?

1. Steve only likes easy courses.
2. ~~easy~~ EC courses are hard.
3. All the courses in CSE department are easy.
4. CS324 is a CSE department course.

Following Predicate are used:

$L(x,y)$: ~~E~~(x likes y) x like easy courses

$E(x)$: ∀x (is easy ~~cours~~ courses are hard.

$ECE(x)$: (x x) is : ECE department ~~course~~ of CSE

$CSE(x)$: x is CSE ~~depa~~ department course.

① $\forall x (E(x) \to L(steve, x))$

② $(\forall x)\ ECE(x) \to \neg E(x)$

③ $(\forall x)\ CSE(x) \to E(x)$

④ $CSE(CS324)$

clauses

① $\neg E(x) \vee L(steve, x)$ $C(CS324)$

② $\neg ECE(x) \vee \neg E(x)$

③ $\neg CSE(x) \vee E(x)$

④ $CSE(CS324)$

Applying resolution b/w clauses ③ ② ④ by Binding $x = CS324$.

⑤ $E(CS324)$

apply resolution b/w clause ① & ⑤ by Binding $x = CS324$.

⑥ $L(steve, CS324)$

steve like CS324 course.

Complex Statements

$(\forall x)\ A(x) \rightarrow (\exists y)\ B(x,y) \vee C(y,z)$

PRENEX NORMAL FORM

A predicate logic expression is said to be in PNF if it is in following form.

$(Q_1 x)\ (Q_2 x)\ (Q_3 x)\ \cdots\ (Q_n x)\ M$

Here $Q_1, Q_2, \cdots Q_n$ are the quantifiers and 'M' is the logic expre. without any quantifier.

" LAWS FOR CONVERSION OF LOGICAL EXPRESS INTO PNF

1. $(Qx)\ (Fx) \vee G(y) = Q(x)\ [F(x) \vee G(y)]$

OR

$F(x) \vee (Qy)\ G(y) = Q(y)\ [F(x) \vee G(y)]$

variable associated with quantifers does not part of 2nd predicate

2. $(Qx)\ F(x) \wedge G(y) = Q(x)\ [F(x) \wedge G(y)]$

$F(x) \wedge Q(y)\ G(y) = Q(y)\ [F(x) \wedge G(y)]$

3. $\neg(\exists x)\ A(x) = (\forall x)\ \neg A(x)$

$\neg(\forall x)\ A(x) = (\exists x)\ (\neg A(x))$

4. $(\forall x)\ (A(x)) \wedge (\forall x)\ B(x) = (\forall x)\ [A(x) \wedge B(x)]$

5. $(\exists x)\ A(x) \vee (\exists x)\ B(x) = (\exists x)\ [A(x) \vee B(x)]$

6. $(\forall x)\ A(x) \vee (\forall x)\ B(x) = (\forall x)\ (\forall y)\ [A(x) \vee B(y)]$

Renamed the variable x of B in y

Sabyasachi
DATE / /

7. $(\exists x)\, A(x) \wedge (\exists x)\, B(x) = (\exists x)\,(\exists y)\,[A(x) \wedge B(y)]$
Renaming of x of B by y.

8. $\underbrace{(Q_1(x))\, A(x) \vee (Q_2(x))}_{\text{Both are diff. Quantifier}}\, B(x) = (Q_1 x)\,(Q_2 y)\,[A(x) \vee B(y)]$

9. $(Q_1 x)\, A(x) \wedge (Q_2 x)\, B(x) = Q_1(x)\, Q_2(y)\,[A(x) \wedge B(y)]$

10/3/21

Ex. 1 Convert the expression into PNF

$(\forall x)\, P(x) \longrightarrow (\exists x)\, Q(x)$

$\neg\{(\forall x)\, P(x)\} \vee (\exists x)\, Q(x)$

$(\exists x)\, \neg P(x) \vee (\exists x)\, Q(x)$

$(\exists x)\,[\neg P(x) \vee Q(x)]$

2. $(\forall x)\,(\forall y)\,((\exists z)\,(P(x,z) \wedge P(y,z)) \rightarrow (\exists u)\, Q(x,y,u)$
convert to PNF.

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

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

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

$\underbrace{(\forall x)\,(\forall y)\,(\forall z)\,(\exists u)}_{\text{all quantifiers}}\,[\underbrace{\neg(P(x,y) \wedge P(y,z)) \vee Q(x,y,u)}_{\text{logical expression.}}]$

PNF

Quantifers can only be removed if the expression is in following "form".

$(\exists x_1)\,(\exists x_2) \dots (\exists x_n)\,(\forall x_{n+1})\,(\forall x_{n+2}) \dots (\forall x_{n+m})\, P$

P: expr. does not have any quantifier.

Skolen Form:

[All existential Quantifers] [All universal quantifiers] [Expr. do not have any quantifiers]

Ex: $(\forall x)\ (\exists y)\ [A(x,y) \vee B(y,z)]$

$\cancel{(\exists y)}\ (\forall x)\ [A(x, f(x)) \vee B(f(x), z)]$

can be removed.

Ex-2 $(\forall x)\ (\forall z)\ (\exists y)\ [A(x,y) \vee B(y,z)]$

y depend on both x, y

$(\exists y)\ \cancel{(\forall x)}\ (\forall z)\ [A(x, f(x,z)) \vee B(f(x,z), z)]$

Now quantifier can be removed.

Ex-3 $(\forall x)\ \underset{f(x)}{(\exists y)}\ \underset{g(x)}{(\exists z)}\ [A(x,y) \vee B(y,z)]$

$\cancel{(\exists y)\ (\exists z)\ (\forall x)}\ [A(x, f(x)) \vee B(f(x), g(x))]$

PRINCIPLE OF RESOLUTION:

It can only be applied if expression are in "clause form".

How to convert Predicate logic Expression into clause form.

1. Convert the given expersion in PNF.
2. Apply skolen function so that all existential quatifer are in prefix followed by universal quantifiers & then expr without any quantifier.
3. Now remove all the quantifiers.
4. Convert remaining expres. into CNF.

Sabyasachi
DATE / /

5. Partition this expression into clauses forom conjunction operator.

12/3/21

Que: consider the following sentences:

1. John like all kind of food.
2. Apples are food
3. chickens is food
4. Anything anyone eats and isn't killed by is food.
5. Bill eats peanuts and is still alive.
6. Sue eats everything Bill eats.

a) Translate all statements in FOPL expression.
b) Obtain various clauses.
c) By resolution prove "John like peanuts" & also give answer What food does sue eats.

1.) food(x) : x is a food.
2.) JL(x) : Johan like x
3. Eat(x,y) : x eats y
4. K(x) : x is killed
5. A(x) : x is alive

Let us take $A(x) \iff \neg K(x)$

① $(\forall x)$ food $(x) \rightarrow JL(x)$
② Food(Apple)
③ food(chicken)
④ $(\forall x)(\forall y)$ Eat $(x, y) \wedge \neg K(x) \rightarrow$ food (y)
⑤ Eat(Bill, peanut) $\wedge$ A(Bill)
⑥ $\forall x$ Eat(Bill, x) $\rightarrow$ Eat(sue, x)
⑦

Sabyasachi
DATE / /

Clauses:

1 $\neg (\forall x)$ food $(x) \vee JL(x)$

$(\exists x)$ $\neg$food $(x) \vee JL(x)$

4. $(\exists x)(\exists y)\ \neg(Eat(x,y) \wedge \neg K(x)) \vee food(y)$

$(\exists x)(\exists y)\ \neg Eat(x,y) \vee K(x) \vee food(y)$

Clause:

① $\neg$food$(x) \vee JL(x)$

② food (Apple)

③ food (Chicken)

④ $\neg Eat(x,y) \vee K(x) \vee$ food (y)

⑤ Eat (Bill, peanut)

⑥ A (Bill)

⑦ $\neg$Eat (Bill, x) $\vee$ Eat (Sue, x)

→ Applying resolution b/w 4 & 5.
by doning unification b/w {Eat (x,y), Eat(Bill, Peanut)}

⑧ K(Bill) $\vee$ food (Peanuts)

Applying resolution b/w 8 & 6 by unifying
{food (x), food (peanut)}

~~A (Bill)~~

9. food (Peanut)

Applying resolution b/w 9 & 1.
JL (Peanuts) ⟷ John likes peanuts.

Applying resolution b/w 5 & 7 by unifying
{Eat (Bill, Peanut), Eat (Bill, x)}
x = peanut.

Eat (Sue, Peanut)
Sue eats peanuts

Single ton : W only contains one term

Eg. $W = \{ P(x,a), P(x,a) \}$ $\{ P(x,w) \}$

DATE / /

UNIFICATION ALGORITHM:

Input = $W = \{ P(x, g(a)) \}$ $P(b, y) \}$

1. Set $K=0$, $W_K = W$ and $\sigma_K = \phi$

2. If W_K is a single ton, stop, σ_K is a most gener unifier (mgu) for W, otherwise find disagreement set (D_K) of W_K.

3. If there exist elements V_K and t_K in D_K such that V_K is a variable that does not occur in t_K goto step 4. Otherwise stop, W is not unifiable.

4. Let $\sigma_{K+1} = \sigma_K \cup \{ t_K / v_K \}$ and

$W_{K+1} = W_K (t_K / V_K)$

Put t_K in place of V_K in W_K.

5. Set $K = K+1$ and goto step 2.

Eg. Apply unification Algorithm on W.

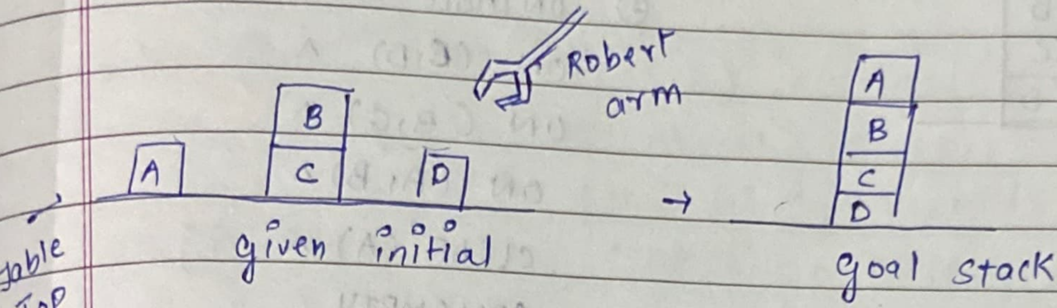
$W = \{ P(a, x), f(g(y)) \}$ $P(z, f(z), f(u)) \}$

step 1: $K=0$

$W_0 = \{$

17-03-2021

BLOCK World Problem / Goal Stack Problem-



To describe / represent the state, we need to use some predicate :

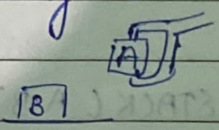
Predicate :

ON (A,B) : Block A is on Block B

ONTABLE (A) : Block A is on the table (surface)

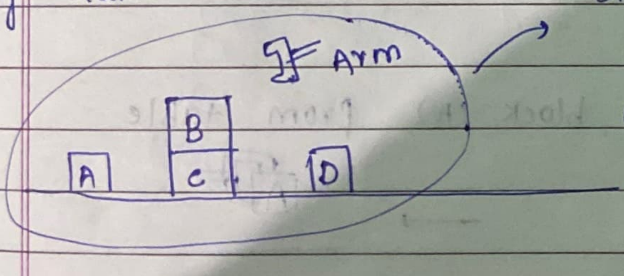
CLEAR (A) : There is nothing on top of Block A.

HOLDING (A) : The robot arm is holding Block A.



ARMEMPTY : The robot arm is holding nothing.

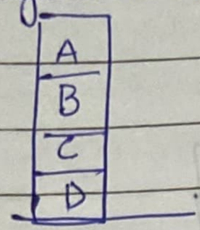
E.g. state



state description / Representation

ONTABLE (A) ∧
ONTABLE (C) ∧
ON (B,C) ∧
ONTABLE (D) ∧
CLEAR (A) ∧
CLEAR (B) ∧
CLEAR (D) ∧
ARM EMPTY

Eg Goal state



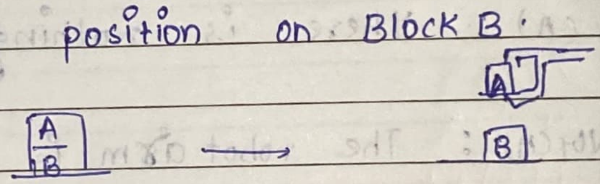
State discription

(1) ONTABLE (D) ∧
ON (C, D) ∧
ON (B, C) ∧
ON (A, B) ∧
CLEAR (A) ∧
ARMEMPTY

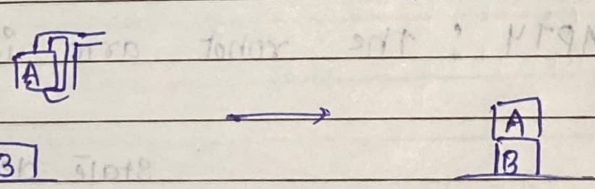
* To change the state we need to perform actions

ACTIONS

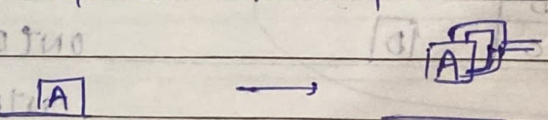
* UNSTACK (A, B) : Pickup Block A from its current position on Block B.



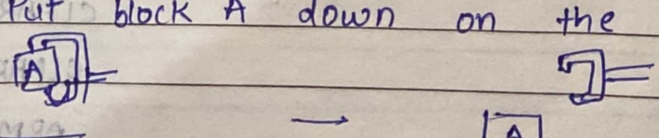
* STACK (A, B) : Place Block A on block B.



* PICKUP (A) : Pick up block (A) from table



* PICKDOWN (A) Put block A down on the table.



* PRECONDITION, DELETION AND ADDITION OF PREDICATES FOR ACTIONS:

* STACK (A,B): P: CLEAR(B) ∧ HOLDING(A)
D: HOLDING(A) ∧ CLEAR(B)
A: ON (A,B) ∧ ARMEMPTY

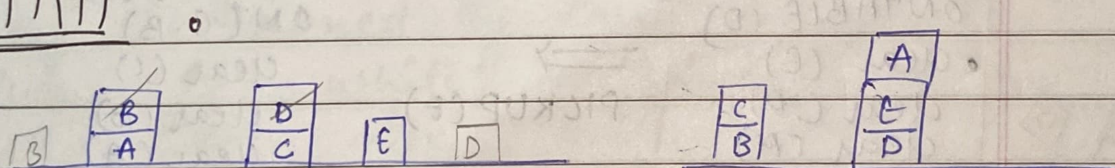
* UN STACK(A,B) P: CLAR (A) ∧ ON (A,B) ∧ ARMEMPTY
D: ARMEMPTY ∧ ON(A,B)
A: HOLDING(A) ∧ CLEAR(B)

19/03/2021

* PICKUP (A): P: ARMEMPTY ∧ ONTABLE (A) ∧ CLEAR(A)
D: ARMEMPTY ∧ ONTABLE (A)
A: HOLDING(A)

* PUTDOWN (A): P: HOLDING (A)
D: ~~ARMEMPTY~~ HOLDING (A)
A: ARMEMPTY ∧ ON TABLE (A)

STRIP:



Initial Stack   Goal Stack

Initial State
ONTABLE (A) ∧
ONTABLE (C) ∧
ONTABLE (E) ∧
ON (B,A) ∧
ON (D,C) ∧
clear (E) ∧
clear (B) ∧ ARMEMPTY
clear (D) ∧

UNSTACK (B,A) ⟹

ONTABLE(A) ∧
ONTABLE(C) ∧
ONTABLE (E) ∧
ON(D,C) ∧
clear (E) ∧
clear (B) ∧
HOLDING (B) ∧
CLEAR (A)

PUTDOWN (B) ⟹

ONTABLE(A) ^
ONTABLE(C) ^
ONTABLE(E) ^
• ON(D,C) ^
~~clear(E)~~ ^
• clear(D) ^
clear(B) ^
clear(A) ^
+ • ARMEMPTY ^
+ ONTABLE(B) ^

UNSTACK(D,C)

ONTABLE(A)
ONTABLE(C)
ONTABLE(E)
ONTABLE(B)
clear(E) ^
clear(D) ^
clear(B) ^
clear(A) ^
+ • HOLDING(D) ^
+ clear(C) ^

PUTDOWN(D)

ONTABLE(A)
• ~~ONTABLE(C)~~
ONTABLE(E)
ONTABLE(B)
+ ONTABLE(D)
~~clear(E)~~
clear(D)
clear(B)
• clear(C)
clear(A)
+ • ARMEMPTY

PICKUP(C)

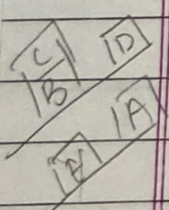
ONTABLE(A)
ONTABLE(E)
~~ONTABLE(B)~~
ON TABLE(D)
clear(E)
clear(D)
• ~~clear(B)~~
clear(C)
clear(A)
• HOLDING(C)

STACK (C,B)

ONTABLE(A)
• ONTABLE(E)
~~ONTABLE(B)~~
ONTABLE(D)
• clear(E)
clear(D)
clear(A)
clear(C)
• ARMEMPTY
ON(C,B)

PICKUP(E)

ONTABLE(A)
ONTABLE(B)
ONTABLE(D)
ON(C,B)
clear(C)
clear(D)
clear(A)
clear(E)
HOLDING(E)



Sabyasachi
DATE / /

KNOWLEDGE REPRESENTATION

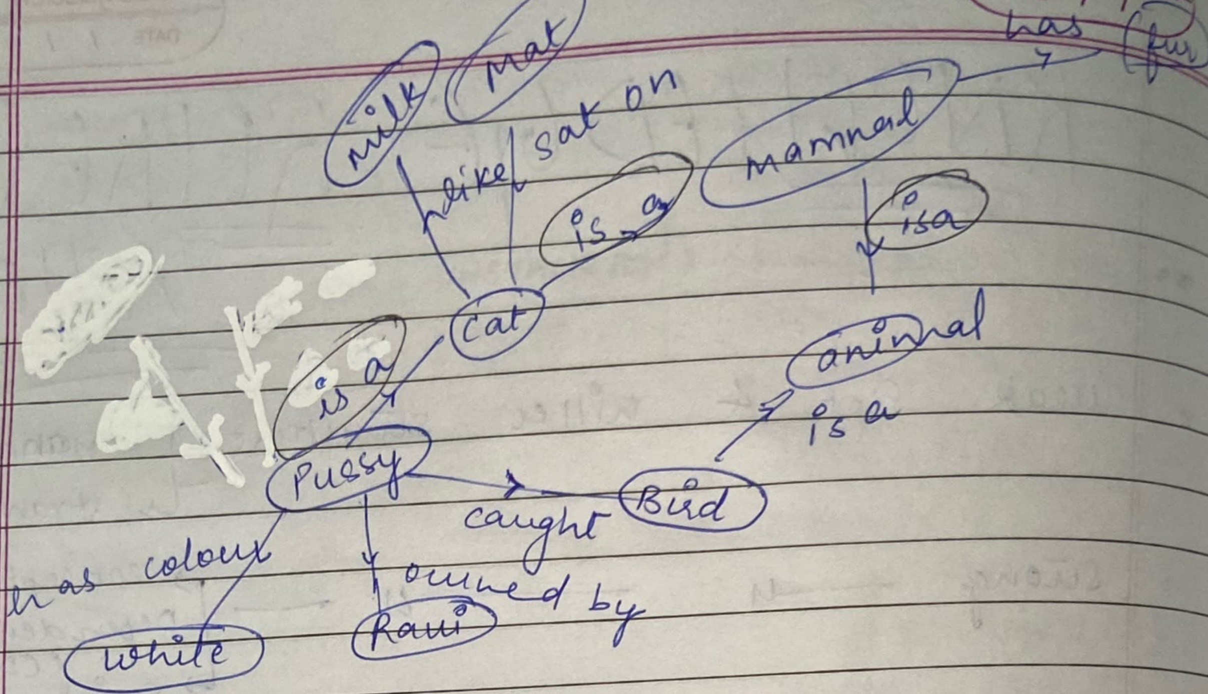
- weak slot & filler structure → Semantic Net., → Frames.
- strong — " ——— " → Conceptional Dependencies (CD), → script

Semantic Network :

→ Simple Semantic Network
→ Partition Semantic Network.

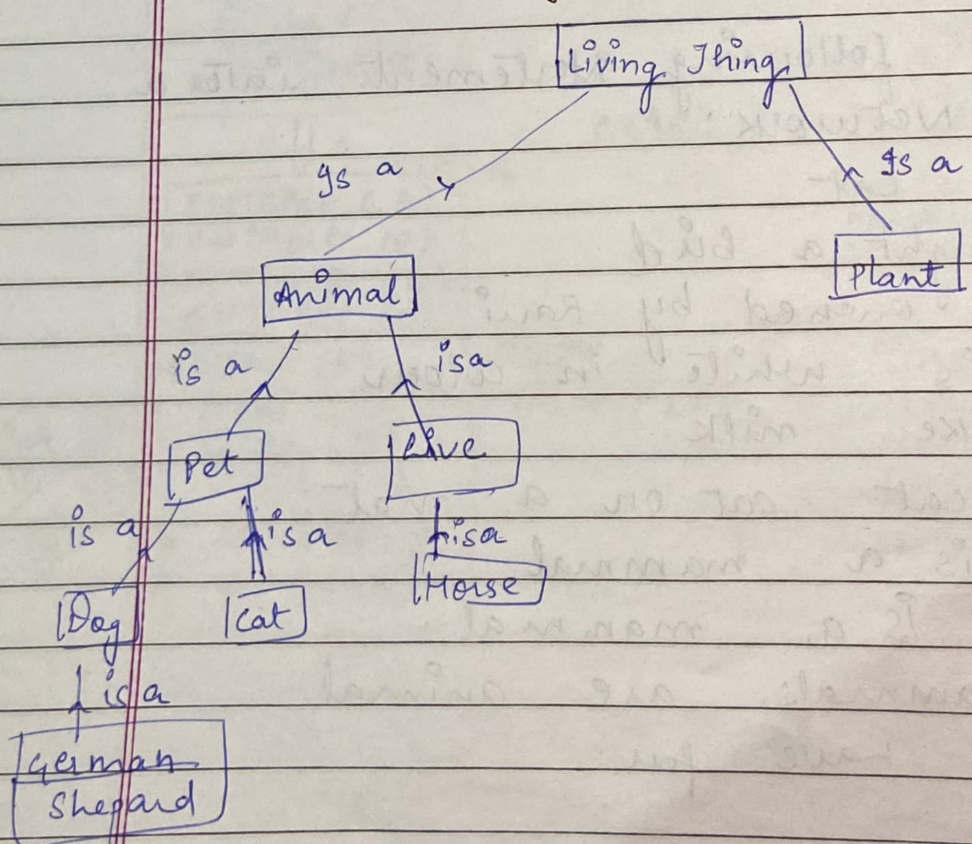
Ex: Represent following statements into a semantic Network.

1. Pussy is a cat
2. Pussy caught a bird
3. Pussy is owned by Ravi
4. Pussy is white in colour
5. Cats like milk
6. The cat sat on a mat
7. A cat is a mammal
8. A bird is a mammal.
9. All mammals are animal
10. Mammals have fur.

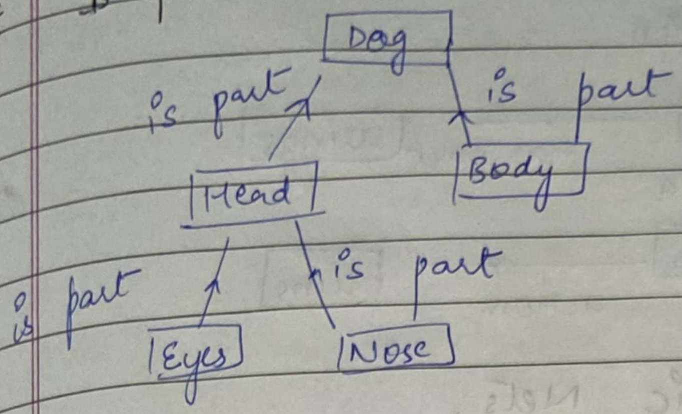


31/3/21 Hierarchy { ~~Is~~ is
Is part

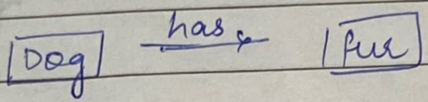
* Is- a Hierarchy



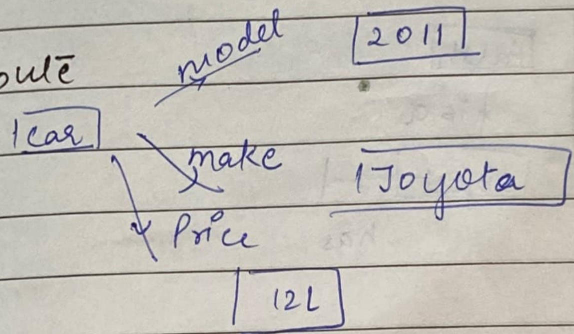
* Is part:



* Has

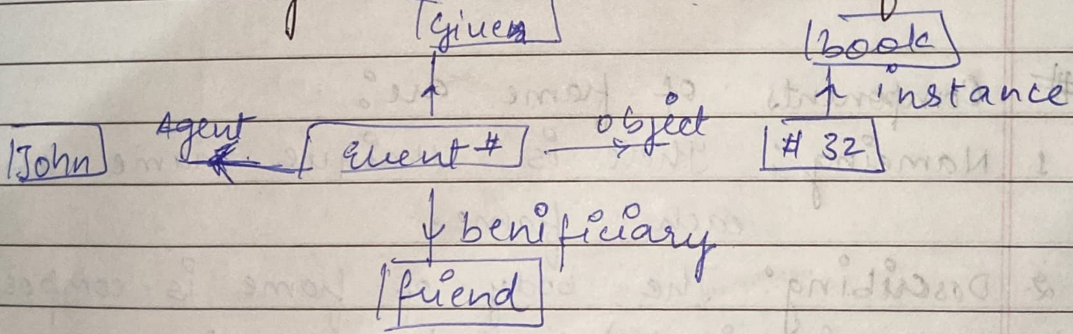


* Attribute

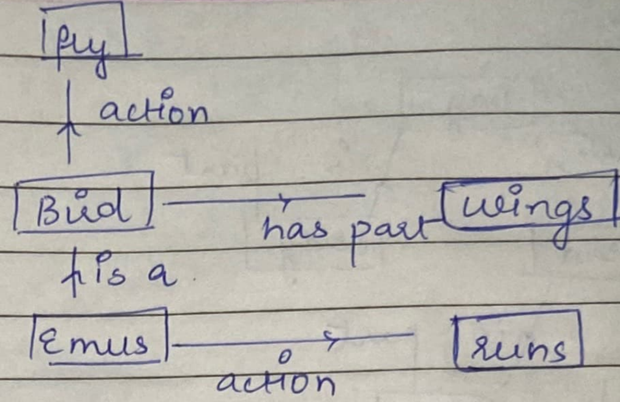


Semantic net can also be used to represent statements with event.

"John gave book to his friend"

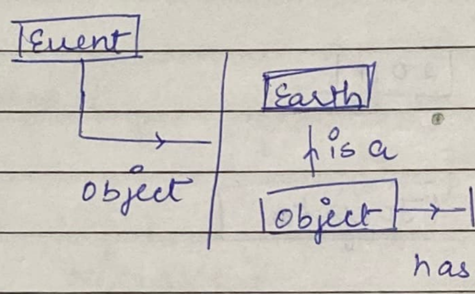


1. Emus are birds
2. Typically birds fly & have wings.
3. Emus runs.



Partion Semantic Nets:

"Ramesh believes that earth is flat."



Frame: It provides a method to combined declarative and procedure within it.

Components of frame are:

1. **Nameing:** There is unique name assigned to each frame.
2. **Describing:** The body of frame is composed of slots (attribute) that have value.
   These slot describes the properties of frame and also used to link other frame.
3. **Organising** - Each frame (except the top level) has one or more parent.
4. **Relating** - slot of frame may be another frame

5. Constraining - with various slot we can specify some constraints by attaching some predicat.

Predicate → If needed
→ If added.

Pictorial Representation

- Frame : $\boxed{\text{Name}}$
Parent1 : $\boxed{\text{Frame Name}}$
Parent 2 :
Slot : $\boxed{\text{slot 1}}$

STRONG SLOT & FILLER STRUCTURES

① Concep

BAY'S THEOREM:

PROBABILITY THEORY:

Sample Space: It is the set of all possible outcomes of the experiment.

Eg 1 Tossing a coin

$S_1 = \{T, H\}$

$S_2 = \{1, 2, 3, 4, 5, 6\}$

Event: It is the outcomes in which we are intended e.g.s.

$E_1 = \{2, 4, 6\}$

$$\text{Probability} = \frac{|E|}{|S|} = \frac{3}{6} = \frac{1}{2}$$

Conditional Probability:

Given two events A and B, if P(B) > 0 then conditional probability that 'A' occurs given that B occurs is defined by.

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

Ex.1 Que. Suppose ~~two~~ a fair coins ~~are~~ is tossed twice what is the probability that both tosses results in head given that atleast one of the result is a head.

A	B
H	T
T	H
T	T
H	H

$$\boxed{= \frac{1}{3}}$$

$S = \{HT, TH, TT, HH\}$ $A = \{HH\}$ $B = \{HT, TH, HH\}$

Sabyasachi DATE / /

Sabyasachi
DATE / /

$P(A\cap B) = 1/4$

$P(B) = 3/4$

Hence

$$\frac{P(A\cap B)}{P(B)} = \frac{1/4}{3/4} = \frac{1}{3}$$

Ex: 2 Suppose two dice are thrown what is the probability that the sum of two die is '8' given that one of the die is giving '3'.

S = { 11, 12, 13, 14, 15, 16 |S| = 36
21, 22, 23, 24, 25, 26
31, 32, 33, 34, 35, 36
41, 42, 43, 44, 45, 46
51, 52, 53, 54, 55, 56
61, 62, 63, 64, 65, 66 }

A = { 62 53 44 35 26 }

B = { 31 32 33 34 35 36 }, 13, 23, ~~3~~, 43, 53, 63 }

$= \frac{2}{11}$

BAY'S THEOREM

$$P(A|B) = \frac{P(A\cap B)}{P(B)} \quad \text{—(1)}$$

$$P(B|A) = \frac{P(B\cap A)}{P(A)} \quad \text{—(2)}$$

$A\cap B = B\cap A$

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

From (1) $\quad P(A\cap B) = P(A|B)\cdot P(B)$

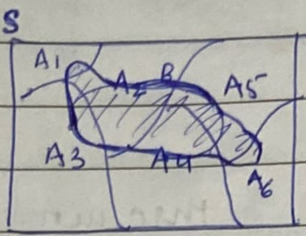
From (2) $\quad P(B\cap A) = P(B|A)\cdot P(A)$

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

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

Extension of Bay's Theorem

Total Probability Theorem



Assume $P(A_i) > 0$ for all "i"

$P(B) = P(A_1 \cap B) + P(A_2 \cap B) + P(A_3 \cap B) \cdots + P(A_n$

$P(B) = P(B/A_1) \cdot P(A_1) + P(B/A_2) \cdot P(A_2) + \cdots + P(B/A_n) \cdot P(A$

$$P(B) = \sum_{j=1}^{n} P(B/A_j) \cdot P(A_j)$$

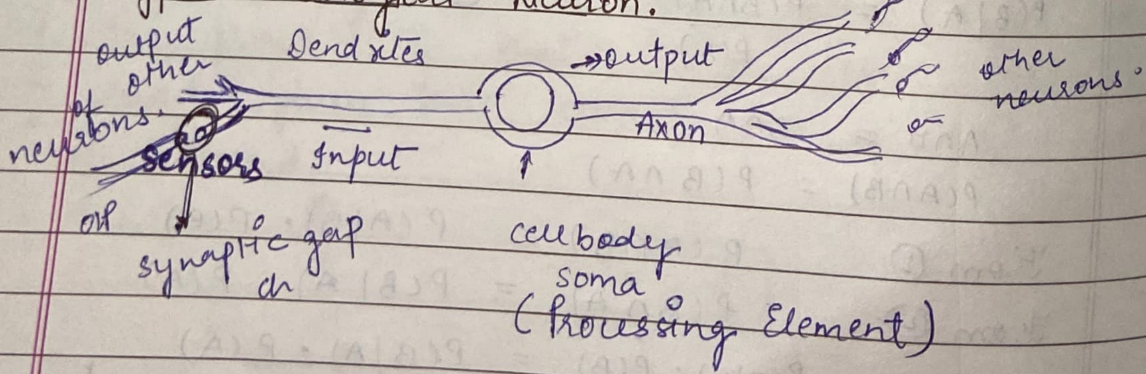
Extendend Bay's Theorem:

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

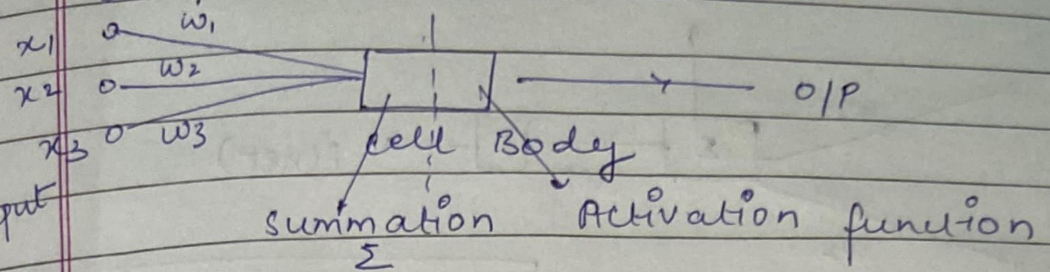
$$\boxed{P(A/B) = \frac{P(B/A) \cdot P(A)}{\sum_{j=1}^{n} P(B/A_j) \cdot P(A_j)}}$$

NEURAL NETWORKS

Typical Biological Neuron:



Artificial Neurons:



$x_1 w_1$ $\quad$ $x_2 w_2$ $\quad$ $x_3 w_3$

net $= x_1 w_1 + x_2 w_2 + x_3 w_3$ (summation)

f(net) will be the output (Activation function)

E.g.

If net < 0 | f(net) $= -1$

If net > 0 | f(net) $= +1$

Hebb Net: Learning | Training algo.

Step 1: Initiallize all weights and bias to zero.

Step 2: for each inputs vector and target (output) perform step 3 to 6.

Step 3: Set $x_i = s_i$

Step 4: Set $y = t$

Step 5: Adjust the weight by Hebb rule.

$w_i(\text{new}) = w_i(\text{old}) + x_i \cdot y$

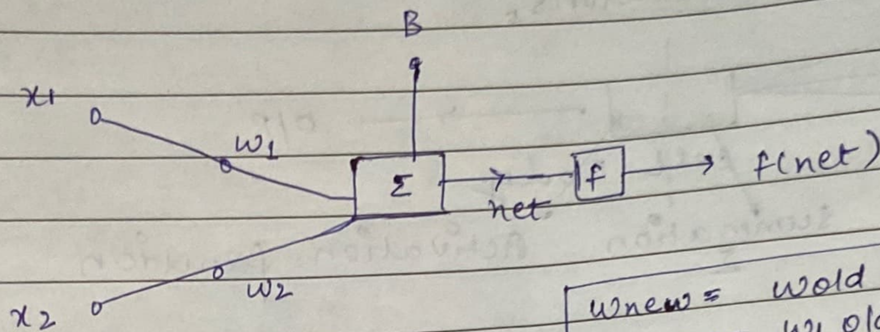
Step 6: $b(\text{new}) = b(\text{old}) + y$

1. We want to train a artifical neuron So it should work like AND gate.
2. For training we need inputs.

$1 \rightarrow 1$
$0 \rightarrow -1$

	x_1	x_2	B	t
s_1	1	1	1	1
s_2	1	-1	1	-1
s_3	-1	1	1	-1
s_4	-1	-1	1	-1

Sabyasachi
DATE / /



$w_{new} = w_{old} + x_i \cdot y$

$w_1 new = w_1 old + x_1 \cdot y_1$

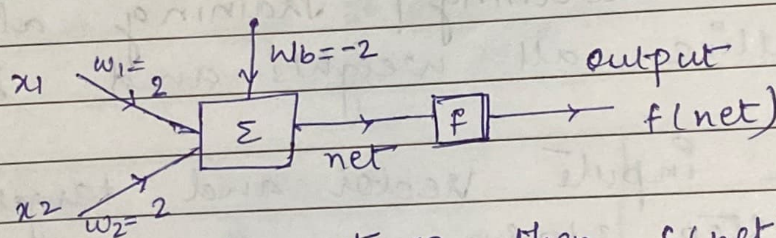
$= 0 + 1 \cdot 1 = 1$

$w_b(new) = w_b old + y$

S1

Input

x_1	x_2	B	Target(y)	w_1	w_2	w_b
-1	-1	1	1	1	1	1
1	-1	1	-1	0	2	0
-1	1	1	-1	1	1	-1
-1	-1	1	-1	2	2	-2



If net < 0 then f(net) = -1

net > 0 f(net) = 1

If $x_1 = -1$ and $x_2 = -1$

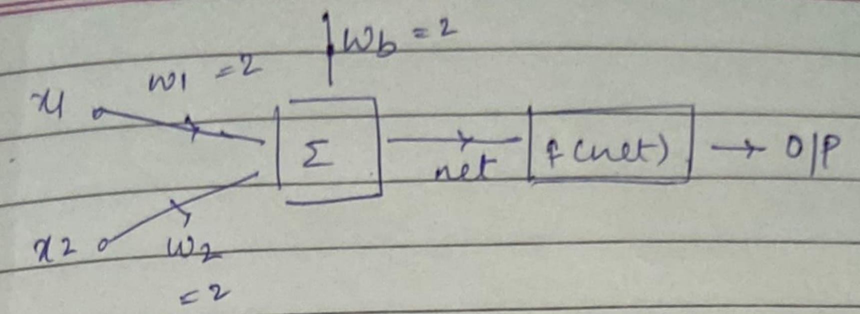
so, net = -6 so output = -1.

OR GATE

x_1	x_2	b	y	w_1	w_2	w_b
1	1	1	1	1	1	1
-1	1	1	1	0	2	2
1	-1	1	1	1	1	3
-1	-1	1	-1	2	2	2

Fuzzy sets

~~fuzzy~~



XOR

	x_1	x_2	B	y
S1	-1	-1	1	-1
S2	1	-1	1	1
S3	-1	1	1	1
S4	1	1	1	-1

Train a neuron

AND x_2

	0	1
0	0	0
1	0	1

OR x_2

	0	1
0	0	1
1	1	1

EXOR

	0	1
0	0	1
1	1	0

O/P's are linearly separable

O/p not linearly separable

* $\boxed{x_1 \oplus x_2 = (x_1 \text{ OR } x_2) \text{ AND } (x_1 \text{ AND } x_2)}$