K.G.C.E.
Karjat - Raigad

Assignment -2

Page No. :
Date :

Name : Shubham P. Bawalekar

Roll no : 04

Class : BE IT

Sem : VII

Sub : ISLab

DOP	DOC	marks	Sign

Q1 Solve the following with forward or backward chaining or resolution (any one). Use predicate logic as the language of knowledge representation. Clearly specify the facts and inference rule used.

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 a pointed hat.
6. Prove: Every child gets candy.

→ A) Facts into FOL.

1) $\exists x \wedge y\,(child(x), witch(y) \rightarrow sees(x, y))$
 $\neg \exists y\,(witch(y) \rightarrow has(y, black\ cat) \wedge has(y, pointed\ hat))$

~~2) ∃y (witch (y) → has (y, black cat~~

2) $\exists y\,(witch(y) \rightarrow good(y) \vee bad(y))$

3) $\in x\,((sees(x, y) \rightarrow (witch(y) \rightarrow good(y)) \rightarrow get(x, candy))$

4) $\in y\,((witch(y) \rightarrow bad(y)) \rightarrow has(y \rightarrow black\ hat))$

5) $\in y\,(sees(x, y) \rightarrow has(y, pointed\ hat)$

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) switch (y) → good (y)) → gets (x, candy)

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

4) ∃y [bad (y) → has (y, bad 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}

¬ seen (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 hat) ∨ gets (x, candy)

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

seen (x, good) ∨ gets (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 train or lump of coal
3) No boy gets any doll.
4) Every child who is bad gets any lump of coal.
5) ~~Ram~~ No child gets train
6) Ram gets lump of coal
7) Prove: Ram is bad.

→ 1) $\forall x$ (boy(x) or girl(x) → child(x))
2) $\forall y$ (child(y) → gets(y, doll) or gets(y, train) or gets(y, coal))
3) $\forall w$ (boy(w) → !gets(w, doll))
4) For all z (child(z) and bad(z)) → gets(z, coal))
$\forall y$ 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)
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)
13) gets (ram, coal)
6) (a) get (ram, coal)
13) gets (ram, coal)
Hence, bad(ram) is proved.

Q2 Difference between STRIPS and ADL.

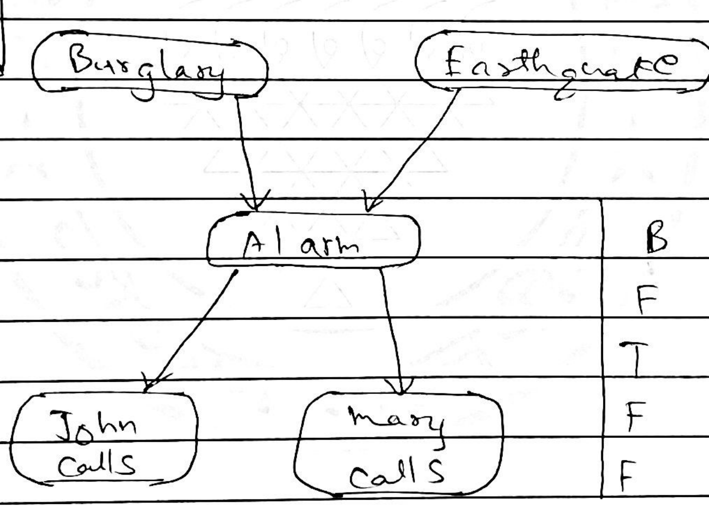
STRIPS Language	ADL
1) Only allow positive literals in states for eg. A valid sentence is STRIPS is expressed as ⇒ Intelligent ∧ Beautiful	1) Can support both positive & negative literals For eg : same sentence is expressed as ⇒ stupid ∧ ¬ugly.
2) STRIPS stand for Standard Research Institute Problem Solver.	2) Stands for Action Description Language.
3) Makes use of closed world assumption (i.e.) unmentioned literals are false.	3) Makes use of open world Assumption (i.e.) unmentioned literals are unknown.
4) Does not support equality.	4) Equality predicate (x = y) is build in.
5) Does not have support for types	5) Support for types for eg : The variable P : person.
6) Goals are conjunctions	6) Goals may involve conjunctions & disjunctions

Q4 You have two neighbors J & M, who have promised to call you at work when they hear the alarm. J always calls when he hears alarm, but sometimes confused telephone ringing with alarm and calls then too. M likes loud music & sometimes misses the alarm altogether. 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	0.10

1. The topology of the network indicates that
- Burglary and earthquake affect the probability of the alarms going off.
- whether John and Mary call depends on alarm.

- They do not perceive any burglaries directly they do not notice minor earthquakes and they do not confer before calling.

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 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 & report & alarm because they are out to lunch on vacation, temporarily deaf, passing helicopter, etc.

4) The condition probability takes in n/w gives probability for values of random variables.

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 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 variable.

9) Every entry in full joint distribution is probability of conjunction 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($x_1, \ldots, x_n$) = π i - 1, n P(1, parents(x_i)), where parents(x_i) denotes the specific values of variables parents (x_i)

- P(j ∧ m ∧ a ∧ ¬b ∧ ¬e)

= P(j|a) P(m|a) P(a|¬b ∧ ¬e) P(¬b) e(¬e)

= 0.09 × 0.07 × 0.001 × 0.999 × 0.998

= 0.000628

12) Bayesian Network