MSIT6000F 2019 Fall Semester Assignment #3

Date assigned: November 12, 2019

Due time: 11:59PM on December 3, 2019

How to submit it: Submit the written part a pdf file, and the programming part as a zip archive on canvas.ust.hk

Penalties on late papers: 20% off each day (anytime after the due time is considered late by one day)

1 The Written Part

Problem 1 (10 pts) Given the following, can you prove that the unicorn is mythical? How about magical? Horned?

> If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.

Problem 2 (10 pts) Represent the following sentences in first-order logic, using a consistent vocabulary (which you must define):

a. Not all students take both History and Biology.

b. Only one student failed History.

c. Every person who dislikes all vegetarians is smart.

d. No person likes a smart vegetarian.

e. There is a student who does homework for those and only those who do not do homework for themselves.

Problem 3 (10 pts) Sam, Clyde, and Oscar are elephants. We know the following facts about them:

1. Sam is pink.
2. Clyde is gray and likes Oscar.
3. Oscar is either pink or gray (but not both) and likes Sam.

Use resolution refutation to prove that a gray elephant likes a pink elephant; that is, prove $\exists x, y[Gray(x) \wedge Pink(y) \wedge Likes(x, y)]$.

Problem 4 (10 pts) Consider a company in Hong Kong that decides whether to hire someone based on the following features:

- UST: true if the applicant graduated from HKUST.
- HKU: true if the applicant graduated from HKU.
- CU: true if the applicant graduated from CU.
- GPA: true if the applicant received good grades in school.
- REC: true if the applicant got good recommendation letters.
- EXP: true if the applicant had prior experience in related jobs.

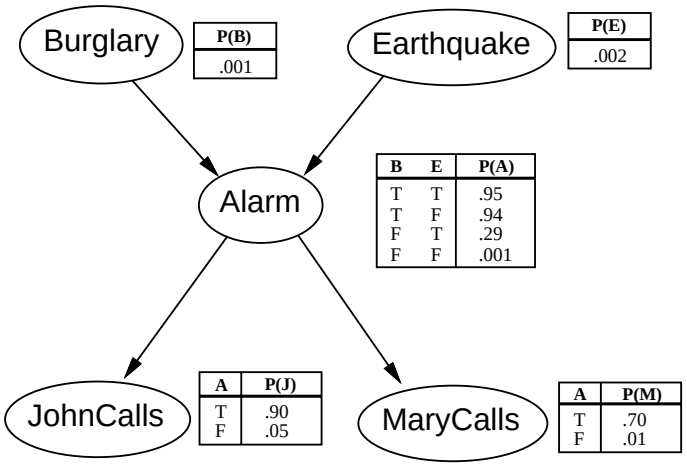
Suppose that you're hired by the company to develop a system that will help the compnay to decide whom to hire, and are given the following examples from the company's hiring record:

Example	Attributes						Hire?
	GPA	UST	HKU	CU	REC	EXP	
1	1	0	1	0	1	1	1
2	1	1	0	0	0	0	0
3	1	0	1	0	1	1	1
4	1	1	0	0	1	0	1
5	1	0	0	1	1	1	1
6	1	0	0	1	1	0	0
7	1	0	1	0	1	0	0
8	0	1	0	0	0	1	0
9	0	0	1	0	0	0	0
10	0	0	0	1	1	0	0
11	0	0	0	1	1	0	0

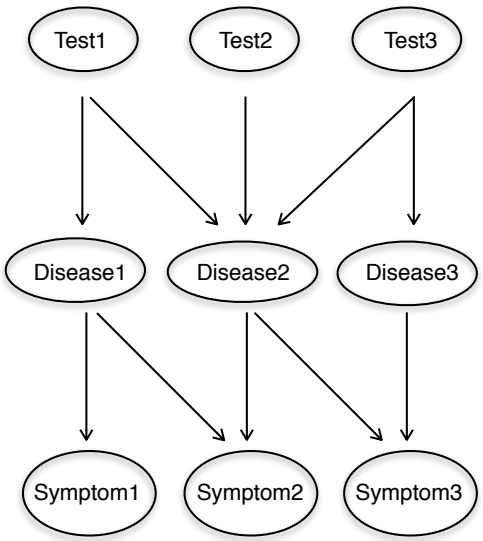
Use GSCA to learn a set of rules from these examples about when to hire an applicant based on the values of the given attributes.

Problem 5 (10 pts)

Compute the probablity of $JohnCalls$ given $MaryCalls$ in the following network:



Problem 6 (10 pts) Consider the following belief network about blood tests (positive or negative), diseases, and symptoms:



For the following first three questions, explain your answers, but for the last two, just give your answers. To explain your answers: if yes, either show the d-separation by enumerating all undirected paths between the two variables in question or pointing out an obvious pattern in all such paths; and if no, give a path showing that the d-separation is not true;

1. Are `Test1` and `Test2` independent?
2. Are `Disease1` and `Disease2` independent?
3. Are `Disease1` and `Disease3` independent?
4. Describe set E under which `Disease1` and `Disease2` are conditionally independent.
5. Describe set E under which `Disease1` and `Disease3` are conditionally independent.

Problem 7 (5 pts) (Russell and Norvig) The following payoff matrix shows a game between politicians and the Federal Reserve in the US:

	Fed: contract	Fed: do nothing	Fed: expand
Pol: contract	1,7	4,9	6,6
Pol: idle	2,8	5,5	9,4
Pol: expand	3,3	7,2	8,1

The politicians can expand (increase spending) or contract (cut spending) fiscal policy (or stay idle), while the Fed can expand (lower interest rate) or contract (increase interest rate) monetary policy (or do nothing). Find the Nash equilibria of this game.

problem 8 (10 pts). Consider an auction by first price with two agents, with ties broken randomly. It's a common knowledge that both agents have value 6 for the item, and they bid only positive integers up to their value. Formulate this auction as a game and find all Nash equilibria if there is any.

2 The Programming Part

The following two programming questions are exercises for you to use a mechanical theorem prover to solve problems by reasoning or finding models. The theorem prover you need to use is Z3. The official release is here:
`https://github.com/Z3Prover/z3`
If you use unix, you can install it using pip:
`https://pypi.org/project/z3-solver/#description`
Here is a good description of how to install it by Bob Moore of UT Austin:
`http://www.cs.utexas.edu/users/moore/acl2/manuals/current/manual/index-seo.php/SMT____Z3-INSTALLATION`

Problem 1: Lady or Tiger (10 pts) There are three rooms. Each contains either a lady or a tiger but not both. Furthermore, one room contained a lady and the other two contained tigers. Each of the rooms has a sign, and at most one of the three signs was true. The three signs are:

- Room I: A TIGER IS IN THIS ROOM.
- Room II: A LADY IS IN THIS ROOM.
- Room III: A TIGER IS IN ROOM II.

Which room contains the lady?

Problem 2: The Ranking Problem (15 pts) Given the following facts:

1. Lisa is not next to Bob in the ranking

2. Jim is ranked immediately ahead of a biology major
3. Bob is ranked immediately ahead of Jim
4. One of the women (Lisa and Mary) is a biology major
5. One of the women is ranked first

What are possible rankings for the four people?

What to submit

For the programming part, you must submit a zip file named as `YourStudentID.zip`, which includes:

- for Problem 1: `lady.py`, which is a z3 query file for your answer. For example, if your answer is that the lady is in room 2, and you use l2 to denote it, then your z3 query file is to check whether not(l2) is consistent with the KB of this problem. For example, the following python code is a z3 query file to check if q follows from p and $p \supset q$:

```
-------------------------------------------------------------
from z3 import *

p = Bool('p')
q = Bool('q')
s = Solver()
s.add(Implies(p,q))
s.add(p)
# to prove q, add not(q) to see if it causes a contradiction
s.add(not(q))
print(s.check())
-------------------------------------------------------------
```

- for Problem 2: `ranking.py`, which is your z3 query file to find a model from which to read out a ranking. For example, the following is a python code to compute a model of $p \vee q$ and $\neg(p \wedge q)$:

```
-------------------------------------------------------------
from z3 import *

p = Bool('p')
q = Bool('q')
s = Solver()
s.add(or(p,q))
```

```
s.add(not(and(p,q)))
print(s.check())
print(s.model())
----------------------------------------------------------------
```