Solver: Li Haihui

Disclaimer: Remarks are written in *italics* for your better understanding of the solution. You are not required to have them for answering the questions in this paper.

1)

a)

i) False

ii) True

iii) False. *Max(h1, h2) is better than min(h1, h2) as Max provides closer estimate to the cost of reaching the goal, given both are admissible.*

iv) False. *Depth-first search may possibly, sometimes, by good luck, expand fewer nodes than A* search with an admissible heuristic. For example, it is logically possible that sometimes, by good luck, depth-first search may march directly to the goal with no back-tracking.*

v) False

b)

i) S→G

ii) S→A→B→D→G

iii) S→A→C→G

c) Constraint propagation eliminates values from domains of variables that can never be part of a consistent solution. This leads to a reduction of the search space, making the problem easier to solve by some algorithms.

2)

a)

i) Nodes in the priority queue before each expansion of least-cost node are listed below. G(n) values for each node at each state are shown in parentheses.
① PH (0) [PH to be expanded]
② AL (62), HA (107) [AL to be expanded]
③ HA (107), SC (136) [HA to be expanded]
④ SC (136), PI (311) [SC to be expanded]
⑤ PI (311), ER (428) [PI to be expanded]
⑥ ER (428) [ER to be expanded → Goal reached, thus search terminated]
Thus, the shortest path to ER based on Uniform Cost Search is: PH → AL → SC → ER.

ii) Nodes in the priority queue before each expansion of least-cost node are listed below. H(n) values for each node at each state are shown in parentheses.
① PH (330) [PH to be expanded]
② HA (225), AL (290) [HA to be expanded]
③ PI (100), SC (250), AL (290) [PI to be expanded]
④ ER (0), SC (250), AL (290) [ER to be expanded → Goal reached, thus search terminated]
Thus, the shortest path to ER based on Greedy Search is: PH → HA → PI → ER.

iii) Nodes in the priority queue before each expansion of least-cost node are listed below. F(n) values for each node at each state are shown in the parentheses.

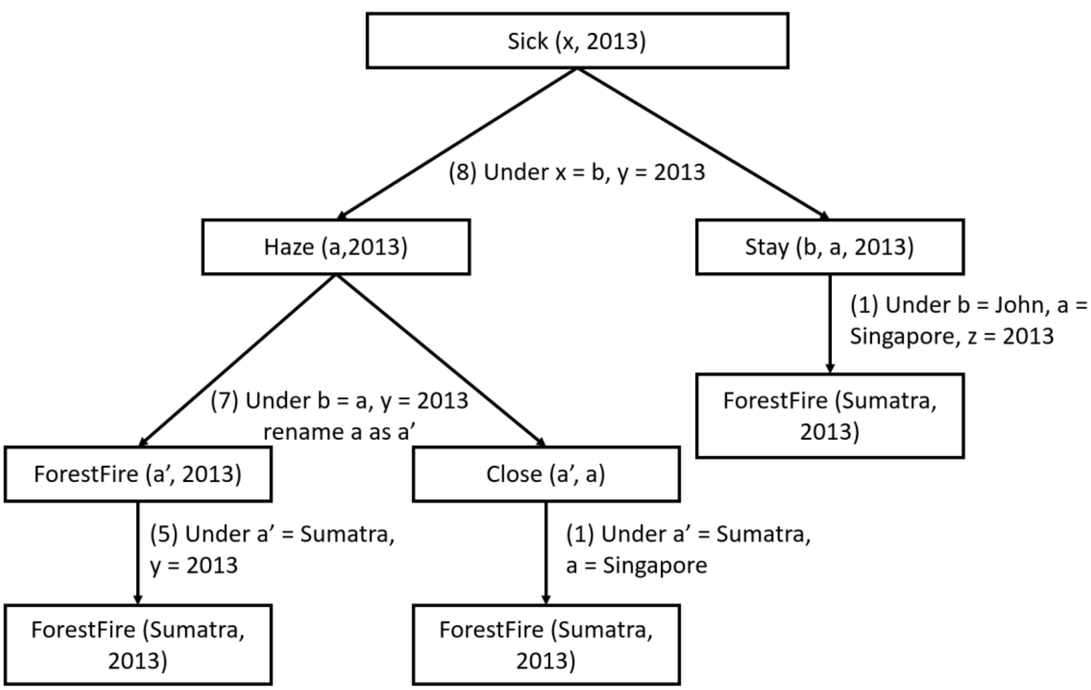
① PH (0 + 330 = 330) [PH to be expanded]
② HA (107 + 225 = 332), AL (62 + 290 = 352) [HA to be expanded]
③ AL (62 + 290 = 352), PI (311 + 100 = 411), SC (228 + 250 = 478) [AL to be expanded]
④ SC (136 + 250 = 386, cost updated), PI (311 + 100 = 411) [SC to be expanded]
⑤ PI (311 + 100 = 411), ER (428 + 0 = 428) [PI to be expanded]
⑥ ER (428 + 0 = 428) [ER to be expanded → Goal reached, thus search terminated]
Thus, the shortest path to ER based on A* search is: PH → AL → SC → ER.

b)
- i) A: 4, B: 4, C: 2, D: 4, E: 10, F:2, G: 9
- ii) Move to Node B

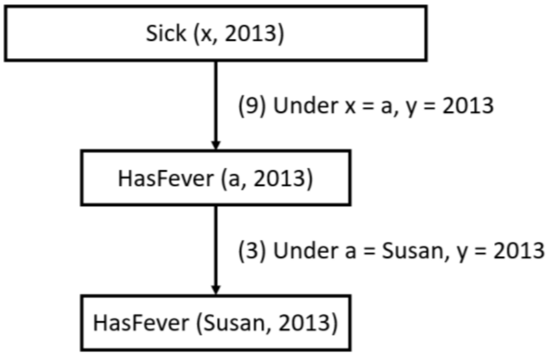
3)

a)
- i) False
- ii) True
- iii) False. *Generalized Modus Ponens (GMP) is complete for KBs containing only HORN clauses. A Horn clause is a sentence of the form:* $\forall x, P1(x) \wedge P2(x) \wedge \ldots \wedge Pn(x) \Rightarrow Q(x)$*. The statement is true for resolution inference rule.*
- iv) False. *The execution of Prolog is done via depth-first backward chaining*.

b) The query Sick(x,2013) will be solved using backward chaining GMP.
1-7: Fails
8:



9:



From the above inference, we know that John and Susan are sick.
You may have other ways of presenting the answer; it's perfectly fine so long as you show it clearly.

c)

i) Constants: Adrian, Brian, Calvin, David

Predicates:

Insomnia(x)	"x has insomnia"
Headache(x)	"x has a headache"
BlkedNose(x)	"x has blocked nose"
Appetite(x)	"x has appetite"
Sick(x)	"x is sick"
Sad(x)	"x is sad"

Sentences:

i. $Insomnia(Adrian) \wedge Headache(Adrian)$
ii. $BlkedNose(Brian) \wedge \neg Appetite(Brian)$
iii. $Headache(Calvin) \wedge BlkedNose(Calvin)$
iv. $\neg Appetite(David) \wedge Headache(David)$
v. $\forall x\ Insomnia\ (x) \wedge \neg Appetite(x) \Rightarrow Sick(x)$
vi. $\forall x\ BlkedNose(x) \Rightarrow Insomnia(x)$
vii. $\forall x\ Sick(x) \Rightarrow Sad(x)$

ii) ① $Insomnia(Adrian)$
② $Headache(Adrian)$
③ $BlkedNose(Brian)$
④ $\neg Appetite(Brian)$
⑤ $Headache(Calvin)$
⑥ $BlkedNose(Calvin)$
⑦ $\neg Appetite(David)$
⑧ $Headache(David)$
⑨ $\neg Insomnia\ (x) \vee Appetite(x) \vee Sick(x)$
⑩ $\neg BlkNose(x) \vee Insomnia(x)$
⑪ $\neg Sick(x) \vee Sad(x)$

iii) A. <u>Is Adrian sad?</u>
To prove KB ⊨ Sad (Adrian) by contradiction, we assume Adrian is not sad,

i.e. ⑫ $\neg Sad\ (Adrian)$.
Resolve ①+⑨ under x = Adrian: ⑬ $Appetite(Adrian) \vee Sick(Adrian)$
Resolve ⑪+⑬ under x = Adrian: ⑭ $Appetite(Adrian) \vee Sad(Adrian)$
Resolve ⑫+⑭ under x = Adrian: ⑮ $Appetite(Adrian)$
Resolve ⑨+⑪: ⑯ $Appetite(x) \vee Sick(x) \vee Sad(x)$
There are no new clauses that can be added, in which case KB does not entail Sad (Adrian).

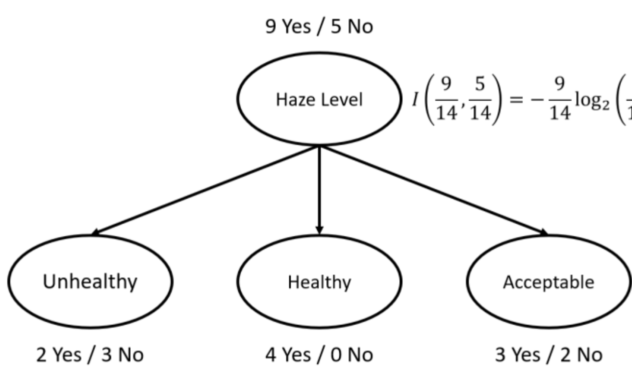
Conversely, to prove KB ⊨ $\neg Sad(Adrian)$ by contradiction, we assume Adrian is sad,
i.e. ⑫ $Sad\ (Adrian)$. Obviously, we cannot reach a contradiction given KB and assume ⑫.
Thus, KB also does not entail $\neg Sad\ (Adrian)$.
Therefore, we cannot deduce whether Adrian is sad from the knowledge base in (ii). Same conclusion can be reached Calvin and David based on similar proof. *Intuitively, this is because we don't know whether Adrian, Calvin and David have appetite.*

B. Is Brian sad?
Assume Brian is not sad, i.e. ⑫$\neg Sad(Brian)$.
Resolve ⑪+⑫ under x = Brian: ⑬$\neg Sick(Brian)$
Resolve ⑨+⑬ under x = Brian: ⑭$\neg Insomnia(Brian) \vee Appetite(Brian)$
Resolve ⑩+⑭ under x = Brian: ⑮$\neg BlkedNose(Brian) \vee Appetite(Brian)$
Resolve ③+⑮ under x = Brian: ⑯ $Appetite(Brian)$
Resolve ④+⑯ under x = Brian: {} (empty clause reached → There's contradiction!)
Thus, $\neg Sad(Brian)$ is false. Brian is sad.

Part B is likely to have more weight than Part A. Given it is only worth 6 marks, you may not need to show as detailed as my solution for Part A during exam.

4)

a) We first calculate the information gain for split on each attribute.

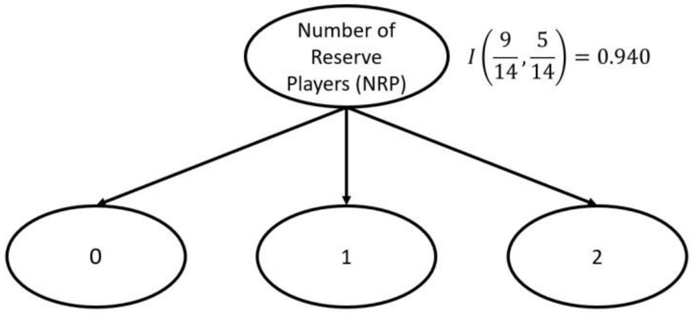


$$I\left(\frac{9}{14},\frac{5}{14}\right) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

$$I\left(\frac{2}{5},\frac{3}{5}\right) = 0.971 \qquad I\left(\frac{4}{4},\frac{0}{4}\right) = 0 \qquad I\left(\frac{3}{5},\frac{2}{5}\right) = 0.971$$

Gain (Haze Level)= $I\left(\frac{9}{14},\frac{5}{14}\right) - \frac{5}{14} * I\left(\frac{2}{5},\frac{3}{5}\right) - \frac{4}{14} * I\left(\frac{4}{4},\frac{0}{4}\right) - \frac{5}{14} * I\left(\frac{3}{5},\frac{2}{5}\right) = 0.246$

9 Yes / 5 No



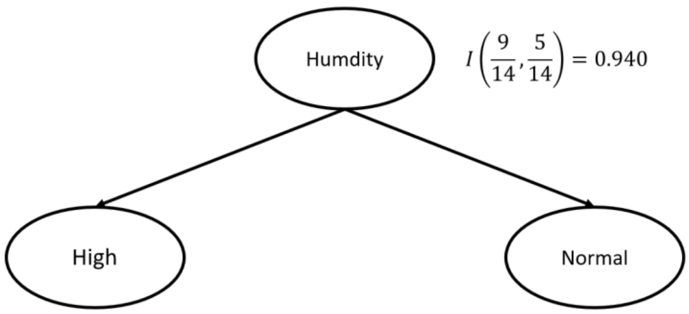
$I\left(\frac{9}{14},\frac{5}{14}\right) = 0.940$

3 Yes / 1 No | 2 Yes / 2 No | 4 Yes / 2 No

$I\left(\frac{3}{4},\frac{1}{4}\right) = 0.811$ | $I\left(\frac{2}{4},\frac{2}{4}\right) = 1$ | $I\left(\frac{4}{6},\frac{2}{6}\right) = 0.918$

Gain (NRP)= $I\left(\frac{9}{14},\frac{5}{14}\right) - \frac{4}{14} * I\left(\frac{3}{4},\frac{1}{4}\right) - \frac{4}{14} * I\left(\frac{2}{4},\frac{2}{4}\right) - \frac{6}{14} * I\left(\frac{4}{6},\frac{2}{6}\right) = 0.0291$

9 Yes / 5 No



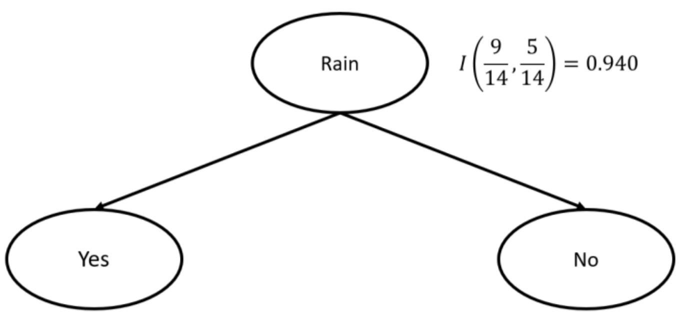
$I\left(\frac{9}{14},\frac{5}{14}\right) = 0.940$

3 Yes / 4 No | 6 Yes / 1 No

$I\left(\frac{3}{7},\frac{4}{7}\right) = 0.985$ | $I\left(\frac{6}{7},\frac{1}{7}\right) = 0.592$

Gain (Humidity)= $I\left(\frac{9}{14},\frac{5}{14}\right) - \frac{7}{14} * I\left(\frac{3}{7},\frac{4}{7}\right) - \frac{7}{14} * I\left(\frac{6}{7},\frac{1}{7}\right) = 0.152$

9 Yes / 5 No



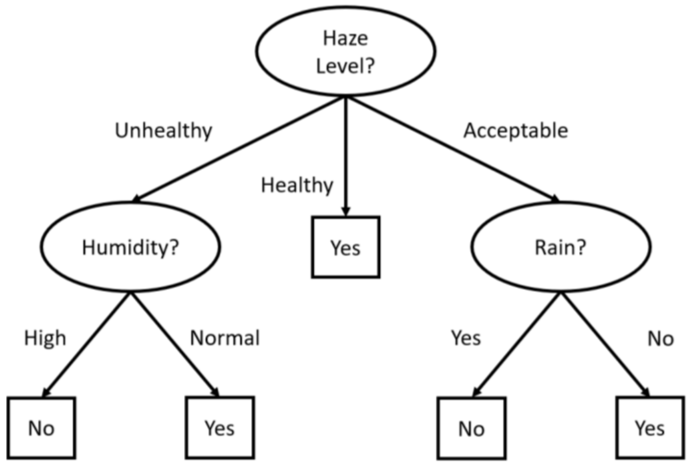
$I\left(\frac{9}{14},\frac{5}{14}\right) = 0.940$

3 Yes / 3 No | 6 Yes / 2 No

$I\left(\frac{3}{6},\frac{3}{6}\right) = 1$ | $I\left(\frac{6}{8},\frac{2}{8}\right) = 0.811$

Gain (Humidity)= $I\left(\frac{9}{14},\frac{5}{14}\right) - \frac{6}{14} * I\left(\frac{3}{6},\frac{3}{6}\right) - \frac{8}{14} * I\left(\frac{6}{8},\frac{2}{8}\right) = 0.048$

Based on first-level split on the four attributes above, Haze Level has the highest information gain of 0.246. Thus Haze Level is selected as the first node for the decision tree. Similar processes can be done for the rest three attributes to decide the second node on each branch. The resultant decision tree is:



It is not possible to do the complete second-level split given the time constraint in exam. Thus, do it smartly. You may notice that the second node is obvious for each branch.

b) Let c and p be the following:
c: A patient has liver cancer.
p: A patient receives a positive lab test.
Given P(c) = 0.006, P(p|c) = 0.97, P($\neg p|\neg c$) = 0.98, find P(c|p).
Based on Bayes' Law, P(c|p) = $\frac{P(p|c)P(c)}{P(p)}$, where $P(p) = P(p|c)P(c) + P(p|\neg c)P(\neg c)$[Law of total probability].
$P(\neg c) = 1 - P(c) = 0.994, P(p|\neg c) = 1 - P(\neg p|\neg c) = 0.02.$
Thus, $P(c|p) = \frac{0.97*0.006}{0.97*0.006+0.02*0.994} = 0.226$
Since P(c|p) is far less than 1, we should not diagnose the patient as having liver cancer. Rather, it's highly possible that it's an incorrect test result.

c)

i) $P(B, C, \neg G, J) = P(B) * P(C|B) * P(\neg G|B, C) * P(J|\neg G)$
$= 0.9 * 0.3 * (1 - 0.8) * 0 = 0$

ii) Find $P(J|B, C)$.

$$P(J|B,C) = \frac{\sum_G P(J,B,C,G)}{P(B \wedge C)} = \frac{\sum_G P(B) * P(C|B) * P(G|B,C) * P(J|G)}{P(B \wedge C)}$$
$$= \frac{P(B) * P(C|B) * \sum_G P(G|B,C) * P(J|G)}{P(B \wedge C)}$$
$$P(B) * P(C|B) * \sum_G P(G|B,C) * P(J|G) = 0.9 * 0.3 * (0.8 * 0.3 + 0.2 * 0) = 0.0648$$
$$P(B \wedge C) = P(C|B) * P(B) = 0.3 * 0.9 = 0.27$$
Thus, $P(J|B, C) = \frac{0.0648}{0.27} = 0.24$

Remarks: *This set of questions is pretty straightforward, You can definitely ace it so long as you understand the concepts in lecture. As you might see, Question 3 & 4 are much more time-consuming than the first 2 questions. Thus, do allocate your time wisely in exam. By the way, you might notice large variations among different sets of PYP. Take it easy as the course scope keeps changing for the past few semesters. And my apology for any mistake in the solution! All the best!*

--End of Answers--