K.G.C.E.
Karjat - Raigad

Page No. :
Date :

KGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCE

Assignment No. 2

-> Name :- Tejas Balaso Jadhav.

-> Roll No. :- 24

-> Class :- B.E./IT

-> Sem :- VII

-> Sub :- IS LAB

D.O.P.	D.O.C.	Marks	Sign

Scanned with CamScanner

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. 1) Example 1:

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] ∃ 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] E x ((see s (x,y) → (witch (y) → good (y))
→ get (x, candy)

4] E y ((witch (y) → bad (y)) → has (y → black hat))

5] Ey (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)]

→ ~∀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, candy)

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

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

2] Example :

1] Every body or girl is a child.
2] Every child gets a doll or a train or a lump of a coal.
3] No body gets any doll.
4] Every child who is bad gets any lump of coal.
5] No child gets a train.
6] Prove Ram is bad.
7] Ram gets lump of coal.

→ 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) & bad (z)) gets (z, coal))
∀y child (y) → ! gets (y - train)
5] Child (ram) → gets (ram, coal)
To prove (child (ram) → bad (room))

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] bod (rom)

Resolution:

4) ! child (z) or ! bad (z) or get (z, coal)
5] bad (ram)
6] ! child (ram) or gets (ram, coal)
sub
7] ! child (ram) or gets (ram, coal)
Substituting z by ram.
8] (a) ! boy (x) or child (x)
boy (ram)
9] Child ram (substituting x by ram)
7] ! child (ram) or gets (ram, coal)
8] 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) or gets (ram, train)
11] gets (ram, doll) or gets (ram, train)
12] ! gets (ram, doll)
13] gets (ram, coal)

Hence, bad (ram) is proved.

Q 2 Differentiate between STRIPS & ADL

STRIPs language	ADL
→ Only allow positive literals in the states. For eg: A valid sentence is STRIPs is expressed as ⇒ Intelligent ∧ Beautiful	→ Can support both positive & negative literals. For eg: Same sentence is expressed as ⇒ stupid ∧ ugly
→ STRIPs stand for standard Research Institute Problem solver	→ Stand for Action Description language
→ Makes use of closed world assumption (i.e.) un mentioned literals are False.	→ makes use of open world Assumption (i.e.) unmentioned literals are unknown
→ 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.
→ Goals are conjunctions for eg :- (Intelligent ∧ Beautiful)	→ Goals may involve conjuctions &

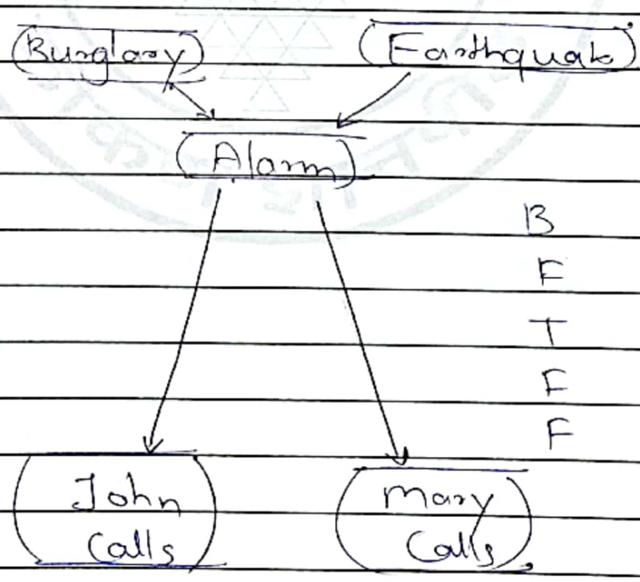
	disjunctions for eg:- (Intelligent ∧ (Beautiful ∧ Rich))
:> Effects are conjunctions	:> Conditional effects are allowed: when P : E means E is an effect only if P is satisfied.
:> Does not support equality	:> Equality predicate (x = y) is build in
:> Does not have support for types	:> Support for types for eg: The variable p : person

Q. 4 You have two neighbor's I & m, who have promised to call you at work when they here the alarm. J always calls when he here the alarm but sometimes confused, telephone ringing with alarms & calls then too. m liks loud music & sometimes misses then 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)
0.002



B	E	P(A)
F	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

A	P(T)
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.
- Wheather John & Mary call depends only on alarm.
- They do not perceive any burglaries directly they do not notice minor earthquake & they do not confer before calling.
- 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.
- The probability actually summarize potentially infinite sets of circumstences.
- 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 & mary might fail to call & report & alarm because they are out to lunch, on vacation, temporaily def. passing helicopter etc
- The condition probability tables in n/w gives probability for values of random variables depending nodes.

-> Each row must be sum s/to 1 becuse entries represent exhaustive set of cases for variable.

-> All variables are Boolean.

-> In general, a table for a Boolean variable with k parents contains 2^k independently specific probabilities.

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

-> Every entry in full joint probability distribution can be calculated from information in Bayssian network.

-> A generic entry in joint distribution is probability of a conjuction of particular assignment to each variable $P(X_1 = x_1 \wedge \ldots \wedge X_n = x_n)$ abbreviated as $P(x_1, \ldots, x_n)$

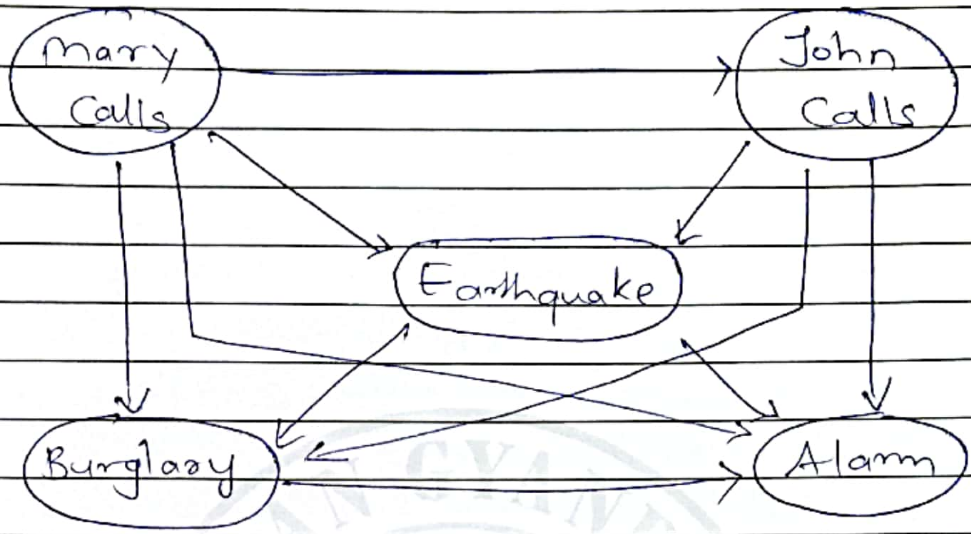
-> The value of this entry is $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(X_i))$, where parents(Xi) denotes the specific values of the variables parents (Xi)

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

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

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

$= 0.000628$


Mary Calls
John Calls
Earthquake
Burglary
Alarm