Linköpings Universitet
Institutionen för Datavetenskap
Patrick Doherty

Tentamen
TDDC17 Artificial Intelligence
24 August 2020 kl. 08-12

Points:

The exam consists of exercises worth xx points.
To pass the exam you need xx points.

Auxiliary help items:

Course Textbook
Course Slides
Hand calculators.
You may not use any additional content on the WWW/Internet, other external information, etc. You may not use other books.

Directions:

You can answer the questions in English or Swedish.
Use notations and methods that have been discussed in the course.

Make reasonable assumptions when an exercise has been under-specified.
State these assumptions explicitly in your answer.
Begin each exercise on a new page.
Write only on one side of the paper.
Write clearly and concisely.

Jourhavande: Mariusz Wzorek, 0703887122.
Mariusz will be available in Zoom via this link at 10.00: (add link)

This exam was given as a distance exam and is formatted differently in LISAM.

1. The following questions pertain to Bayesian Networks.

   Consider the following random variables: O (oil), I (inflation), E (economy health), B (BP oil stock price), and R (retailer stock price). O, I, E and R have the domain $\{\mathsf{high}, \mathsf{low}\}$ and B has the domain $\{\mathsf{high}, \mathsf{low}, \mathsf{normal}\}$. The Bayesian network model for these variables has the following conditional table entries (the rest can be derived easily):

   Table 1: BN table entries

$P(E = \mathsf{low}) = 0.2$	
$P(B = \mathsf{low} \mid O = \mathsf{low}) = 0.9$	$P(B = \mathsf{normal} \mid O = \mathsf{low}) = 0.1$
$P(B = \mathsf{low} \mid O = \mathsf{high}) = 0.1$	$P(B = \mathsf{normal} \mid O = \mathsf{high}) = 0.4$
$P(O = \mathsf{low} \mid E = \mathsf{low}) = 0.9$	$P(O = \mathsf{low} \mid E = \mathsf{high}) = 0.05$
$P(R = \mathsf{low} \mid I = \mathsf{low}, E = \mathsf{low}) = 0.9$	$P(R = \mathsf{low} \mid I = \mathsf{low}, E = \mathsf{high}) = 0.1$
$P(R = \mathsf{low} \mid I = \mathsf{high}, E = \mathsf{low}) = 0.1$	$P(R = \mathsf{low} \mid I = \mathsf{high}, E = \mathsf{high}) = 0.01$
$P(I = \mathsf{low} \mid O = \mathsf{low}, E = \mathsf{low}) = 0.9$	$P(I = \mathsf{low} \mid O = \mathsf{low}, E = \mathsf{high}) = 0.1$
$P(I = \mathsf{low} \mid O = \mathsf{high}, E = \mathsf{low}) = 0.1$	$P(I = \mathsf{low} \mid O = \mathsf{high}, E = \mathsf{high}) = 0.01$

   (a) Draw a Bayesian network for the joint distribution $\mathbf{P}(O, I, E, B, R)$ constrained by the table information above. **[2p]**

   (b) Provide a definition of the joint distribution $\mathbf{P}(O, I, E, B, R)$ in terms of a product of (conditional) probability distributions based on the Bayesian network. **[1p]**

   (c) Given that the BP oil stock price is normal and the retailer stock price is high, what is the probability that inflation is high? Please provide the steps in your computation. Partial credit will be given for setting up the problem correctly. No credit will be given for just a numerical answer. **[2p]**

2. The following questions pertain to Answer Set Programming.

   (a) Given the positive answer set program Π_1, consisting of the following rules:

      r1: $\mathsf{p}(1)$.

      r2: $\mathsf{p}(2)$.

      r3: $\mathsf{p}(3)$.

      r4: $\mathsf{q}(3) \leftarrow \textit{not}\ \mathsf{r}(3)$.

      r5: $\mathsf{r}(1) \leftarrow\ \mathsf{p}(1), \textit{not}\ \mathsf{q}(1)$.

      r6: $\mathsf{r}(2) \leftarrow \mathsf{p}(2), \textit{not}\ \mathsf{q}(2)$.

      r7: $\mathsf{r}(3) \leftarrow \mathsf{p}(3), \textit{not}\ \mathsf{q}(3)$.

      i. What are the number of *possible* answer sets for Π_1? Explain why. **[1p]**

      ii. Given the following possible answer sets:

         A. $S_1 = \{\mathsf{r}(2), \mathsf{r}(1), \mathsf{p}(1), \mathsf{p}(2), \mathsf{p}(3), \mathsf{r}(3)\}$

         B. $S_2 = \{\mathsf{r}(1), \mathsf{p}(1), \mathsf{p}(2), \mathsf{p}(3), \mathsf{r}(3)\}$

         C. $S_3 = \{\mathsf{r}(2), \mathsf{r}(1), \mathsf{p}(1), \mathsf{p}(2), \mathsf{p}(3), q(3)\}$

      iii. For each possible answer set S_i, provide the reduct, $\Pi_1^{S_i}$ for Π_1. **[1.5p]**

      iv. Generate $Cn(\Pi_1^{S_i})$ for each reduct $\Pi_1^{S_i}$ of Π_1. **[1.5p]**

      v. Which of the three answer sets S_1, S_2, S_3 are *actual* answer sets for Π_1 (Explain why)? **[1p]**

   (b) The consequence relation, $\models$, for classical logic is said to be monotonic. What does this mean? (Be precise and succinct in your explanation) **[1p]**

   (c) Why is Answer Set Programming considered to be a nonmonotonic reasoning formalism (Be precise and use an example)? **[1p]**

3. The following questions pertain to Constraint Satisfaction Problems (CSP).

   (a) Suppose one is given a constraint satisfaction problem with two variables X and Y, whose domains are $D_X = \{1, 2, 3\}$, $D_Y = \{1, 2\}$ and with a constraint $X < Y$.

      i. Is the arc $X \rightarrow Y$ arc consistent? Explain why or why not. **[1p]**

      ii. Is the arc $Y \rightarrow X$ arc consistent? Explain why or why not. **[1p]**

   (b) Suppose one is given a constraint satisfaction problem CSP1 with two variables X and Y, whose domains are $D_X = \{1, 2, 3\}$, $D_Y = \{1, 2, 3\}$ and with a constraint $X < Y$. What happens to D_X and D_Y after the REVISE() function in the AC-3 arc-consistency algorithm is called: REVISE(CSP1, X,Y) ? **[1p]**

   (c) The 4×4 matrix in Figure 1 depicts a simplified version of Sudoku. The goal is to fill in each cell in the matrix with a number between 1 and 4 in such a way that no number is repeated on the same column or the same row. The matrix has already been filled in partially. The remaining cells have been named with a letter for easy reference. These letters are the variables in the constraint satisfaction problem.

2	A	3	B
4	C	1	2
1	D	E	F
3	G	4	1

Figure 1: Sudoku Board

      i. Specify the constraint problem by providing the Variables, Domains, and Constraints. **[1p]**

      ii. Fill in the table below (some values are already provided as examples). For instance A has two remaining values and it has constraints with four other variables, B, C, D, G. **[1p]** (Write your own complete table in the answer page. Do not fill in this table on the exam page.):

Variable	A	B	C	D	E	F	G
Remaining Values	1,4			2,3,4			
# of constraints with other variable	Four (B,C,D,G)			Five (A,C,G,E,F)			

Figure 2: MRV Table

      iii. Based on the table in Figure 2 and using the Minimum Remaining Values (MRV) heuristic, list the variable that the CSP search algorithm will select next. If there are ties, list all the variables that have the same MRV. **[1p]**

      iv. If the above was a tie, use the degree heuristic (i.e., variable with the most constraints on remaining variables) to break the tie. What variable would be selected? If a tie still remains, provide a systematic way to deal with the tie so that only one variable is selected. Explain your work. **[1p]**

(d) Starting from the following possible values in the table in Figure 3, use forward checking to propagate constraints. Show the propagation of just one constraint at a time neatly on a separate row in the table below, until no more constraints can be propagated. An example is provided on the 3rd row. The "?"'s in the constraint propagation column should also be filled in correctly. (Write your own complete table in the answer page. Do not fill in this table on the exam page.): **[2p]**

Assignment	Constraint Propagation	A	B	C	D	E	F	G
	Possible Values	1,4	4	3	2,3,4	2	3,4	2
B = 4	Constraint between **A** and **B**	1	4	3	2,3,4	2	3,4	2
–	Constraint between **?** and **?**							
E=2	Constraint between **?** and **?**							
F = 3	Constraint between **?** and **?**							

Figure 3: MRV Table

4. The following questions pertain to search.

   (a) Which of the following are true and which are false? Explain your answers.

      i. Depth-first search always expands at least as many nodes as A^* search with an admissible heuristic. **[1p]**

      ii. $h(n) = 0$ is an admissible heuristic for the 8-puzzle. **[1p]**

      iii. Breadth-first search is complete even if zero step costs are allowed. **[1p]**

   (b) Consider a state space where the start state is number 1 and each state has two successors: numbers $2k$ and $2k + 1$.

      i. Draw the portion of the state space for states 1 to 15. **[1p]**

      ii. Suppose the goal state is 11. List the order in which the nodes will be visited for breadth-first search, depth-first search, depth-limited search with limit 3, and iterative deepening search. **[2p]**

   (c) What's the difference between a world state, a state description, and a search node? Why is this distinction useful? **[2p]**

5. The following questions pertain to the extended Davis-Putnam algorithm (DPLL) considered in the course book. DPLL takes as input a formula in propositional logic, transforms it to its equivalent conjunctive normal form (CNF) representation and determines whether the formula is satisfiable or not. The questions pertain to the different heuristics used in DPLL.

   (When asked to "complete the table ..." in the questions below, write your own complete table in the answer page. Do not fill in these tables on the exam page.)

   (a) Complete the table below by applying the Splitting rule to variable s and CNF: **[1p]**

   $$\alpha = (P \vee Q \vee R \vee S) \wedge (\neg P \vee Q \vee \neg R) \wedge (\neg Q \vee \neg R \vee S) \wedge (P \vee \neg Q \vee R \vee S)$$
   $$\wedge (Q \vee \neg R \vee \neg S) \wedge (\neg P \vee \neg R \vee S) \wedge (\neg P \vee \neg S) \wedge (P \vee \neg Q)$$

   Table 2:

Heuristic	Model_1	Simplified CNF_1	Model_2	Simplified CNF_2
Initial call	$\{\}$	α	-	-
$SR[S]$	$\{S : True\}$		$\{s : False\}$	

   (b) Complete the table below by applying the Unit Clause Heuristic (UCH) as long as possible to the CNF: **[1.5p]**

   $$\alpha = (\neg P \vee Q \vee \neg R) \wedge (Q \vee \neg R) \wedge (\neg P) \wedge (P \vee \neg Q)$$

   Use one row for each call and state the variable the heuristic is being applied to in the 1st column. Add as many additional rows as required.

   Table 3:

Heuristic	Model	Simplified CNF
Initial call	$\{\}$	α
$UCH[-]$		
$UCH[-]$		
$UCH[-]$		

   (c) Complete the table below by applying the Pure Symbol Heuristic (PSH) as long as possible to the CNF: **[1.5p]**

   $$\alpha = (\neg P \vee Q) \wedge (R \vee \neg Q) \wedge (S \vee \neg U) \wedge (U \vee \neg S)$$

   Use one row for each call and state the variable the heuristic is being applied to in the 1st column. Add as many additional rows as required.

   Table 4:

Heuristic	Model	Simplified CNF
Initial call	$\{\}$	α
$PSH[-]$		
$PSH[-]$		
$PSH[-]$		

   (d) Suppose we had three formulas in propositional logic, $F1, F2$ and $F3$. Explain how one would show whether $F1 \wedge F2 \models F3$ using the DPLL algorithm. **[2p]**

6. The following question pertains to automated planning. **[4p]**

   We will use a typed planning problem instance (with true object types) defined as follows:

   - Objects:
     - Type location: { HWS, SM, Home, Work }
     - Type object: { Drill, Milk, Bananas }
   - Operators:
     - Go(?from - location, ?to - location)
       * Precond: At(?from)
       * Effects: At(?to), not At(?from)
     - Buy(?product - object, ?store - location)
       * Precond: At(?store), Sells(?store, ?product)
       * Effects: Have(?product)
   - Initial state: { At(Home), Sells(HWS, Drill), Sells(SM, Milk), Sells(SM, Bananas) }
   - Goal: { At(Home), Have(Drill), Have(Milk), Have(Bananas) }

   Assume we are doing *partial order planning*, also known as *plan-space search* because we are searching in the space of *partially ordered plans*. Every node in this search space, including the one illustrated below, corresponds directly to a partially ordered plan that has a set of *flaws* that can be *resolved*.

   A partial order plan for this problem is shown in Figure 4. The notation is the same as in the course slides, though it is turned on its side, with preconditions above and effects below. As usual, a thin red dotted arrow indicates a threat and is not actually part of the plan structure.

   Your first task is to find and list *all flaws* in this plan.

   Your second task is to show, for each of those flaws, *all alternative ways in which the flaw can be resolved*. To do this, you *describe in plain text* how each alternative resolution would modify the illustrated search node. That is, exactly which changes could be made to the already illustrated search node in order to resolve the flaw in a particular way, in terms of precedence constraints, causal links and actions?

   It must be very clear from your description exactly which successors, and how many, there are. For example, you can say that "For each ..., there is one successor created by adding / removing / changing ..., resulting in 5 successors. Then, for each ..., there is ..., resulting in ...". This is just an example – what is important is that you clearly describe exactly which successors exist.

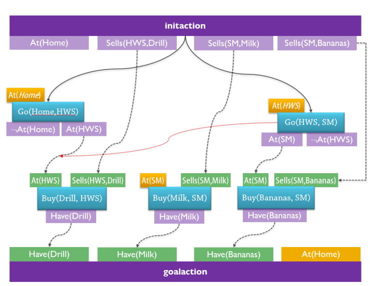

Figure 4:

7. The following questions pertain to machine learning. Give *short and informative* answers.

   (a) Assume you want to train a linear model $y = h_w(x) = w_1 x + w_0$ from examples (x_i, y_i). You are given the training data $\{(4,9),(1,1),(3,7)\}$. What would a suitable loss function $\mathcal{L}(w)$ be for this regression problem (here parameterized by the weight parameter vector $w = [w_1, w_0]$)? **[2p]**

   (b) Out of two alternative assignments of values to the parameter vector, $w = [1,2]$ or $w = [2,0]$, calculate which gives the best fit to the training data. **[2p]**

   (c) Assuming you have already calculated $\nabla_w \mathcal{L}(w)$ (treat it as known), outline an iterative algorithm for automatically finding the optimal fit to the training data in this case. **[2p]**

8. The following questions pertain to neural networks and deep learning. Give *short and informative* answers.

   (a) Explain (short paragraph) the purpose of *early stopping*, and how it is done. **[2p]**

   (b) Assume you want to do classification with a regular fully-connected (dense) neural network with 35 ReLU neurons in both the first and second layers. If an example input x to the network is a vector of size 20, calculate how many *parameters* (weights) are needed in each of the first and second layer respectively. **[2p]**