Lunds Tekniska Högskola
Institutionen för datavetenskap

EDA132
VT 2013

Tillämpad Artificiell Intelligens
Applied Artificial Intelligence
Tentamen 2013-03–11, 08.00–13.00, Vic 3:C

You can give your answers in Swedish or English.
You are welcome to use a combination of figures and text in your answers.
8 questions, 65 points, 50% needed for pass.

1. **Search:** (10 points) Consider the search space shown on the next page, where S is the start node and G1 and G2 satisfy the goal test. Arcs are labeled with the cost of traversing them and the estimated cost to a goal is reported inside nodes (so lower scores are better). For each of the following search strategies, indicate which goal state is reached (if any) and list, in order, all the states popped off of the OPEN (aka FRINGE) list. When all else is equal, nodes should be removed from OPEN in alphabetical order.

   You may use the form below or copy its structure on an answer sheet.

   Breadth-First Goal state reached: ______
   States popped off OPEN:____

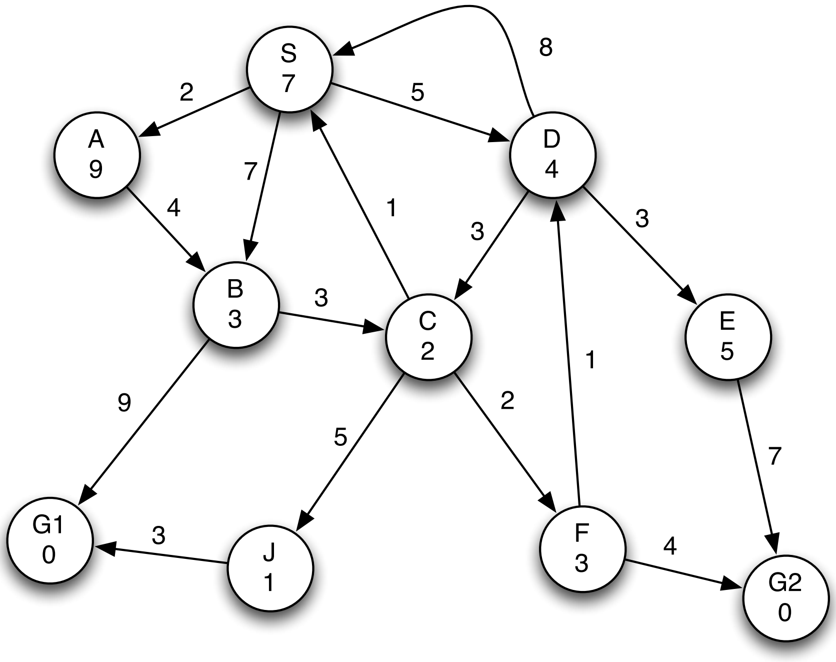
   Best-First (Greedy) using $f = h$
   Goal state reached: ______
   States popped off OPEN:____

   Iterative Deepening Goal state reached: ______
   States popped off OPEN:____

   A* Goal state reached: ______
   States popped off OPEN:____

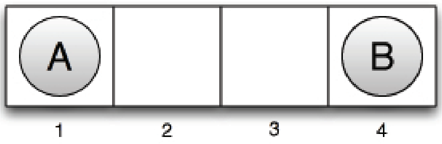
   Now imagine that you wish to run a Genetic Algorithm on this search space. You use four bits to represent nodes: A = 0001, B = 0010, C =0011, D = 0100, E = 0101, F =0110, G1 =1001, G2 = 1110, J = 0111, and S = 0000.

   - If nodes B and E are chosen for the cross-over operator, show two possible children that can be produced.
   - With 4 bits one can represent 16 distinct nodes, but only 10 are in this task's search problem. What do you think should be done when a bit string is generated that matches none of the nodes?



End of question 1.

2. **Games:** Consider a two-player game featuring a board with four locations, numbered 1 through 4 and arranged in a line. Each player has a single token. Player A starts with his token on space 1, and player B starts with his token on space 4. Player A moves first.



The two players take turns moving, and each player must move his token to an open adjacent space *in either direction.* If the opponent occupies an adjacent space, then a player may jump over the opponent to the next open space if any. (For example, if A is on 3 and B is on 2, then A may move back to 1.) The game ends when one player reaches the opposite end of the board. If player A reaches space 4 first, then the value of the game is $+1$; if player B reaches space 1 first, then the value of the game is -1.

(a) (3 points) Draw the complete game tree, using the following conventions:

- Write each state as (s_A, s_B) where s_A and s_B denote the token locations.
- Put the terminal states in square boxes, and annotate each with its game value in a circle.
- Put *loop states* (states that already appear on the path to the root) in double square boxes. Since it is not clear how to assign values to loop states, annotate each with a "?" in a circle.

(b) (2 points) Now mark each node with its backed-up minimax value (also in a circle). Explain in words how you handled the "?" values, and why.

(c) (3 points) Explain why the standard minimax algorithm would fail on this game tree and briefly sketch how you might fix it, drawing on your answer to 2b. Does your modified algorithm give optimal decisions for all games with loops?

End of question 2.

3. **Reasoning:**

   (a) Describe the *modus ponens* inference rule. Give a concrete example of how it works. (2 points)

   (b) What does it mean that resolution is a sound inference rule? (1 point)

   (c) Explain the terms *forward chaining* and *backward chaining* in the context of logical reasoning. (2 points)

   (d) Explain what SPARQL is and how is it related to reasoning. (2 points)

End of question 3.

4. **Planning:**

   A simple computer has a bunch of memory cells M and some registers R. Two computer instructions could be LOAD(M,R) (copy contents of M into R, overwriting what's there) and ADD(M,R) (add contents of M to contents of R, leaving result in R.)

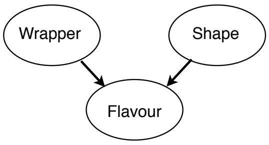
   (a) (2 points) Express these instructions as STRIPS operators.

   (b) (2 points) With several registers, you could imagine a compiler being inefficient with register usage (overwriting partial results that could be reused, for instance). Relate what you know or can imagine about code optimization to what you know about planning.

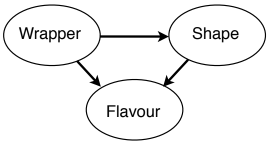
   (c) (2 points) Would the planning graph representation be useful for this problem? Motivate how or why not?

End of question 4.

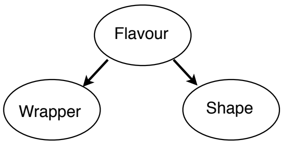
5. **Probabilistic reasoning:** (8 points)



(i)



(ii)



(iii)

The Sweet Guess Company produces sweets in two flavours, "Cherry" and "Mustard". 70% of the produced pieces have cherry flavour, the rest (30%) have mustard flavour. In the production line all sweets are initially more or less sphere shaped, but further on a certain percentage of them are picked randomly to be trimmed into cubes. After that, the sweets are wrapped in either red or yellow paper, also this randomly chosen. In the end, 80% of the cherry flavoured sweets are round (spheres), and 80% have a red wrapper, while 90% of the mustard flavoured pieces are cubes and 90% have yellow wrappers. Each piece is then put into an individual, sealed, black box.
Your friend surprises you with a Sweet Guess – now you are sitting there, wondering what you will get when you open the box. Consider the three networks (i), (ii), and (iii) in Figure above.

a) Which network(s) can *correctly* represent $\mathbf{P}(Flavour, Wrapper, Shape)$? In case there are several, which one of them is the *best* choice? Explain your answer!

b) What is the probability that your Sweet Guess has a red wrapper?

c) You find a sphere shaped sweet wrapped in red paper. What is (approximately) the probability p that your sweet has cherry flavour? Motivate your answer!
1) $p \leq 0.7$ 2) $0.7 \leq p \leq 0.99$ 3) $p > 0.99$

End of question 5.

6. **Logistic Regression:** (8 points)

(a) Automatic learning can be roughly divided into two categories: unsupervised and supervised learning:

- Describe what supervised learning is and give an example of a data set that would correspond to supervised learning. You can invent such a very small data set for the purpose of your answer.
- Describe what unsupervised learning is and give, or invent, a similar example.

(b) Logistic regression is a widely used technique to classify data in a supervised context. It uses the logistic curve:

- Give the equation of the logistic curve and describe its original purpose.
- Draw the logistic curve and describe it qualitatively.

(c) In its most basic classification use, logistic regression applies to binary data. We will use the data set in Table 1 that reports how many individuals die when given a certain dosage of drug. The first row means that when ten individuals are given the dosage of 40, eight survive and two die.

Table 1: A data set. Adapted and simplified from the original article that described how to apply logistic regression to classification by Joseph Berkson, Application of the Logistic Function to Bio-Assay. *Journal of the American Statistical Association* (1944).

Drug concentration	Survive (class 0)	Die (class 1)
40	8	2
60	6	4
80	4	6
100	3	7
120	2	8
140	1	9
160	1	9
180	1	9
200	1	9
250	0	10
300	0	10

- Draw the examples in a 2-dimensional space. You will use the x axis for the numerical parameter (the dosage) and the y axis to represent the class, 0 or 1, respectively survive and die. Use small dots to represent each individuals.

- Explain how logistic regression can be used in relation with probabilities.
- Fit manually the curve to your data. This does not need to be really accurate, but for each row in Table 1, you will draw a point where the curve will pass through.
- Describe with an example, for instance the dosage of 90, how logistic regression is used in a classification context.

(d) The regression process fits a model to the logistic curve:

- Describe what is modeled in the fitting process. Your answer can either follow the slides or the textbook.
- The regression algorithm can use the gradient ascent (or descent if you follow the textbook). Describe what the gradient is.
- Describe qualitatively the relation between the gradient and the iteration step of the fitting process.

End of question 6.

7. **Decision trees:** (8 points)

(a) ID3 is a machine-learning algorithm induces decision trees from data sets:

Table 2: A data set. After Quinlan (1986)

Object	**Attributes**				**Class**
	Outlook	**Temperature**	**Humidity**	**Windy**	
1	Sunny	Hot	High	False	*N*
2	Sunny	Hot	High	True	*N*
3	Overcast	Hot	High	False	*P*
4	Rain	Mild	High	False	*P*
5	Rain	Cool	Normal	False	*P*
6	Rain	Cool	Normal	True	*N*
7	Overcast	Cool	Normal	True	*P*
8	Sunny	Mild	High	False	*N*
9	Sunny	Cool	Normal	False	*P*
10	Rain	Mild	Normal	False	*P*
11	Sunny	Mild	Normal	True	*P*
12	Overcast	Mild	High	True	*P*
13	Overcast	Hot	Normal	False	*P*
14	Rain	Mild	High	True	*N*

- Using the data set in Table 2, draw **three** different decision trees that separate the positive and negative classes.
- Out of your three decision trees, discuss which one is the best and what could make a decision tree optimal.

(b) ID3 computes the difference of entropies before and after an argument is applied to split a set. This difference is called the information gain:

- Give the definition of entropy.
- Give the definition of the information gain.

(c) We can evaluate the efficiency of an argument in a decision tree using a significance test:

- What is the null hypothesis? Use a set of 20 examples, 10 positives and 10 negatives, and draw a decision tree with one argument with 4 values, that exemplifies this null hypothesis.
- Give a second example of a classification with the same set up, where this time the argument is relevant and classifies the data.

- Give a formula to measure the deviation from the null hypothesis (the bias) and compute it for the two examples you provided.

End of question 7.

8. **Language Technology:** (10 points)

   (a) Describe what part-of-speech tagging is and tag the sentence *That round table might collapse* with parts of speech you learned at school.

   (b) Describe how to model automatic part-of-speech tagging using the noisy channel.

   (c) The model consists of a product of two terms: the probability of a part-of-speech sequence and the word sequence given a part-of-speech sequence.

   - Write the formula describing the probability of a part-of-speech sequence.
     Approximate this probability using bigrams.
   - Write the probability a word sequence given a part-of-speech sequence.
     Write a sensible approximation of this sequence.

   (d) In the next steps, you will tag the sentence *That round table might collapse* using the model above.

   i. Using a part-of-speech annotated corpus, your instructor extracted all the pairs (word, part of speech) and their frequencies:

```
438 That DT
  5 That IN
  3 That WDT

  5 round JJ
 23 round NN
  3 round VB
  1 round VBP

 35 table NN

328 might MD
  4 might NN

 57 collapse NN
  1 collapse NNP
  5 collapse VB
```

   where DT is the code for determiner, IN, for preposition, WDT, for relative pronoun, JJ, for adjective, NN, for noun, NNP, for proper noun, VB, for verb infinitive, and VBP, for verb present, non third person.

Table 3: Frequencies of parts of speech.

Freq.	POS	Freq.	POS	Freq.	POS	Freq.	POS	Freq.	POS
39279	BOS	98083	IN	9634	MD	25816	NNP	12433	VBP
84349	DT	51594	JJ	144992	NN	89817	VB	4555	WDT

Table 4: Frequencies of part-of-speech bigrams.

Freq.	POS bigram	Freq.	POS bigram	Freq.	POS bigram	Freq.	POS bigram
8881	BOS DT	4968	BOS IN	22	BOS WDT		
15141	DT JJ	7284	IN JJ	43	WDT JJ		
40210	DT NN	11797	IN NN	92	WDT NN		
21	DT VB	40	IN VB	7	WDT VB		
194	DT VBP	24	IN VBP	603	WDT VBP		
22518	JJ NN	19807	NN NN	1763	VB NN	92	WDT NN
2364	NN MD						
7	MD NN						
6	MD NNP	2717	NN NNP				
7680	MD VB	187	NN VB				

Annotate the words of the sentence with their most frequent part of speech and give the accuracy of this method.

ii. To improve the accuracy, you will use bigrams and the Viterbi algorithm. The Viterbi algorithm is a dynamic programming technique that uses a table. Given the statistics we extracted from the corpus, fill Table 5 with zeros when the value in the table is a 0 and a cross, when the value is a nonzero real number.

iii. Write the formulas to compute the values of the nonzero cells of the 3rd column, corresponding to *That*. There are three cells to fill and hence three formulas.

iv. Compute the values of these cells using the part-of-speech frequencies in Table 3, the part-of-speech bigram frequencies in Table 4, and the pairs above. The `<s>` symbol and BOS mean a beginning of sentence. If you do not have a calculator, write the fractions.

v. In Table 6, draw arrows from the cells of the 3rd column that will contribute to fill the cells in the 4th column corresponding to *round*.

vi. Compute the values of the nonzero cells in the 4th column.

vii. If you have the time, compute the values of all the cells in the table and give the optimal part-of-speech sequence.

Table 5: Fill in the table with zeroes and crosses.

DT	0.0					
IN	0.0					
JJ	0.0					
MD	0.0					
NN	0.0					
NNP	0.0					
VB	0.0					
VBP	0.0					
WDT	0					
<s>	1.0					
	<s>	That	round	table	might	collapse
		$P(That \mid t_1)$	$P(round \mid t_2)$	$P(table \mid t_2)$	$P(might \mid t_4)$	$P(collapse \mid t_5)$

Table 6: Fill in the nonzero cells of the 3rd and 4th columns.

DT	0.0					
IN	0.0					
JJ	0.0					
MD	0.0					
NN	0.0					
NNP	0.0					
VB	0.0					
VBP	0.0					
WDT	0					
<s>	1.0					
	<s>	That	round	table	might	collapse
		$P(That \mid t_1)$	$P(round \mid t_2)$	$P(table \mid t_2)$	$P(might \mid t_4)$	$P(collapse \mid t_5)$

End of question 8.

Good Luck!