Assignment no:- 2

Name :- Harikrishna B. Padhi

Roll no :- 40

Class :- B.E-IT

Sem :- VII

Subject :- IS lab

DOP	DOC	Marks	Sign

Q.1 Solve the following with forward chaining or backward chaining or resolution (any one) use predicate logic as language of knowledge representation clearly specify the facts & inference rule used

Q. Example :
1) Every child sees some witch. No witch has both a black cat & 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 a pointed hat.
6) Prove: Every child gets candy

→ A) Facts into fol
1) $\exists x \forall y$ (child (x), witch (y) → sees (x, y))
~ $\exists$ y (witch (y) → has (y, black cat) ^ has (y, pointed hat))
2) $\exists$ y (witch (y) → good (y) v bad (y))
3) E x ((sees (x, y) → (witch (y) → good (y)) → get (x, candy))
4) Ey ((witch (y) → bad (y)) → has (y → Back hat)))
5) Ey (sees (x, y) → has (y, pointed hat)

B) Fol into CNF
1) $\exists x \forall y$ (child (x), witch (y) → Sees (x, y))
→ ~ $\exists$y, (witch (y) → has (y, black hat)
→ ~ $\exists$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)

→ ∃x ((sees(x, good(y)) → gets(x, candy))]

4) ∃y [bad(y) → has(y, black hats)]

5) ∃y [Seen(x,y) → has(y, pointed hat)]

→ ~∀y [Seen(x,y) → has(y, black hat)]

c)

Sees(x,y)

witch(y) ∨ sees(x,y)
{good ∨ bad / y}

~ sees(x, good) ∧ sees(x, bad)

has(y,z)
{y / good ∨ bad}
{z / black cat ∨ pointed hat}

Seen(x, good) ∨ Seen(x, bad)

has(good, pointed hats ∨ get(x, candy))

Seen(x, good) ∨ seen(x, bad)

has(good, pointed hats ∨ get(x, candy))

gets(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 gets any doll
4) Every child who is bad gets any lump of coal.
5) No child gets 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) → gets(y, doll) or gets(y, train) or gets(y, coal))
3) ∀w (boy(w) → ! gets(w, doll))
4) for all z (child(z) and bad(z)) → gets(z, coal))
∀y child(y) → ! gets(y, train)
5) child(ram) → gets(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 gets(y, doll) or gets(y, train) or gets(y, coal)
3) ! boy(w) or ! gets(w, doll)
4) ! child(z) or ! bad(z) or gets(z, coal)
5) ! child(ram) → gets(ram, coal)
6) bad(ram)

Resolution

4) ! child (z) or ! bad (z) or get (z, coal)

6) bad (ram)

7) ! child (ram) or gets (ram, coal)

substituting z by ram

1) (a) ! boy (x) or child (x)

boy (ram)

8) child ram (substituting x by ram)

7) ! child (ram) or gets (ram, coal)

5) child (ram)

9) gets (ram, coal)

2 ! child (y) (or gets (y, doll) or gets (y, train) or

gets (y, coal)

8) child (ram)

10) gets (ram, doll) or gets (ram, train) or gets (ram, coal)

(substituting y by ram)

9) gets (ram, coal)

10) gets (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, doll)

13) gets (ram, coal)

1c) (a) get (ram, coal)

13) gets (ram, coal)

Hence, bad (ram) is proved.

Q.2 Differentiate between STRIPs and ADL

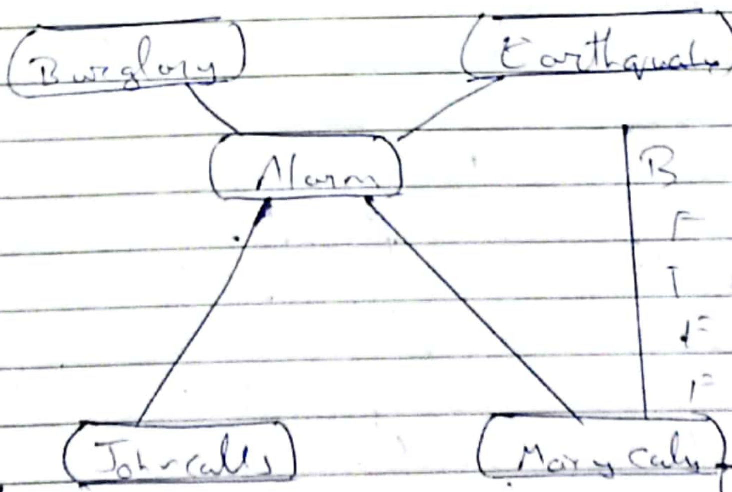
STRIPs language	ADL
1) Only allow positive literals in the states for eg: A valid sentence in STRIPs is expressed as => Intelligent ∧ Beautiful	Can support both positive & negative literals for eg: same sentence is expressed as => stupid ∧ ¬ ugly
2) STRIPs stands for Standard Research Institute Problem solver	Stands for Action Description language
3) Makes use of closed world assumption (i.e) unmentioned literals are false	makes use of open world Assumption (i.e) unmentioned literals are unknown
4) We only can find ground literals in goals for eg:- Intelligent ∧ 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 the example of blocks
5) Goals are conjunctions for eg:- (Intelligent ∧ Beautiful)	5) Goals may involve conjunctions & disjunctions for eg:- (Beautiful ∧ Rich)
6) Effects are conjunctions	Conditional effects are allowed: when P:E means E is an effect only if P is satisfied.

7) Does not support equality	Equality Predicate (x=y) is build in
8) Does not have support for types	Support for types for: The variable P: person

Q.4 You have two neighbors J and M, who have promised to call you at work when they here the alarm. J always calls when he hears the alarm, but sometimes confused telephone ringing with alarm & calls then too. M likes loud music & sometimes misses the alarm music, & sometimes 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	F	0.001

A	P(J)
T	0.09
F	0.05

A	P(M)
T	0.70
F	0.01

① The topology of the network indicates that:
- Burglary & earthquake affect the probability of the alarms 'going off.'
- whether John & Mary calls depends only on alarm.

② Mary listening to loud music & John confusing phone ringing to sound of alarm can be read from network only implicitly as uncertainty associated to calling at work.

③ The probability actualy summarize potentially infinite sets of circumstances.
- The alarm might fall to go off due to high humidity, power failure, dead battery, cut wire, a dead mouse stuck inside the bellet.

④ The condition probability table in n/w gives probability for values of random variables depending on combination of value for the parent nodes.

⑤ Each row must be sum to 1, because entries represent exhaustive set of cases for variable.

⑥ All variables are Boolean.

⑦ In general, a table for a Boolean variable with k parents contains 2^k indepently specific probabilities.

8) A variable with no parent, has only one row representing prior probabilities of each possible value of the variable.

9) Every entry in full joint probability distribution can be calculated from information in Bayesian n/w.

10) A generic entry in joint distribution is probability of a conjuction of particular assignments to each variable $P(x_1 = x_1 \wedge \ldots \wedge x_n = x_n)$ abbreviated as $P(x_1, \ldots, x_n)$

11) The value of this entry is $P(x_1, \ldots x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(x_i))$, where $parents(x_i)$ denotes the specific values of the variables $parents(x_i)$

- $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$
  $= P(j|a) P(m|a) P(a|\neg b \wedge \neg e) P(\neg b) P(\neg e)$
  $= 0.09 \times 0.07 \times 0.001 \times 0.999 \times 0.998$
  $= 0.000628$

12) Bayesian Network