Assignment No:- 2

KGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCE

Name :- Rohan shridhar Bhawar

Roll No :- 08

class :- B.E / I.T

Batch :- I1

Subject :- IS Lab

Assignment No :- 2

KGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCE

Q. 1] Solve the following with forward chaining or backward chain-ing or resolution (any one) use predicate logic as language of knowledge representation clearly specify the fact and inference rule used

Q. 1] Example 1 :

1) Every child sees some witch no witch has both a black cat and a pointed hat
2) Every witch is good or bad
3) Every child who sees any good witch gets candy
4) Every witch that is bad has a black cat
5) Every witch that is seen by any child has pointed hat
6) Prove :- ∀ Every child gets candy

→ A) facts into fol

1) ∃x∀y (child (x), witch (y) → sees (x, y))
   ∼∃y (witch (y) → has (y, black cat) ∧ has (y, Pointed hat))
2) ∃y (witch (y) → good (y) ∨ bad (y))
3) ∃x ((sees (x, y) → (witch (y) → good (y)) → get (x, candy)
4) ∃y ((witch (y) → bad (y)) → has (y - black hat))
5) ∃y (sees (x, y) → has (y, Pointed hat)

B) FOL into CNF

1) ∃x ∀y (child (x), witch (y) → sees (x, y))
→ ∼∃y, (witch (y) → has (y, black hat)
→ ∼∃y (witch (y) → has (y, Pointed hat)
2) ∀y (witch (y) → good (y))
   ∀y (witch (y) → bad (y))
3) ∃x [(sees (x, y) → witch (y) → good (y)) → gets (x, candy)
4) ∃y [bad (y) → has (y ; black hats)]
5) ∃y (seen (x, y) → has (y, Pointed hat)]

KGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCE

→ ~∀y [seen (x,y) → has (y, black hat)]

(ɩ)

Sees (x, y)

Witch ~~we~~ (y) v sees (x, y)
{good v bad / y}

~seen (x, (good) ∧ sees (x, bad)

has (y, z)
{y / good v bad} {z / black cat v Pointes hat)}

Seen (x, good) v seen (x, bad)

has (good, pointes hats
v gets (x, candy)

Seen (x, good) v has good,
Pointes hat) v gets (x, ~~b~~(candy)

Seen (x, good) v gets
(x, candy)

get (x, candy) gets (x, candy)

2) Example 2 :
1) Every boy or girl is a child
2) Every child gets a doll or a train or a lump of a coal
3) No boy get any doll
4) Every child who is bad gets any lump of coal
5) No child get a train
6) Ram gets lump of coal
7) Prove Ram is bad

⇒ 1) ∀x (boy (x) or girl (x) → child (x))
2) ∀y (child (y) → get (y, doll) or get (y, train) or get (y, coal))
3) ∀w (boy (w) → ! gets (w, doll))
4) for all z (child (z) and bad (z)) → get (z, coal)) ∀y child(y) → ! get (ram, coal)
5) child (ram) → get (ram, coal)
To Prove (child (ram) → bad (ram)

CNF clauses
1) ! boy (x) or child (x)
! girl (x) or child (x)
2) ! child (y) or get (y, doll) or get (y, train) or ! get (w, doll)
3) ! boy (w) or ! get (y, coal)
4) ! child (z) or ! bad (z) or get (z, coal)
5) ! child (ram) → get (ram, coal)
6) bad (ram)
Resolution
4) ! child (z) or ! bad (z) or get (z, coal)
6) bad (ram)
7) ! child (ram) or get (ram, coal)
Substitutions z by ram
1) (a) ! boy (x) or child (x) boy (ram)
8) child ram | substitutions x by ram
7) ! child (ram) or get (ram, coal)
8) child (ram)
9) get (ram, coal)
2) ! child (y) or get (y, doll) or get (y, train) or get (y, coal)
8) child (ram)
10) gets (ram, doll) or get (ram, train) or get (ram, coal)
(Substitutions y by ram)

9) gets (ram, coal)
10) get (ram, doll) or gets (ram, train) or gets (ram, coal)
11) gets (ram, doll) or gets (ram, coal)
3) !boy (w) or !gets (w, doll)
5) boy (ram)
12) !get (ram, doll). (substituting w by ram)
11) gets (ram, doll) or gets (ram, train)
12) !gets (ram, coal)
13) gets (ram, doll)
6) (a) get (ram, coal)
13) gets (ram, coal)
Hence ~~ram~~ is bad (ram) is Proved

Q. 2) Differentiate between STRIPS and ADL

	STRP language	ADL
1)	Only allow Positive literal in the states for eg: A valid sentance is STRIP is expressed as ⇒ Intelligent ^ Beautiful	Can support both Positive & negative literal for e.g. Some sentance is expressed as ⇒ stupid ∧ - ugly
2)	STRIPs stand for Standard Research Institute Problem solve	Stands for Action Description Language.
3)	Make use of closed world assumption (i.e) an mentioned literal ~~are false in goal~~ are false	Makes use open world ~~Assum~~ Assumption (i.e) unmentioned literals are unknown

KGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCE

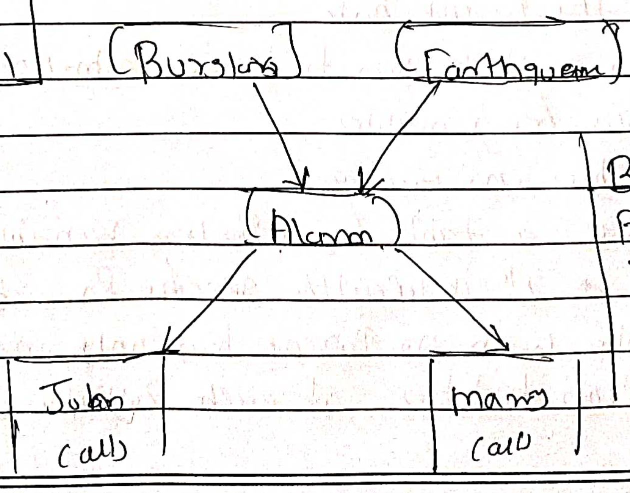
4)	We only can find ground literal in goal For eg :- Intelligent A x Beautiful	We can find Qualified variables in goal for eg :- ∃ x At (P1, X) ∧ At (P2, X) is the goal of having P1 & P2 in the same place in example of P blocks
5)	Goal are conjuction eg (Intelligent ∧ Beautiful	Goal may involve conjuction and disconjuction e (Intelligent ∧ (Beautiful A Rich))
6)	Does not support equality	Equality Predicate (x = y) is build in

Q. 4) You have two neighbours J and M who have promised to call you at work when they here the alarm J always call when he hears the alarm but sometimes confuses telephone ringing with alarm & call then too M likes loud music and sometimes misses the alarm together Given the evidence of who has or has not called we would like to estimate the probability of burglary Draw a Bayesian network for this domain with suitable probability table.

→

P(B)
0.001

P(E)
0.02



B	E	P(A)
F	T	0.95
T	F	0.94
F	T	0.29
F	P	0.01

A	P(T)
T	0.09
F	0.05

A	P(M)
T	0.70
F	0.01

1) The topology of the network indicate that - Burglary and earthquake affect the Probability of the alarm going off
- Wheather John and mary call depends only on alarm
- They do not precive any burglaries directly they do not notice minor earthquake and they do not confer before calling
2) Mary listening to loud music & John confusing phone ringing to sound of alarm can be ready from network only implicity as uncertainity associated calling at work
3) The Probability actual summarizes potentially infinite set of circumstance
- The alarm might fail to go off due to high humidity, Power failure etc
- John and mary might fail to call and report a alarm because they are out to lunch on vacation, temporarily deaf. Passing helicopter etc
4) The condition Probability table in n/w gives probability for values of random variables depending on combination of value for the parent node
5) Each row must be sum to because entries represent exhaustive set of case for variables
6) All variables are Boolean
7) In general, a table for a Boolean variable with k, Parent contain ~~2*~~ 2^k independently specific Probabilities
8) A variable with no Parent has only one row representing Prior Probabilities of each possible value of variable

9) Every entry in full join Probability distribution can be calculated from information in Bayessian network

10) A generic entry in joint distribution is Probability of a conjuction of Particular assignment to each other $P(X_1 = x_1 \wedge \cdots \wedge \cdots X_n = x_n)$ abbervivated as $P(x_1 \cdots x_n)$

11) The values of this 'entry is $P(x_1 \cdots x_n) = \pi_{i=1}$, ~~AA~~ np (1, Parent (xi)) where Parent s(xi) denotes the specific values of the variable Parent (xi)

$- P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$

$= P(j|a)\, P(m|a)\, P(a|\neg b \wedge \neg e)\, P(\neg b)\, e(\neg e)$

$= 0.09 \times 0.07 \times 0.001 \times 0.999 \times 0.998$

$= 0.0006828$

12) Bayesian Network