Name : Rubi. R. Shaikh

Class : BE-IT

Roll no : 59

Subject : IS Lab.

DOP	DOA	Remark	Sign.

Assignment - 2

Q. 1)

1.)

→

A] Facts into fol .

1. ∃ X∧Y (child (X) , with (Y) → sees (X,Y))
~∃Y (with (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, (with (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, and Y)
→ ∈X [(sees (X, good (Y) → gets (X, and Y)]
4. ∈Y [seen (X, Y) → has (Y, pointed hat)]
→ ~∀Y [seen (X,Y) → has (Y, black hat)]

C]

sees (x, y)   witch(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 pointed hat}}

seen (x, good) V seen (x, bad)   has (good, pointed hats V get (x, and y)

seen (x, good) V has (good, pointed hat) V gets (x, candy)   seen (x, good) V gets (x, candy)

gets (x, and y)   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) .

Resolut? :
4. ! child (z) or ! bad (z) or get (z, coal)
6. bad (ram)
7. ! child (fram)
1. (a) ! boy (x) or child (x)
boy (ram)
8. child ram / substituting x by ram)
7. ! child (ram) or gets (ram, coal) .

8. child (ram)
10. gets (ram, doll) or gets (ram, train) or
gets (ram, wal)
(substituting Y by ram)
9. gets (ram, wal)
10. gets (ram, doll) or gets (ram, wal)
3. ! boy (w) or ! gets (w, doll)
5. boy (ram)
12. ! get (ram, doll) (substracting w by ram)
11. gets (ram, doll) or gets (ram, train)
12. ! gets (ram, doll)
13. gets (ram , wal)
6. <a> get (ram, wal)
13. gets (ram, wal)
Hence, bad (ram) is proved.

Q. 2. →

STRIPs language	ADL .
1.) Only allow positive literals in the states. For eg: A valid sentence is STRIP's is expressed as	1.) Can support both positive & negative literals For eg :- Same sentence is expressed → stupid ∧ - ugly.
2.) STRIPs stand for Standard Research Institute Problem solve.	2.) Stands for Action Descript? language.

3.) Makes use of closed world assumpt^n literals are false	3.) Makes use of open world Assumpt^n (i.e.) unmentioned literals are unknown.
4.) Goals for conjunct^n	4.) 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 eg of blocks.
5.) Goals are conjunct^n for eg:- (Intelligent ∧ beautiful)	5.) Goals may involves conjunct^n & disjunct^n for eg:- (Intelligent ∧ (Beautiful ∧ Rich)
6.) Effect are conjunct^n	6.) Conditional effects are allowed: when P:E means E is an effect only if P is satisfied.
7.) Does not support equality.	7.) Equality prediate (X = Y) is build in.
8.) Does not have support for types.	8.) Support for types. For eg: The variables P : person.

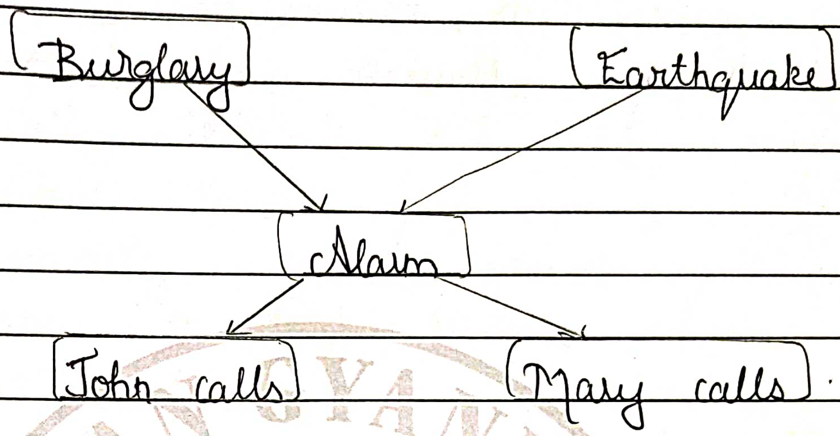
Q. 4.

GCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCE

→

P(B)
0.001

P(E)
0.002



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 n/w indicates that
- Burglary and earthquake affect the probability of the alarm going off.
- Wheather John and Mary call depends only on alarm.

2. Mary listening to loud music & John confusing phone ringing to sound of alarm

can be read from n/w only implicity as uncertainly associated to calling at work.

3. The probability actually summarize potentially infinite sets of circumstances.
- The alarm might fail to go off due to high humidity, power failure, dead battery, cut wires, a dead mouse stuck inside the bell, etc.
- John and mary might fail to call and report & alarm because they are out to lunch, on vacat^n, temporarily deaf, passing helicopter, etc.

4. The cond^n probability tables in n/w gives probability for values of random variables depending on combinat^n of values for the parent nodes.

5. Each row must be sum to 1, becoz 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 independently specific probabilities.

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

9. Every entry in full joint probability distribut^n can be calculated from informat^n in Bayesian n/w.

10. A generic entry in joint distribut^n is probability of a conjuct^n of particular assignments to each variable $P(X_1 = x_1 \wedge \dots \wedge \dots X_n = x_n)$ abbreviated as $P(x_1, \dots, x_n)$.

11. The value of this entry is $p(x_1, \dots, x_n) = \prod_{i-1}, n P(1, Parents(x_i))$, where parents $S(x_i)$ denotes the specific values of the variables parents (x_i).

$- p(j \wedge m \wedge a \wedge \sim b \wedge \sim e)$

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

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

$= 0.000628$

12. Bayesian n/w :