UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR08010 INFORMATICS 2D: REASONING AND AGENTS

Thursday 30 $^{\underline{\text{th}}}$ April 2015

14:30 to 16:30

INSTRUCTIONS TO CANDIDATES

1. **Answer Parts A, B and C.**
2. **The multiple choice questions in Part A are worth 50% in total and are each worth the same amount. Mark one answer only for each question - multiple answers will score 0. Marks will not be deducted for incorrect multiple choice exam answers.**
3. **Parts B and C are each worth 25%. Answer ONE question from Part B and ONE question from Part C.**
4. **Use the special mark sheet for Part A. Answer Parts B and C each in a separate script book.**

CALCULATORS ARE PERMITTED.

Convener: D. K. Arvind
External Examiner: C. Johnson

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

Part A
ANSWER ALL QUESTIONS IN PART A. Use the special mark sheet.

1. Which of the following propositional logic formulae is a tautology?
   (a) $A \Rightarrow ((A \Rightarrow B) \Rightarrow B)$
   (b) $B \vee B$
   (c) $(B \wedge (A \Rightarrow B)) \Rightarrow A$
   (d) $A \wedge \neg A$
   (e) $(A \Rightarrow (B \Rightarrow A)) \Rightarrow A$

2. Consider a vocabulary with only four propositions, p, q, r, and s. How many models are there for the sentence $p \vee \neg q$?
   (a) 2
   (b) 12
   (c) 4
   (d) 6
   (e) 1

3. If two heuristics h_1 and h_2 are admissible, which one of the following is always admissible?
   (a) $h_1 + h_2$
   (b) $max(h_1, h_2)$
   (c) $h_1 \times h_2$
   (d) $h_1 + h_2^2$
   (e) h_1^2

4. Consider a constraint satisfaction problem with variables A, B, C, where the domain of A is $\{1, 2, 3, 4\}$, the domain of B is $\{1, 2\}$ and the domain of C is $\{2, 3\}$. The constraints are $A \neq C$, $B \neq C$. Which of the following statements is *false*:
   (a) The most constrained variables are B and C.
   (b) The most constraining variable is C.
   (c) The least constraining value of C is 2.
   (d) The domains are arc consistent with regard to all constraints.
   (e) The assignment $A = 1$, $B = 2$, $C = 3$ is a solution.

5. Which of the following, if any, is the most general unifier of

$$\{P(F(y), z), P(x, T), P(F(S), z)\}$$

where F is a function symbol, T/S are constants, and x/y are variables?

(a) $\{x/F(S), z/T\}$

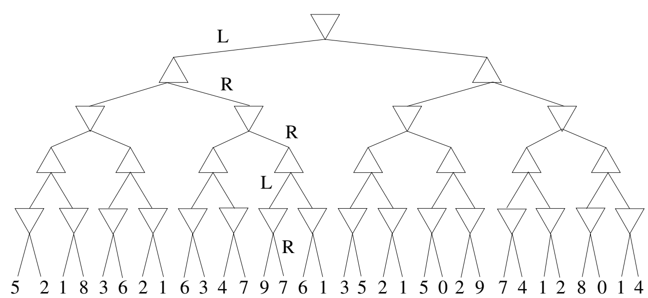
(b) $\{x/F(y), y/S, z/T\}$

(c) $\{P(F(S), T)\}$

(d) $\{x/y, y/z, z/F(S)\}$

(e) Unification fails

6. In the following game tree, nodes labelled Δ are where Max is due to play, and the symbol ∇ indicates Min is due to play. The numbers below the leaves are the final scores for Max. A complete strategy for both agents is a sequence of left-right steps, e.g. L, R, R, L, R for the labelled path.



Which of the following strategies will two optimal Minimax players play?

(a) L, L, R, L, R

(b) L, R, R, L, R

(c) R, L, R, R, L

(d) L, R, R, L , L

(e) R, R, R, L, R

7. Which of the following search algorithms is complete and optimal?

(a) Greedy best-first search

(b) Depth-first search

(c) Minimax search in infinite state spaces

(d) A* search with an admissible heuristic

(e) A* search with a heuristic that overestimates cost to goal

8. In the following question:

   - $At(sq, s)$ means the agent is at square sq in situation s.
   - $Heading(dir, s)$ means the agent is facing in direction dir in situation s.
   - $Next(sq1, dir, sq2)$ means that square $sq2$ is adjacent to square $sq1$ in direction dir.
   - $Forward$ is the action of moving one square in the direction the agent is facing.
   - $Result(a, s)$ is the situation resulting from executing the action a in situation s.

   Which of the following statements is a situation calculus FRAME rule that describes the $Forward$ action in this world?

   (a) $\forall sq, a, s.\, At(sq, s) \wedge \neg a = Forward \Rightarrow At(sq, Result(a, s))$
   (b) $\forall sq_1, dir, sq_2.\, At(sq_1, s) \wedge Heading(dir, s) \wedge Next(sq_1, dir, sq_2)$
   $\Rightarrow At(sq_2, Result(Forward, s))$
   (c) $\forall sq, a, s.\, At(sq, s) \wedge \neg a = Forward \Rightarrow At(sq, s + 1)$
   (d) $At(sq, Result(a, s)) \Rightarrow \forall sq, a, s.\, At(sq, s) \wedge \neg a = Forward$
   (e) $\forall sq, a, s.\, At(sq, s) \wedge \neg a = Forward \Rightarrow Result(a, At(sq, s))$

9. Assume you want to apply A^* search in a route finding problem on a map where the location of each city n is given by its integrer coordinates (x_n, y_n). The destination is known to be located (x^*, y^*). Which of the following heuristics would make the search most efficient?

   (a) $h(n) = |x^* - x_n| + |y^* - y_n|$
   (b) $h(n) = \sqrt{(x^* - x_n)^2 + (y^* - y_n)^2}$
   (c) $h(n) = x_n/x^* + y_n/y^*$
   (d) $h(n) = 0$
   (e) $h(n) = (x^* + y^*)$

10. Which of the following is *not* part of the DPLL algorithm, as given in Russell & Norvig and the lectures?

    (a) A sentence is false if any of its clauses is false.
    (b) A clause is true if one of its literals is true.
    (c) A pure symbol can be set to true.
    (d) A literal in a unit clause can be set to true.
    (e) A pure symbol can be set to false.

11. Using action schema

$$Action(Walk(from, to),$$
$$\quad \text{P}\textsc{recond}\text{:} At(from), City(to), from \neq to$$
$$\quad \text{E}\textsc{ffect}\text{:} (At(to), \neg At(from))$$

which of the following is the state that results from applying all executable actions (simultaneously) in state $At(Manchester) \wedge City(Glasgow) \wedge At(Home)$?

(a) $At(Manchester) \wedge City(Glasgow) \wedge At(Glasgow)$

(b) $At(Glasgow) \wedge At(Home) \wedge City(Glasgow)$

(c) $At(Glasgow) \wedge City(Glasgow)$

(d) $At(Glasgow) \wedge City(Glasgow) \wedge At(Home) \wedge At(Manchester)$

(e) $At(Glasgow) \wedge At(Manchester) \wedge City(Home) \wedge City(Glasgow)$

12. Which of the following statements is incorrect?

(a) Unlike problem-solving agents based on heuristic search, planning agents avoid the problem of having to define heuristic values for every possible state.

(b) State-space search based planning methods are never goal-directed.

(c) Problem-solving agents based on search will often explore irrelevant actions.

(d) Search-based agents may consider undoing the effects of actions achieved by previous actions.

(e) Partial-order planning ensures that the agent does not commit to overly restrictive courses of execution at the time of planning.

13. Assume you have to add an action D with postconditions $\neg q$ and $\neg p$ in a plan with existing causal links $A \xrightarrow{p} B$ and $B \xrightarrow{q} C$. Which of the following total orderings resolves all conflicts that might arise from addition of D?

(a) $A \prec B \prec C \prec D$

(b) $A \prec B \prec D \prec C$

(c) $A \prec D \prec B \prec C$

(d) $D \prec A \prec C \prec B$

(e) $A \prec C \prec D \prec B$

14. You are given four action descriptions:

$Action(A, \textsc{Precond}:\{X\}, \textsc{Effect}:\{(\textbf{when}\ P : \neg X), Z\})$
$Action(B, \textsc{Precond}:\{Y\}, \textsc{Effect}:\{(\textbf{when}\ Z : \neg P), \neg Y, \neg Z, X\})$
$Action(C, \textsc{Precond}:\{\neg X\}, \textsc{Effect}:\{(\textbf{when}\ Q : X)\})$
$Action(D, \textsc{Precond}:\{Z\}, \textsc{Effect}:\{(\textbf{when}\ Q : \neg Z)\})$

What state would result from executing the action sequence $[D, C, A, B]$ in the state $\{P, Q, Y, Z\}$?

(a) $\{P, Q, Y\}$
(b) $\{P, Q, Y, Z\}$
(c) $\{Q, X\}$
(d) The plan isn't executable because the preconditions for action A aren't met.
(e) The plan isn't executable because the preconditions for action B aren't met.

15. A house has 2 rooms, and the robot moves from one room to the other, cleaning each room. It can sense its location and whether the room it is currently in is currently clean or dirty. Sometimes when it cleans, its filter deposits dirt onto the floor. Moreover, rooms sometimes get dusty from the airconditioning system that runs throughout the house.

Which of the following statements is *incorrect*?

(a) The robot is working in a partially observable environment.
(b) The robot could use contingent planning to clean the house.
(c) A plan to clean a particular room in the house would include a loop.
(d) It is impossible for the robot to guarantee that it has achieved the goal of making each room in the house clean.
(e) The robot must use replanning to clean a room in the house.

16. Assume the following inhibition probabilities between Boolean cause variables A, B, C and Boolean effect variable X:

$$\begin{aligned} P(\neg x|a, \neg b, \neg c) &= p \\ P(\neg x|\neg a, b, \neg c) &= q \\ P(\neg x|\neg a, \neg b, c) &= r \end{aligned}$$

What is the probability $P(x|a, b, \neg c)$ assuming that the conditional probabilities of X are computed using a noisy-OR relation?

(a) $1 - pq$

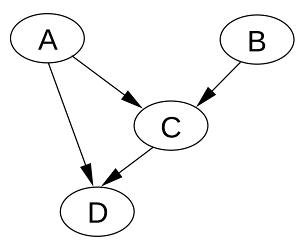
(b) $rp(1 - q)$

(c) $(1 - p)qr$

(d) $(1 - p)(1 - q)$

(e) $(1 - p)(1 - q)r$

17. Consider the following Bayesian network structure with Boolean variables:



Which of the following statements is correct?

(a) All conditional probability tables in Bayesian network have the same size.

(b) Given A, C is conditionally independent of B.

(c) Calculating the value of $P(a \wedge \neg b \wedge c \wedge \neg d)$ in this network involves using only multiplication of probabilities.

(d) All variables in the system are conditionally independent.

(e) It takes a total of $2^4 = 16$ probability values to describe the joint probability distribution represented by the network.

18. The following formula describes the "forward" part of the algorithm used for prediction in temporal probabilistic models:

$$f_{1:t+1} = \alpha \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t) P(\mathbf{x}_t|\mathbf{e}_{1:t})$$

Which of the following statements is incorrect?

(a) α is the normalisation factor
(b) $P(\mathbf{x}_t|\mathbf{e}_{1:t})$ is the current state distribution
(c) $\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})$ is obtained from the sensor model
(d) $\mathbf{P}(\mathbf{x}_{t+1})$ depends on both $\mathbf{P}(\mathbf{x}_t)$ and $\mathbf{P}(\mathbf{e}_t)$
(e) $\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t)$ is obtained from the transition model

19. You are given $P(o_1|a_1)$, $P(o_1|a_2)$, $P(o_2|a_1)$, $P(o_2|a_2)$ and utility function u for a single-shot decision problem with actions $\{a_1, a_2\}$ and outcomes $\{o_1, o_2\}$. Which of the following is the correct formula for computing the expected utility of a_1?

(a) $P(o_1|a_1)u(o_1) + P(o_2|a_1)u(o_2)$
(b) $P(o_1|a_1)u(a_1) + P(o_2|a_1)u(a_2)$
(c) $P(a_1|o_1)u(o_1) + P(a_2|o_1)u(o_1) + P(a_1|o_2)u(o_2) + P(a_2|o_2)u(o_2)$
(d) $P(o_1|a_1)u(o_1) + P(o_2|a_1)u(o_2) + P(o_1|a_2)u(o_1) + P(o_2|a_2)u(o_2)$
(e) $P(o_1|a_1)u(o_1) + P(o_2|a_1)u(o_2) - P(o_1|a_2)u(o_1) + P(o_2|a_2)u(o_2)$

20. Suppose an agent is given a choice between lotteries A and B and then a choice between lotteries C and D.

A: 100% chance of gaining £1M
B: 89% chance of gaining £1M; 1% chance of no change; 10% chance of gaining £5M.

C: 89% chance of no change; 11% chance of gaining £1M.
D: 90% chance of no change; 10% chance of gaining £5M.

Which of the following choices is *not* rational?

(a) To choose B over A.
(b) To choose B over A and D over C.
(c) To choose D over C.
(d) To choose A over B and D over C.
(e) You can't tell without knowing $U(£1M)$ and $U(£5M)$.

Part B
ANSWER ONE QUESTION FROM PART B

1. Skyscraper puzzles are Sudoku-like problems, where each square in an $n \times n$ grid has to be filled with an integer from $\{1, 2, \ldots, n\}$ such that each number appears exactly once in every row and column, and each number represents the height of a building located in that square. "Visibility counters" outside the board indicate how many buildings are visible from the respective position, assuming that a taller building occludes lower buildings behind it. These numbers may appear in any location around the board, or be missing from any of those positions. In the following (partially solved) instance of the problem, for example, the 3 on top of the second column from the left indicates that 3 buildings (in this case 1, 2, and 4) can be seen when looking downward from that location:

		3			
	3	1	4	2	
1	4	2			
	1	4			2
	2	3			
			4		

In your answers below, you may use a constraint $VisCount(x_1, \ldots, x_n) = y$ which is true whenever y is the number of visible buildings when looking at a sequence left-to-right, e.g. $VisCount(3, 2, 4, 1) = 2$ and $VisCount(1, 2, 3, 4) = 4$. You may also use the $AllDiff(x_1, \ldots, x_n)$ constraint that requires the values of variables $x_1, \ldots, x_n$ to be distinct.

(a) Define appropriate variables, domains, and constraints to describe arbitrary Skyscraper problems on a 4×4 grid for arbitrary sets of "visibility counters". [*10 marks*]

(b) Using only the operator $<$ for comparisons among numbers, provide a first-order logic definition of the $VisCount$ constraint for $n = 4$. [*5 marks*]

(c) Starting from the partial solution shown in the picture above, solve the problem using backtracking search with appropriate heuristics. [*10 marks*]

2. Consider a knowledge base that contains the following first-order logic description of a map-colouring problem:

$$Neighbour(A, B) \quad Neighbour(A, C) \quad Neighbour(B, D) \quad Neighbour(C, D)$$
$$\forall x \forall y \; Neighbour(x, y) \Rightarrow Neighbour(y, x)$$
$$\forall x \forall y \forall c \; Neighbour(x, y) \land Colour(x, c) \Rightarrow \neg Colour(y, c)$$

Prove that the statement

$$Colour(A, Red) \land Colour(B, Green) \land Colour(C, Blue) \Rightarrow Colour(D, Red)$$

is entailed by the above knowledge base applying first-order resolution, performing the following steps:

(a) Provide a CNF description of the negated query. *[3 marks]*

(b) Convert all non-literal facts in the knowledge base into CNF. *[4 marks]*

(c) Perform a sequence of suitable resolution steps until the empty clause is derived. *[15 marks]*

(d) Explain why the derivation of the empty clause proves that the statement is entailed in a resolution-based proof. *[3 marks]*

Part C
ANSWER ONE QUESTION FROM PART C

1. **Planning**

   Imagine you are building a planner that will solve problems in a simple trucking domain. There are three types of objects: trucks, cities and packages. Trucks are used to transport packages from one city to another. A truck can go directly from any city to any other city, with the action *drive*. Each truck can transport just one package at a time from one city to another. To *load* a package onto a truck, the truck must be empty and the truck and package must both be at the same city. When you *unload* a package from a truck, the truck becomes empty again.

   The initial state is one where you have 2 packages in Athens, 3 trucks in Rome and 1 truck in Paris. There are no other packages or trucks anywhere. Your goal is to get the 2 packages in Athens to Toulouse, and then empty trucks back to where they were.

   (a) Define a vocabulary for the PDDL language for referring to the 3 trucks, 2 packages and cities in the above description, as well predicates for expressing the properties and relations they bear to each other that are relevant to this simple trucking planning domain. [*4 marks*]

   (b) Define the initial and goal states using your language. [*6 marks*]

   (c) Define the actions *drive*, *load* and *unload*, being sure to specify all of their preconditions and effects. [*6 marks*]

   (d) Define a plan for getting from the initial state to the goal state. [*3 marks*]

   (e) Redefine the action descriptions *load* and *unload* so as to remove the restriction that a truck can only transport one package at a time. [*4 marks*]

   (f) Define a plan for getting from the initial state to the goal using the new action descriptions that uses exactly 3 drive actions. [*2 marks*]

2. **Bayesian Inference**

Consider a population of women, 50% of whom are over 40. If they are over 40 years old, then 10% of them have breast cancer. But only 5% of women under 40 have breast cancer. 70% of the women with breast cancer will have a positive mammography (meaning the test indicates she has cancer). 20% of women who do not actually have breast cancer also get a positive mammography (meaning that they are incorrectly diagnosed with cancer).

(a) Model the above description as a Bayes Net, using Boolean variables and conditional probability tables. [*7 marks*]

(b) Using your Bayes Net, calculate the likelihood that a woman under 40 who has had a negative mammography has breast cancer. Be sure to show each step of your calculation. **Note:** Give you answer to 3 decimal places. [*9 marks*]

(c) Do you need to know the prior probability that a woman is under 40 to compute the posterior probability that a woman having breast cancer gets a positive mammography? Explain your answer. [*2 marks*]

(d) Suppose that a woman, who is under 40, gets a positive mammography test, M1, but goes on to have a second mammography, M2, that turns out to be negative. Use the Naive Bayes assumption to compute the likelihood that she has breast cancer. [*7 marks*]

Specimen Answers

Part A

1. a
2. b
3. b
4. c
5. b
6. c
7. d
8. a
9. b
10. e
11. c
12. b
13. a
14. c
15. e
16. a
17. c
18. d
19. a
20. d

Part B

1. (a) As variables and domains one could use $s_{ij} \in \{1,2,3,4\}$ for the numbers in the squares, and (say) n_j, e_i, w_i, s_j with domains $\{1,2,3,4,\perp\}$ ($\perp$ means "not specified") for the visibility counters on the north, east, west, and south side of the board, respectively, where $1 \leq i,j \leq 4$ (3 marks). To express the uniqueness of each number in every row and column, we need four *AllDiff* constraints for each column $s_{1j},\ldots,s_{4j}$ and another four for each row $s_{i1},\ldots,s_{i4}$ (3 marks). For every $n_j/e_i/w_i/s_j$ that has a value other than $\perp$, we need the following constraints (4 marks):

$$\begin{aligned} n_j &= \mathit{VisCount}(s_{1j},\ldots,s_{4j}) \\ e_i &= \mathit{VisCount}(s_{i4},\ldots,s_{i1}) \\ w_i &= \mathit{VisCount}(s_{i1},\ldots,s_{i4}) \\ s_j &= \mathit{VisCount}(s_{1j},\ldots,s_{4j}) \end{aligned}$$

   (b) The following is a definition of the constraint for the general case:

$$\begin{aligned} \forall a,b,c,d,k\, \mathit{VisCount}(a,b,c,d) = k :\Leftrightarrow &(k = 1 \wedge a > b \wedge a > c \wedge a > d) \vee \\ &(k = 2 \wedge a < b \wedge b > c \wedge b > d) \vee \\ &(k = 3 \wedge a < b \wedge b < c \wedge c > d) \vee \\ &(k = 4 \wedge a < b < c < d) \end{aligned}$$

   One mark for overall idea, one mark for each sub-case.

   (c) s_{23}, s_{24}, s_{33}, s_{34}, s_{43} and s_{44} are the remaining variables. Many different search strategies are possible, here is one possible solution:

      i. All variables are equally constrained (two values left for each). The most constraining variables are s_{23} and s_{33} (two *VisCount* constraints each). Pick $s_{33} \in \{2,3\}$, and none of the two values is less constraining than the other, so we can assign $s_{33} = 2$.
      ii. Among the remaining variables, the most constrained one is now $s_{34} = 3$, which we can assign directly to its only remaining value.
      iii. Next, the tie-breaker for most constraining variable is s_{23}, assign $s_{23} = 3$.
      iv. This leaves $s_{24} = 1$ as the only possible legal assignment for the subsequently most constrained variable.
      v. Now the most constrained variable is s_{43}, we assign $s_{43} = 1$.
      vi. This leaves $s_{44} = 4$, which solves the puzzle.

   Two marks for correct overall understanding of algorithm, four for applying heuristics correctly in each situation, two for reasoned explication of the procedure, two for correct solution.

2. (a) **NOTE: The solution to this question involves an error when the initial CNF of $\forall x \forall y \forall c\ Neighbour(x,y) \land Colour(x,c) \Rightarrow \neg Colour(y,c)$ is computed. If this is correctly converted to CNF, the target statement can*not* be proven from the premises, this would require additionally assuming that every country has to have a colour.**

The negated query in CNF is obtained as follows:

$$\neg(Colour(A,Red) \land Colour(B,Green) \land Colour(C,Blue) \Rightarrow Colour(D,Red)) \equiv$$
$$\neg(\neg(Colour(A,Red) \land Colour(B,Green) \land Colour(C,Blue)) \lor Colour(D,Red)) \equiv$$
$$\neg(\neg Colour(A,Red) \lor \neg Colour(B,Green) \lor \neg Colour(C,Blue) \lor Colour(D,Red)) \equiv$$
$$Colour(A,Red) \land Colour(B,Green) \land Colour(C,Blue) \land \neg Colour(D,Red)$$

One mark every line of correct transformation.

(b) In the following list we repeat literal facts in order to use a consistent numbering for the resolution proof below:

(1) $Neighbour(A,B)$ (2) $Neighbour(A,C)$ (3) $Neighbour(B,D)$ (4) $Neighbour(C,D)$
(5) $\neg Neighbour(a,b) \lor Neighbour(b,a)$
(6) $\neg Neighbour(x,y) \lor \neg Colour(x,c) \lor Colour(y,c)$

Two marks for (5), two for (6).

(c) We refer to the query clauses as (Q) in the following proof:

i. (Q)+(6): $\neg Neighbour(C,y) \lor Colour(y,Blue)$ (7)
ii. (Q)+(6): $\neg Neighbour(B,y) \lor Colour(y,Green)$ (8)
iii. (7)+(4): $Colour(D,Blue)$ (9)
iv. (8)+(3): $Colour(D,Green)$ (10)
v. (7)+(9): $\neg Neighbour(C,D)$ (11)
vi. (8)+(10): $\neg Neighbour(B,D)$ (12)
vii. (3)+(11) or (4)+(12) yields empty clause

Four marks for each of the key steps (technically one only needs the derivation for either C or B), three for overall structure.

(d) If we derive the empty clause for the negated query, this means that it is impossible for the knowledge base to hold, and the query not to be entailed.

Part C

1. **Planning**

   (a) There are several options for describing the vocabulary, but here is one:

   - Variables $x, y, z \ldots$
   - Constants: T_1, T_2, T_3, *Athens*, *Toulouse*, *Paris*, *Rome*
   - $Truck(x)$: x is a truck
   - $Package(x)$: x is a package
   - $Empty(x)$: x is empty
   - $At(x, y)$: x is at a city y
   - $In(x, y)$: x is in y

   Deduct 1 point for each missing part of the vocabulary and for each mistake about the arity of the predicates. Don't deduct anything if they don't include the variables in the list.

   (b) Initial state is:

   $$\begin{aligned}&Truck(T_1) \wedge Truck(T_2) \wedge Truck(T_3) \wedge Truck(T_4) \wedge \\ &Package(P_1) \wedge Package(P_2) \wedge \\ &City(Athens) \wedge City(Toulouse) \wedge City(Paris) \wedge City(Rome) \wedge \\ &Empty(T_1) \wedge Empty(T_2) \wedge Empty(T_3) \wedge Empty(T_4) \wedge \\ &At(T_1, Rome) \wedge At(T_2, Rome) \wedge At(T_3, Rome) \wedge At(T_4, Paris) \wedge \\ &At(P_1, Athens) \wedge At(P_2, Athens)\end{aligned}$$

   Goal state is:

   $$\begin{aligned}&At(P_1, Toulouse) \wedge At(P_2, Toulouse) \wedge \\ &At(T_1, Rome) \wedge At(T_2, Rome) \wedge At(T_3, Rome) \wedge At(T_4, Paris)\end{aligned}$$

   Deduct 0.5 point for each error, including missing information or unnecessary information. Deduct 1 point if the states are syntactically ill-formed, e.g., through using negation.

(c) Action descriptions are:

$$\begin{array}{l} Action(drive(x,y,z), \\ \quad \textsc{Precond:} Truck(x), City(y), City(z), y \neq z, At(x,y) \\ \quad \textsc{Effect:} \neg At(x,y), At(x,z), (\textbf{when } In(p,x) : \neg At(p,y), At(p,z))) \end{array}$$

$$\begin{array}{l} Action(load(x,y), \\ \quad \textsc{Precond:} Package(x), Truck(y), Empty(y) \\ \quad \textsc{Effect:} In(x,y)) \end{array}$$

$$\begin{array}{l} Action(unload(x,y), \\ \quad \textsc{Precond:} Package(x), Truck(y), in(x,y) \\ \quad \textsc{Effect:} \neg In(x,y), Empty(x)) \end{array}$$

Deduct 0.5 points for each missing or incorrect precondition or effect. Deduct 1 point for missing the "when" clause.

(d) There are many plans that work, but here is one:

$$\begin{array}{ll} [& drive(T_1, Rome, Athens); drive(T_2, Rome, Athens), load(P_1, T_1), load(P_2, T_2); \\ & drive(T_1, Athens, Toulouse); drive(T_2, Athens, Toulouse); \\ & unload(P_1, T_1); unload(P_2, T_2); \\ & drive(T_1, Toulouse, Rome); drive(T_2, Toulouse, Rome)] \end{array}$$

Deduct 1 point for each mistake, including not getting the trucks back to their destination!

(e)

$$\begin{array}{l} Action(load(x,y), \\ \quad \textsc{Precond:} Package(x), Truck(y) \\ \quad \textsc{Effect:} In(x,y)) \end{array}$$

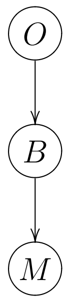
$$\begin{array}{l} Action(unload(x,y), \\ \quad \textsc{Precond:} Package(x), Truck(y), In(x,y) \\ \quad \textsc{Effect:} \neg In(x,y)) \end{array}$$

Deduct 0.5 points for each missing or incorrect precondition or effect.

(f)

$$[\quad drive(T_1, Rome, Athens); load(P_1, T_1); load(P_2, T_1);$$
$$drive(T_1, Athens, Toulouse); unload(P_1, T_1); unload(P_2, T_2);$$
$$drive(T_1, Toulouse, Rome)]$$

Deduct 0.5 point for each incorrect part of the plan. Deduct 1 point for not using 1 truck (which is the only way of doing the plan with 3 drive actions.



$P(o)$
0.5

O	$P(B)$
o	0.03
$\neg o$	0.01

B	$P(m)$
b	0.7
$\neg b$	0.2

Figure 1: The Bayes Net for the Breast Cancer Problem

2. **Bayesian Inference**

(a) The Boolean variables are: O (o if a woman is over 40); B (b if a woman has breast cancer); M (m if the mammography test came out positive). The Bayes Net should be as shown in Figure 1.

Deduct 1 point for not having the right variables; 2 points for each incorrect dependency, and 2 points for each mistake in the CPTs.

(b)

$$\begin{aligned}
\vec{P}(B|\neg m, \neg o) &= \alpha P(\neg o)\vec{P}(B|\neg o)\vec{P}(\neg m|B) \\
P(b|m, o) &= \alpha P(\neg o)P(b|\neg o)P(\neg m|b) \\
&= \alpha * 0.5 * 0.05 * 0.3 \\
&= \alpha * 0.008 \\
P(\neg b|\neg m, \neg o) &= \alpha P(\neg o)P(\neg b|\neg o)P(\neg m|\neg b) \\
&= \alpha 0.5 * 0.95 * 0.8 \\
&= \alpha * 0.380 \\
\vec{P}(B|\neg m, \neg o) &= \alpha\langle 0.008, 0.380\rangle \\
&= \langle 0.021, 0.979\rangle \\
P(b|\neg m, \neg o) &= 0.021
\end{aligned}$$

Deduct 2 points for multiplying the wrong probabilities; deduct 1 point for each wrong table lookup, and deduct 1 point for each arithmetic error but don't penalise for the way those errors percolate through the calculations.

(c) No, you don't need to know this, because M is conditionally independent of O given B (as shown in the Bayes Net).

Give 1 point for getting the right answer, 1 point for saying "conditional independence".

(d)

$$\begin{aligned}
P(b|m_1, \neg m_2, \neg o) &= \alpha P(\neg o)P(b|\neg o)P(m|b)P(\neg m|b) \text{ by Naive Bayes} \\
&= \alpha 0.5 * 0.05 * 0.7 * 0.3 \\
&= \alpha * 0.005 \\
P(\neg b|m_1, \neg m_2, \neg o) &= \alpha P(\neg o)P(\neg b|\neg o)P(m|\neg b)P(\neg m|\neg b) \\
&= \alpha * 0.5 * 0.95 * 0.2 * 0.8 \\
&= \alpha * 0.076 \\
P(b|m_1, m_2, \neg 0) &= \tfrac{0.005}{0.005+0.076} \\
&= 0.062
\end{aligned}$$

Give 3 points for getting the formula right, and the remaining 4 points for getting it all right. If they make a mistake, but note something must be wrong when compared with the probability in part (b) and the likelihoodsl given by the CPTs, then give them 1 bonus point for knowing it's wrong!