Assignment 1: Chapters 1, 2, Game Theory.
(Solutions)

NB: The clarity of your answers was considered when grading.

Chapter 1. AI Foundations. 21 points total.

1. (5 points) Consider these two statements:
   - "Animals can do only what their genes tell them".
   - "Therefore animals cannot be intelligent"

   Is the first statement true, and does it imply the second?

The point of this exercise is to notice the parallel with the next one. Whatever you decide about whether computers could be intelligent in 2, you are committed to making the same conclusion about animals (including humans), *unless* your reasons for deciding whether something is intelligent take into account the mechanism (programming via genes versus programming via a human programmer). Note that Searle makes this appeal to mechanism in his Chinese Room argument (see Chapter 26).

2. (5 points) Consider these two statements.
   - "Computers can do only what their programmers tell them".
   - "Therefore computers cannot be intelligent"

   Is the first statement true, and does it imply the second?

This depends on your definition of "intelligent" and "tell." In one sense computers only do what the programmers command them to do, but in another sense what the programmers consciously tells the computer to do often has very little to do with what the computer actually does. Anyone who has written a program with an ornery bug knows this, as does anyone who has written a successful machine learning program. So in one sense Samuel "told" the computer "learn to play checkers better than I do, and then play that way," but in another sense he told the computer "follow this learning algorithm" and it learned to play. So we're left in the situation where you may or may not consider

learning to play checkers to be a sign of intelligence (or you may think that learning to play in the right way requires intelligence, but not in this way), and you may think the intelligence resides in the programmer or in the computer.

NB: For question given full grade if your reasoning about intelligence was consistent fo computers or at least if you clearly explained that you assume the n computer) matters.

3. (5 points) Fo nction, is there at most one agent program that implements a n? If you answer "yes", show why there is at most one agent pro er "no", give an example of how there can be two or more.

The answer is "no" and you could provide any reasonable example. For instance, assume we are given an agent function whose actions only depend on the previous p percepts. One program can remember the previous p percepts to implement the agent function, while another could remember greater than p percepts and still implement the same agent function.

4. (6 points) Match the following concepts/statements with a discipline related to AI. Give your answer by filling in letters in the table provided. For example, if you think that "Statistics" best matches "Decision Theory", put a "b" into the empty square next to "Decision Theory".

	a. Philosophy
	b. Mathematics
	c. Economics
	d. Statistics
	e. Psychology
	f. Homeostasis
The mind operates according to rules.	a
Decision Theory	c
Laws of probability	d
Behaviourism	e
Computation Theory	b
Control Theory	f

Chapter 2. Agents. 14 points total.

4. a. (6 points) Fill in the table below for Watson to describe its environment when Watson plays Jeopardy. This question refers to Watson as a whole system in the Youtube video. Briefly explain your answer.

Observable	A	nistic	Episodic	Static	Discrete

Observable	Ag		Episodic	Static	Discrete
Partially (don't know what questions are hidden)	mi	is or Nondeterministic responses from other players	Yes Question by question. No if taking into account accumulated money from previous bets.	Semi Time ticks down. Dynamic when other players may move first.	Yes But potentially very large set of possible answers. (Could also be continuous if you consider bets as continuous, or probabilities as outputs)

4.b. (8 points) Specify a PEAS model for IBM's Watson system.

Performance: winnings, coming first.
Environment: questions to solve, other players' actions.
Actuator: output answers, amounts to bet on screen.
Sensors: reads questions via text message, sees other players' bets.

Game Types. 10 points total.

Summary. Different payoff numbers can represent the same type of game. What matters is not so much the exact qualitative relationships, like which numbers are bigger than others and dominant strategies are. To transform a game matrix into another you can always change a player's utility function *u* by a positive linear transformation: add constants or multiply by a positive number. For example, if *u* is the utility function for a rational agent, then the linear transformation

$u'(x) = -5 + 3u(x)$

defines a new utility function. If you can transform one game matrix into another using a positive linear transformation, then they represent the same game.

1. (5 points) What type of game (i.e., BoS, PD, etc. – see lecture notes) does the following game matrix represent? Write down a positive linear transformation for each player that transforms the game matrix shown into the one shown in the lecture notes.

	L	R
T	-1,-1	-3,1
B	1,-3	-2,-2

Solution: Prisoner's Dilemma.
(ii) game matrix from lecture notes

	L	R
T	0,0	-2,2
B	2,-2	-1,-1

Linear transformations that transform game matrix (i) into (ii):

- Row player: $U_r'(x) = a*U_r(x) + b$ where $U_r'(x)$ is the utility function for the row player obtained from matrix (i) and $U_r(x)$ is the utility function for the row player obtained from matrix (ii).

Solve the linear system for constants a and b:

$0 = a*(-1) + b \rightarrow a = b$

$2 = a*(1) + b \rightarrow b = 1$

$\rightarrow U_r'(x) = U_r(x)+1$

- Column player: same reasoning as above

$\rightarrow U_c'(x) = U_c(x)+1$

NB: you were given full grade if you explained the steps leading to finding the linear transformation.

2. (5 points) Wh…e., BoS, PD, etc. – see lecture notes) does the following game …ite down a positive linear transformation for each player that transf… shown into the one shown in the lecture notes.

		R
T		-1,-4
B	-1,-4	5,-1

BoS
(ii)

	L	R
T	1,2	0,0
B	0,0	2,1

Linear transformations:

1) Row: $U_r'(x) = a*U_r(x)+b$

-1 a + b = 0 → a = b

5 a + b = 2 → a = 1/3 , b = 1/3

→ **$U_r'(x) = 1/3*U_r(x)+1$**

2) Col: $U_r'(x) = a*U_r(x)+b$

2 a + b = 2

-4 a + b = 0 → a = 1/3 , b = 4 /3

→ **$U_r'(x) = 1/3*U_r(x)+4/3$**

Nash Equilibrium Analysis. 16 points.

Summary. Finding the Nash equilibria of a game is the first step in game-theoretic analysis.

1. (8 points) BoS: Find all deterministic Nash Equilibria, and at least one mixed Nash equilibrium.

	L	R
T	2,4	0,0
B	0,0	4,2

Sample Answer (not necessarily the solution):

[T,L] is the only (uilibrium.
[p(T) = ½, p(L) = equilbirum.

Reminder:

1. A pure or quilibria (NE) is a pure or deterministic pair of strategies s a best response (maximizes payoff) against s2 and s2 is a t s1. In the case of a single move game, a pure strategy is a single action.
2. A mixed NE is a pair of mixed strategies that are each the best response to the other. A mixed strategy is a randomized strategy that selects actions according to a probability distribution.

BoS deterministic NE: (T, L) and (B,R)*.
If row player chooses T, best solution for column player is to choose L and vice versa.
If row player chooses B, best solution for column player is R and vice versa.
*These are deterministic NE because if any player moves to a different strategy, that player will have a worst utility.

BoS Mixed NE:

- Mixed strategy for row player = p
- Mixed strategy for col player = q

	L q	R (1-q)
T p	2,4	0,0
B (1-p)	0,0	4,2

- If row chooses T with probability p, the optimal strategies for Col are:
  - Expected Payoff(L, p) = p x 4 + (1-p) x 0 = 4p
  - Expected Payoff(R, p) = p x 0 + (1-p) x 2 = 2-2p
- According to the support theorem (if Col player randomizes, both L and R are equally good strategies):
  - Payoff(L,p) = Payoff(R,p) ⇔ 2p = 1-p ⇔ p = 1/3
  - In any mixed equilibrium, row player chooses T or B with P(T) = 1/3 and P(B) = 2/3
- If col chooses L with probability q, the optimal strategies for row are:
  - Payoff(T,q) = q x 2 + (1-q) x 0 = 2q
  - Payoff(B,q) = q x 0 + (1-q) x 4 = 4-4q
- Support theorem:
  - Payoff(T,q) = Payoff(B,q) ⇔ q = 2-2q ⇔ q = 2/3

➔ Mixed NE is P(L) = P(B) = 2/3 and P(T) = P(R)=1/3

2. (4 points) The Game of Chicken (featured in movies "Rebel Without a Cause", and "Charlie's Angel") [illegible] heading towards each other. Whoever turns away first, loses. A hist[illegible] Cuba crisis between the USSR and the USA.

		Keep Going
Keep Going		-15, -15
Turn		1, 3

a. (2) Find all deterministic Nash Equilibria.

b. (2) Find at least one mixed Nash equilibrium in the Game of Chicken.

Deterministic Nash Equilibria.
(Keep Going, Turn), (Turn, Keep Going)

Mixed Nash Equilibrium:
P(row → Keep Going) = 3/19, P(col → Turn) = 16/19

NB: full grade if you explained the steps to find a deterministic and mixed NE.

3. (4 points) An issue that arises in technology industries is that an inferior standard may become entrenched even if a better one is available. A historical example is the use of VHS tapes vs. Beta. Or Facebook vs. GooglePlus? This illustrates *network effects*: users like to use technology used by others. Let's consider a simple game-theoretic model of this situation.

	User 2	
User 1	Superior technology	Inferior technology
Superior technology	3, 3	1, 1
Inferior technology	1, 1	2, 2

a. (2) Find all deterministic Nash Equilibria.

b. (2) Find at least one mixed Nash equilibrium in this game.

Deterministic Nash Equilibria: These are cases where neither side randomizes, (Superior, Superior), (Inferior, Inferior)

Mixed Nash Equilibrium:
When col randomizes, p = 1/3
When row randomizes, q = 1/3

NB: **full grade if you explained the steps to find a deterministic and mixed NE.**

Assignment 2: Chapters 4, 3, 5.

(SOLUTIONS)

NB: clarity of you red in the final grade.

Chapter 4. Local Search in Continuous Spaces. 26 points.

1.

a. (3) Write down the gradient (derivative) of the function you chose. (Hint: this is probably easier for *l*.)

$$\frac{\partial l}{\partial p} = -\left(\frac{6}{p} - \frac{4}{1-p}\right)$$

b. (10) Try to get close to the minimum in 5 gradient descent steps. Use as your initial guess $p=1/2$ (the coin is fair), and for the first 3 step sizes, use 0.04, 0.02, 0.01. The last 2 step sizes you can choose for yourself. For your answer fill in the table below.

step	p	step size
0	0.5	0.04
1	0.66	0.02
2	0.607	0.01
3	0.604	0.005
4	0.603	0.0025
5	0.603	

Step size of 4 and 5 can be other reasonable values (e.g. both 0.1).

2.

a. (3) Write down the second-order gradient (derivative) of the function you chose. (Hint: this is probably easier for *l*.)

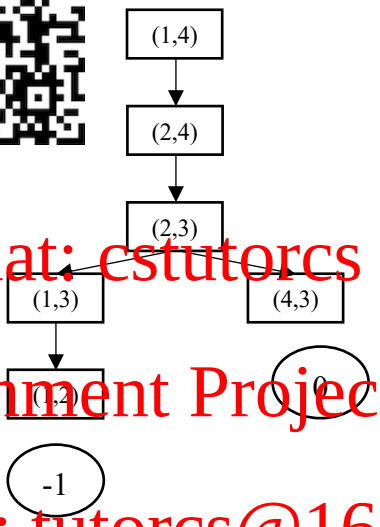
$$\frac{\partial^2 l}{\partial p^2} = \left(\frac{6}{p^2} + \frac{4}{(1-p)^2}\right)$$

b. (10) Show the results of 5 Newton-Raphson steps. Use as your initial guess $p=1/2$ (the coin is fair).

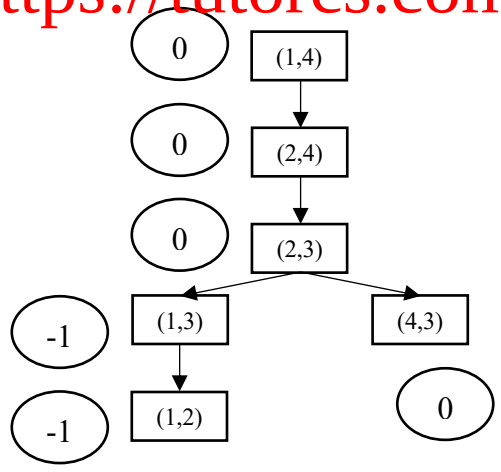
	p
0	0.5
1	0.6
2	0.6
3	0.6
4	0.6
5	0.6

Chapter 5: Adversarial Search. 26 points totals.

1. (10) Draw the complete game tree, using the following conventions.
   a. Write [illegible] where s_A and s_b denote the token locations.
   b. Put ea[illegible] square box and write its game value in a circle.



2. (10) Now mark each node with its backed-up minimax value (also in a circle).



3. (6) Suppose tl ed states and otherwise kept the rules the same.
   a. Descri gy for each player (informally but precisely).

   For Playe herself moving in right direction.

   For Playe herself moving in left direction.

   b. Does the standard minimax algorithm terminate with a strategy for each player? If yes, does it find the optimal strategy (from part a)? If not, how could you modify the standard minimax algorithm so that it terminates after finding the optimal strategy for each player in this game? The modification should be general in the sense that the modified algorithm can be applied to any game tree, not just this game.

Standard minimax is depth-first and would go into an infinite loop. It can be fixed by comparing the current state against the stack and if the state is repeated, then return a "?" value. Propagation of "?" values is handled as above. Although it works in this case, it does not always work because it is not clear how to compare "?" with a drawn position; nor is it clear how to handle the comparison when there are wins of different degrees (as in backgammon). Finally, in games with chance nodes, it is unclear how to compute the average of a number and a "?". Note that it is not correct to treat repeated states automatically as drawn positions; in this example, both (1,4) and (2,4) repeat in the tree but they are won positions.

What is really happening is that each state has a well-defined but initially unknown value. These unknown values are related by the minimax equation at the bottom of 164. If the game tree is acyclic, then the minimax algorithm solves these equations by propagating from the leaves. If the game tree has cycles, then a dynamic programming method must be used, as explained in Chapter 17. These algorithms can determine whether each node has a well-determined value (as in this example) or is really an infinite loop in that both players prefer to stay in the loop (or have no choice). In such a case, the rules of the game will need to define the value (otherwise the game will never end). In chess, for example, a state that occurs 3 times (and hence is assumed to be desirable for both players) is a draw.

Other modifications, like using a iterative depth first search, are also acceptable.

Decision Tree Learning: 10 points total

Summary. The purpose of this assignment is to give you practice with computing expected information gain and how the ID3 algorithm uses it to construct a decision tree.

…er a customer will wait for a table in a restaurant or …the Hun(gry) attribute. Show the following for the …

…y ID3. (3 points)

next attribute chosen can be both have max gain = 0.292. …e Pat that will split to two node.

2. The expected information gain associated with the next attribute. There may be more than one attribute with the same information gain, finding one is sufficient. (3 points)

$E = -\frac{2}{7}\log_2(\frac{2}{7}) - \frac{5}{7}\log_2(\frac{5}{7}) = 0.8631$

Pat.

$E(\text{some}) = -\frac{3}{3}\log_2(\frac{3}{3}) - 0 = 0.$

$E(\text{full}) = -\frac{2}{4}\log(\frac{2}{4}) - \frac{2}{4}\log(\frac{2}{4}) = 1.$

$\text{Gain}(\text{Pat}) = 0.8631 - \frac{3}{7}\cdot 0 - \frac{4}{7}\cdot 1$

$= 0.292.$

$\text{Gain}(\text{Est}) = 0.292.$

3. How you calculated the expected information gain. (3 points)

I calculated the Entropy of each attributes first. Entropy = $\sum_i -P_i \log_2 P_i$. Then I generated P(x) for each attribute. Information Gain = E(Hungry) − [average entropy (children)].

I used matlab to help calculate each expected information gain.

```
Entropy= 0.863121
Gain(Alt)= 0.169584
Gain(Bar)= 0.005978
Gain(Fri)= 0.005978
Gain(Pat)= 0.291692
Gain(Price)= 0.183851
Gain(Rain)= 0.169584
Gain(Res)= 0.005978
Gain(Type)= 0.183851
Gain(Est)= 0.291692
>>
```

Trace Backpropagation. 13 points total.

	a	b	d
x_1		0	1
x_2		1	0

Table 1. Th

Definition	Derivative
σ(x) = 1/(	dσ/dx = σ(x)(1-σ(x))

1. Using the notation in the text, show the formulas for computing the following quantities, for any node *i*: (4 points)

- in_i
- a_i
- Δ[i]
- Weight updates for w_j

- $in_i = \sum_{j=0}^{n} w_{ji} a_j$.
- $a_i = g(in_i) = g(\sum_{j=0}^{n} w_{ji} a_j)$.
- $\Delta[i] = g'(in_i) \sum_j w_{ij} \Delta_j$.
- Weight update for w_j: $W_{ij} = W_{ij} + \alpha * x_i + \Delta_j$.

2. Fill in the following table using the formulas from above. You can expand this to include more information (e.g. derivatives of activation functions) if you like. (9 points)

Iteration	a_c	Δ[c]	a_d	Δ[d]	w_{0c}	w_{ac}	w_{bc}	w_{cd}	w_{0d}
x_1	0.5987	0.0049	0.5793	0.1025	0.2015	0.2015	0.2000	0.2184	0.2308
x_2	0.599	-0.0075	0.5894	-0.1426	0.1992	0.2015	0.1978	0.1928	0.1880

x_1: $in_c = W_{0c} \cdot a_0 + W_{ac} \cdot a_a + W_{bc} \cdot a_b$.
$in_d = W_{0d} \cdot a_0 + W_{cd} \cdot a_c$.
$a_c = \frac{1}{1+\exp(-in_c)}$ $a_d = \frac{1}{1+\exp(-in_d)}$

$\Delta[d] = \frac{1}{1+\exp(-in_d)} \cdot (1 - \frac{1}{\exp(-in_d)}) \cdot (d - a_d)$
$\Delta[c] = \frac{1}{1+\exp(-in_c)} \cdot (1 - \frac{1}{\exp(-in_c)} \cdot (W_{cd} \cdot \Delta[d]))$
$W_{0c} = W_{0c} + 0.3 \cdot a_0 \cdot \Delta[c]$
$W_{ac} = W_{ac} + 0.3 \cdot a_a \cdot \Delta[c]$.

$W_{bc} = W_{bc} + 0.3 a_b \cdot \Delta[c]$
$W_{cd} = W_{cd} + 0.3 a_c \cdot \Delta[d]$
$W_{0d} = W_{0d} + 0.3 a_d \cdot \Delta[d]$

Assignment 4: Probabilistic Reasoning and Learning

Instructions: The universi[illegible]ic dishonesty and plagiarism (cheating) will be taken very seriously in this course. *Everything submitted should be your own writing or coding.* You must not let other students copy your work. On your assignment, put down your **name**, the number of the assignment and the number of the course. Spelling and grammar count. You should show your work to get the result in order to get the full mark.

Group Work: Discussions of the assignment is okay, for example to understand the concepts involved. If you work in a group, put down the name of all members of your group. There should be no group submissions. Each group member should write up their own solution to show their own understanding.

For the due date please see our course management server https://courses.cs.sfu.ca. The time when you upload your assignment is the official time stamp. If your assignment is late because you did not figure this out soon enough, you will lose marks according to the syllabus policy.

Terminology: The questions are not self-explanatory. Even ordinary English words (e.g., “rationality”) may not have their ordinary meaning in an AI context. Part of your task is to learn the AI terminology required to understand the questions. You will likely not understand the questions if you have not studied the course material.

Getting Help. Check the syllabus for communication policy. You have the textbook, the lecture notes, the discussion forum, and you can ask us in office hours or class sessions. We do not provide individual email support.

Handing in the Assignment. Please use the submission system on courses.cs.sfu.ca.

You should post

- a pdf document that contains your written answers, as well as the screenshots and diagrams required.
- A Bayesian network in executable file format (.xml or .bif), for the power station problem.

Probabilistic Reasoning With Bayesian Networks
78 points total.

Go to www.aispace.org and start the "belief and decision network" tool. Load the sample file "File/Load Sample P... nostic Example". We will use this to test some of the basic probability law... an do many of these calculations for you, but the purpose of the exercise is to l... chind the calculations. You can use the tool to check your answers, but *you sho... urself using the probability calculus together with the conditional probabil... n network.* If you have difficult running the tool in the browser, try downloa...

Joint probabilities

1. P(all nodes false)
= P(Influenza = F) * P(Smokes = F) * P(Sore Throat = F | Influenza = F) * P(Fever = F | Influenza = F) * P(Bronchitis = F | Influenza = F and Smoke = F) * P(Coughing = F | Bronchitis = F) * P(Wheezing = F | Bronchitis = F)
= 0.95 * 0.8 * 0.999 * 0.95 * 0.9999 * 0.93 * 0.999
= 0.670051

2. P(Sore Throat = T, all other nodes false)
= P(Influenza = F) * P(Smokes = F) * P(Sore Throat = T | Influenza = F) * P(Fever = F | Influenza = F) * P(Bronchitis = F | Influenza = F and Smoke = F) * P(Coughing = F | Bronchitis = F) * P(Wheezing = F | Bronchitis = F) =
0.95 * 0.8 * 0.001 * 0.95 * 0.9999 * 0.93 * 0.999
= 0.000671

3. P(all nodes others than Sore throat false) = P(Sore Throat = T, all other nodes false) or P(Sore Throat = F, all other nodes false) = P(Sore Throat = T, all other nodes false) + P(all nodes false)
= 0.670051 + 0.000671
= 0.670722

4. P(Sore Throat = F | all other node are false) * P(all other nodes are false)
= 0.999 * 0.670722
= 0.670051
= P(all nodes false)

5. P(Sore Throat = T, Fever = T)
= P(Sore Throat = T | Influenza = T) * P(Fever = T | Influenza = T) P(Influenza = T) + P(Sore Throat = T | Influenza = F) * P(Fever = T | influenza = F) * P(Influenza = F)
= 0.3 * 0.9 * 0.05 + 0.001 * 0.05 * 0.95
= 1.35475 * 10^(-2)

Independence
20 points

"If no evidence is observed, Influenza and Smoking are probabilistically independent."

1. (10) Prove the numerical semantics, i.e. show that the values of Influenza and Smoking are independent for all possibilities. You may use queries in the tool.
   i. Must show values of joint.

Influenze	Smoking	product	joint
TRUE	TRUE	0.01	0.01
TRUE	FALSE	0.04	0.04
FALSE	TRUE	0.19	0.19
FALSE	FALSE	0.76	0.76

Checking the equivalent of P(A)=P(A|B) is also accentable.

$$P(A \cap B) = P(A)P(B) \Leftrightarrow P(A) = \frac{P(A)P(B)}{P(B)} = \frac{P(A \cap B)}{P(B)} = P(A \mid B)$$

2. (10) Prove this from the topological semantics, i.e. using the Markov condition that given values for its parents, a variable is probabilistically independent of all its nondescendants.

Neither has a parent, so the condition holds vacuously. Neither is a descendant of the other, so they are independent.

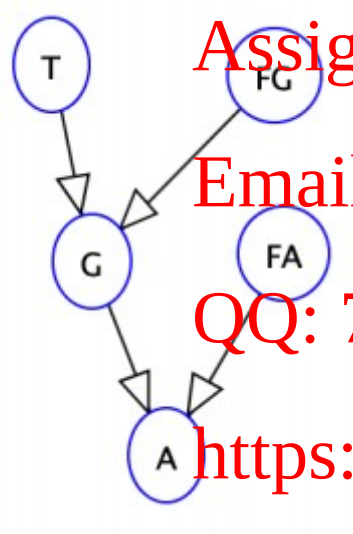
Bayesian Networks and Constraints (30 points total)

In a power station, an alarm senses when a temperature gauge exceeds a given threshold. The gauge measures the temperature of the core. Using the AIspace tool, draw a Bayesian network that represents the following information about the plant. You may consider editing your Baye to facilitate copying of similar entries.

1. (5 points) Cre he following variables (=nodes) and domains. Each domain possible locations.

Variable	Do ing
A	{T,F} nds
FA	{T,F} aulty
FG	{T,F} gauge is faulty
G	{Normal, High} gauge reading
T	{Normal, High} core temperature

The graph should look like this



2. Draw edges between the nodes and specify conditional probability tables to represent the following information.
    a. (6) The alarm works correctly unless it is faulty. If it is faulty, it never sounds.

FA	G	P(A=T)	P(A=F)
T	Normal	0.0	1.0
T	High	0.0	1.0
F	Normal	0	1.0
F	High	1	0.0

b. (6) If the gauge is not faulty, there is a 90% chance that it gives the correct temperature. If the gauge is faulty, there is a 10% chance that it gives the correct temperature.

FG		ormal)	P(G=High)
T			0.9
T			0.1
F			0.1
F	High	0.1	0.9

c. (10) Suppose the alarm and gauge are working correctly and the alarm sounds. Using the probability calculus probability, and the various conditional probabilities in the network, calculate the probability that under these circumstances, the temperature of the core is high. You can make the following assumptions: (i) The temperature is normal 70% of the time, high 30% of the time. (ii) The gauge is faulty 10% of the time, not faulty 90% of the time.

Since the gauge is not faulty, and the alarm is true, we know that the gauge must be high(G=H).

Moreover, given gauge=high, from the Bayes net, we can conlude that alarm and faulty alarm is conditional independent to the temperature and faulty gauge.

P (T= High | A = True, FA = False, FG = False. G = High)

= P (T= High | FG = False, G = High)

= P (T= High, FG = False, G = High) / P (FG = False, Gauge = High)

Now P (FG = False, Gauge = High) =
P (T=High, FG = False, Gauge = High) + P (T= Normal, FG = False, Gauge = High) = **((0.9*0.3*0.9) + (0.7*0.9*0.1)**

So **P (T= High, FG = False, G = High) / P (FG = False, Gauge = High)**

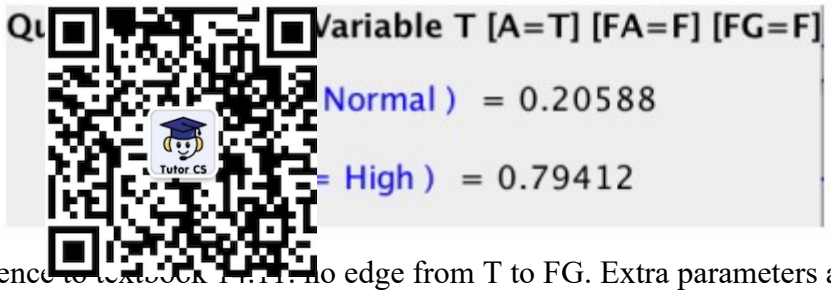
= (0.3*0.9*0.9) / ((0.9*0.3*0.9) + (0.7*0.9*0.1)

= 0.243/ (0.243 + 0.063) = 0.243/0.306

P (T=high, FG = False, G = High) = 0.79411764711

d. (3) Use the AIspace query tool to confirm your answer from c.
According to our solution for c), you can use any estimate you like for the

prior P(FA=T), e.g. 50% or 80%. We suggest experimenting with the tool to verify that you get the same result as c) no matter what prior value you use. e.



Difference to Task3ck 14.1: no edge from T to FG. Extra parameters are specified in the problem.

What to Submit for this question. Everything specified in the problem, including answers and brief explanations. In addition, A loadable .xml or .bif file for your Bayesian network in part 1. A screenshot of your Bayesian network, screenshots of your CP-tables, and a screenshot of your query and the answer from AIspace.