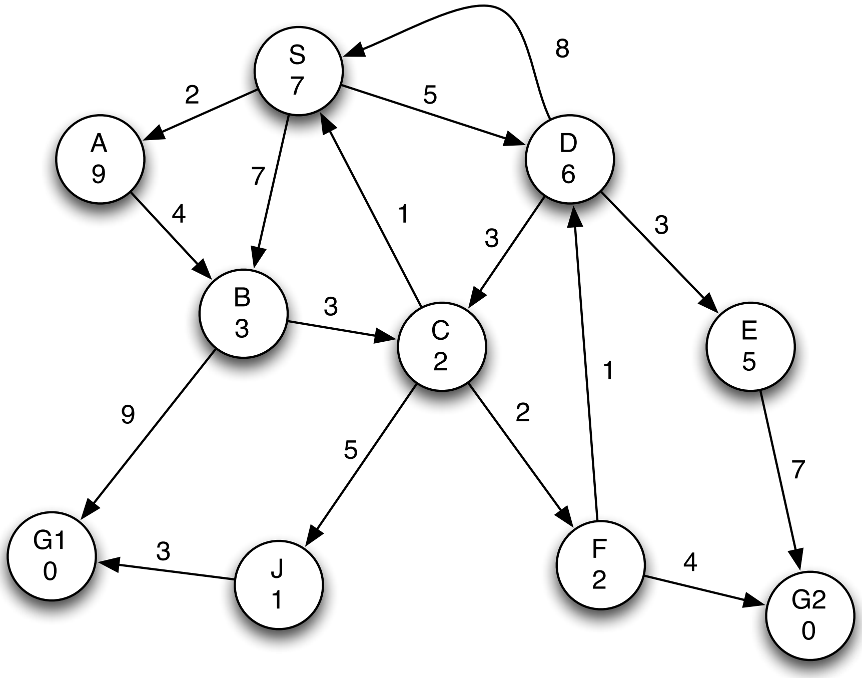
Lunds Tekniska Högskola EDA132
Institutionen för datavetenskap VT 2014

Tillämpad Artificiell Intelligens
Applied Artificial Intelligence
Tentamen 2014–03–11, 08.00–13.00, Sparta A,B

You can give your answers in English or Swedish.
You are welcome to use a combination of figures and text in your answers.
8 questions, 70 points, 50% needed for pass.

1 Search: (10 points)

Consider the search space shown below, 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. Remember that some nodes may occur on this list more than once!

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:____

Is the heuristic function you used in A* admissible? Motivate your answer.

2 Games: (8 points)

Consider the game of 2 × 2 tictactoe (Swedish: luffarschack) where each player has the additional option of passing (i.e., marking no square). Assume X goes first.

- Draw the full game tree down to depth 2. You need not show nodes that are rotations or reflections of siblings already shown. (Your tree should have five leaves.)
- Suppose the evaluation function is the number of Xs minus the number of Os. Mark the values of all leaves and internal nodes.
- Circle any node that would not be evaluated by alpha-beta during a left-to-right exploration of your tree.
- Suppose we wanted to solve the game to find the optimal move (i.e., no depth limit). Explain why alpha-beta with an appropriate move ordering would be much better than minimax.
- Briefly discuss how one might modify minimax so that it can solve the really exciting game of suicide 2 × 2 tictactoe with passing, in which the first player to complete 2-in-a-row loses. Describe optimal play for this game. [Hint: which is better — a move that definitely loses or a move whose value is unknown?]

3 Reasoning: (8 points)

In this question we will consider Horn Knowledge Bases (KBs), such as the following:

$$SpeaksSwedish(motherOf(x)) \Rightarrow SpeaksSwedish(x)$$

$$LivesInLund(x) \Rightarrow SpeaksSwedish(motherOf(x))$$

$$SpeaksSwedish(Eva)$$

$$LivesInLund(Anna)$$

where x is a variable, Eva and $Anna$ are constants and $motherOf$ is a function symbol.

Let FC be a "breadth-first" forward-chaining algorithm that repeatedly adds all consequences of currently satisfied rules; let BC be a "depth-first left-to-right" backward-chaining algorithm that tries clauses in the order given in the KB. (Hint: read **carefully** this paragraph again before answering!)

1. *True/False:* FC will infer the literal $LivesInLund(Eva)$.
2. *True/False:* FC will infer the literal $SpeaksSwedish(Anna)$.
3. *True/False:* If FC has failed to infer a given literal, then it is not entailed by the KB.
4. *True/False:* BC will return *True* given the query $SpeaksSwedish(Anna)$.
5. *True/False:* If BC does not return *True* given a query literal, then it is not entailed by the KB.

4 Planning: (8 points)

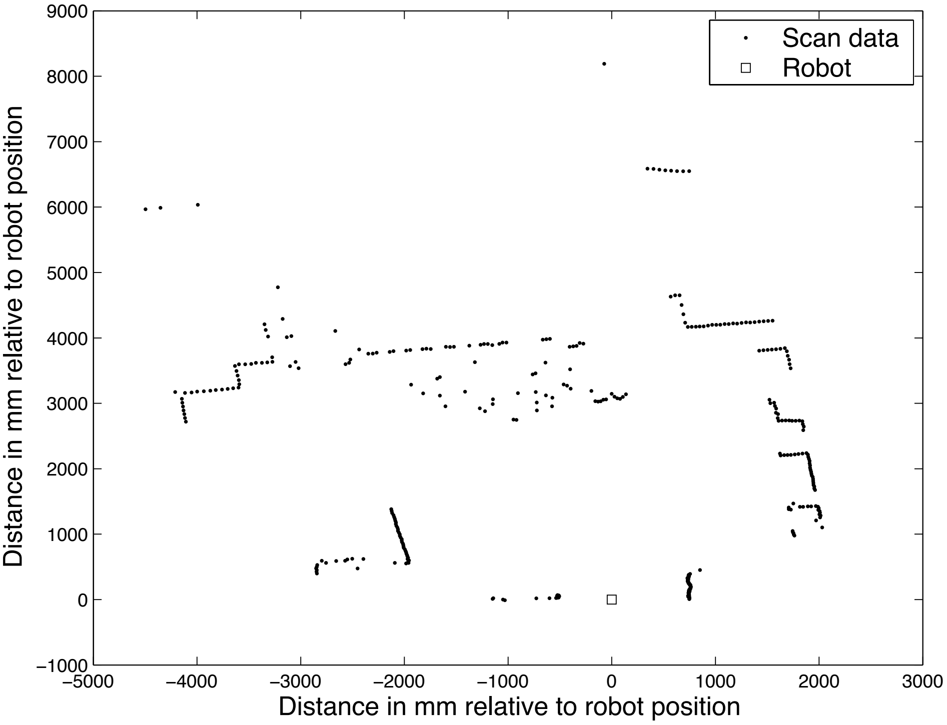
Consider the following planning problem, described using PDDL:

```
(define (domain bakery)
  (:requirements :strips)
  (:predicates (have-cake) (eaten-cake))
  (:action bake
    (:precondition (not (have-cake)))
    (:effect (have-cake)))
  (:action eat
    (:precondition (have-cake))
    (:effect (and (eaten-cake) (not (have-cake)))))
)
```

```
(define (problem baking)
  (:domain bakery)
  (:objects)
  (:init (have-cake))
  (:goal (and (have-cake) (eaten-cake)))
)
```

- Create the complete planning graph for it. Remember to put mutexes in appropriate places!
- There is an error in the domain description part. Modify it so that it is correct.

5 Learning and prediction: (8 points)



Imagine you want to find a method to determine whether a certain "leg-like" pattern observed in laser range data was caused by a person (a leg or legs) or some furniture standing in the field of view of a mobile robot (see the figure with an example data set of this kind). You have a bunch of original data sets, all consisting of a number of data points. There are sets stemming from situations where there were different constellations of persons and furniture (in terms of numbers) observable (visible for the robot). By previous analysis the data have been reduced to a collection of pattern sets, where you also have ground-truth annotations for each “leg-like” pattern telling you whether this was actually caused by a person or by furniture. This makes it possible for you to determine the number of leg-patterns and the number of furniture-patterns per data set, and it allows you to classify your data sets into different categories. The categories match the periods of time (situations) described above. Using this knowledge and the data sets as sample base you want to train different classifiers / learners to predict the most likely type of pattern to be observed next, to be able to compare this prediction with an actual new observation. You and your colleague want to keep the whole thing simple and start out with rather few samples (single annotated patterns drawn from one collection of observations), where you have no idea any more which period (situation) the collection represents. However, you have some a priori knowledge about how likely it is to look at a pattern from a certain situation, and for each of the situations, you know how likely it is to observe persons or furniture respectively (remember, there was some preprocessing, so all patterns are “leg-like” anyway). This a priori-knowledge is given as follows:

Situation (H_i)	$P(H_i)$	$P(p\|H_i)$	Comment
H_1: NO_PERSONS	0.1	0	All patterns due to furniture.
H_2: FEW_PERSONS	0.2	0.2	20% of patterns due to some person, rest due to furniture.
H_3: SOME_PERSONS	0.4	0.5	50% of patterns due to some person, rest due to furniture.
H_4: MANY_PERSONS	0.2	0.8	80% of patterns due to some person, rest due to furniture.
H_5: ONLY_PERSONS	0.1	1	No furniture visible due to so many persons around.

Your colleague and you choose two ways for learning and predicting to be able to compare different methods. One of you works with a MAP Learner (predicting based on the maximum-a-posteriori hypothesis), while the other uses the Optimal Bayes Learner (optimal classifier). Answer the following questions:

a) Using 10 samples from any data set, and testing the prediction against a new sample drawn from the same collection, your colleague's method produces quite some bold errors, while your method seems a bit more cautious. Who used the MAP Learner, and who tests the Optimal Learner approach? Why are the results that different? Describe the general advantages and disadvantages of the methods!

b) Assume that you observed 2 person-patterns in a row. What is the likelihood of observing another person pattern according to the MAP Learner?

c) Assume that you observed 2 furniture-patterns in a row. What is the likelihood of observing another furniture pattern according to the Optimal Bayes Learner?

Some hints:

Maximum Likelihood hypothesis: $h_{ML} = argmax_h P(D|h)$

Maximum a posteriori hypothesis: $h_{MAP} = argmax_h P(h|D)$

MAP Learner: $P(X|D) = P(X|h_{MAP})$

Optimal Bayes Learner: $P(X|D) = \sum_i P(X|h_i)P(h_i|D)$

6 Decision trees: (10 points)

1. Automatic learning can be roughly divided into two categories, unsupervised and supervised learning:

   (a) 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.

   (b) Describe what unsupervised learning is and give, or invent, a similar example.

2. ID3 is an algorithm that induces decision trees from data sets. ID3 uses the concept of entropy to evaluate the partition of a set.

   (a) Write the definition of the entropy of a binary partition consisting of P positive examples, and N negative ones.

   (b) Plot this entropy as a function of $x = \dfrac{P}{P+N}$, for x ranging from 0 to 1. You will compute the function value for 5 to 10 points and you will interpolate the rest. You will estimate these values, if you have no calculator.

(c) The Gini index is an alternative to entropy. It is defined as:

$$2 \cdot (\frac{P}{P+N} \cdot (1 - \frac{P}{P+N}) + \frac{N}{P+N} \cdot (1 - \frac{N}{P+N})).$$

Rewrite the Gini index using $x = \frac{P}{P+N}$.

(d) Plot the Gini index for x ranging from 0 to 1.

(e) What conclusion can you draw from these two plots?

(f) Imagine a third function that could replace either the Gini index or entropy.

3. At each step of the tree induction, starting from an input set of positive and negative examples, ID3 creates new partitions.

   (a) Given a set of attributes, where each attribute has a set of values, describe how ID3 creates such partitions. You will describe one single step and you will introduce the concept of information gain to evaluate the new partitions with regard to the input set.

   (b) Using an input set of 4 positive and 4 negative examples, you will evaluate two binary attributes, where:

      i. The first attribute splits the set into two subsets consisting of respectively 3 positive examples and a negative one $(3P, 1N)$, and one positive example and 3 negative ones $(1P, 3N)$.

      ii. The second attribute splits the set into two subsets consisting of respectively 2 positive examples and 4 negative ones $(2P, 4N)$, and 2 positive examples and zero negative ones $(2P, 0N)$.

      Draw graphically the two alternative trees, where each tree starts from a parent node corresponding to the input set.

   (c) Compute the information gain of each of these two attributes using:

      i. The entropy

      ii. The Gini index

      What conclusion can you draw?

7 Perceptron: (8 points)

The perceptron is a technique that learns classification models for sets consisting of linearly separable examples.

1. Defining the perceptron:

(a) Give the definition linear separability for a set of positive and negative examples and, using a figure in a two-dimensional space, draw two examples of:

i. a linearly separable set, and

ii. a nonlinearly separable one.

(b) The perceptron algorithm results in a hyperplane. Give the equation of such a hyperplane in a two-dimensional space. How many weights do you have to compute in such a space?

(c) The perceptron uses a so-called update rule to compute its model. Formulate it in a two-dimensional space using the batch and stochastic variants.

(d) What is the stop condition of the update rule?

2. Running the perceptron on an example:

(a) You will use the set below corresponding to class 0 and class 1:

Negative examples: (2, 1), (3, 2), (5, 1)

Positive examples: (1, 3), (2, 5), (2, 3)

And you will draw the points in a two-dimensional plane as well as a line separating the two classes.

(b) Run manually the perceptron update rule on the examples. You will start from a weight vector of ones and you will use a stochastic update rule. You will present the examples in the same order as above, that is: (2, 1), (3, 2), (5, 1), (1, 3), (2, 5), (2, 3). You should be able to find the weight vector in two epochs.

8 Language Technology: (10 points)

1. Defining part-of-speech tagging:

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

(b) 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

 35 table NN

 24 collapsed VBD
 14 collapsed VBN
```

where DT is the code for determiner, IN, for preposition, WDT, for relative pronoun, NN, for noun, VBD, for verb, past tense, and VBN, for verb, past participle.

Write all the possible part-of-speech sequences that apply to *That table collapsed*, if we consider the data extracted from the corpus.

(c) Out of these sequences, what is the sequence that corresponds to the most frequent parts of speech. Is it the correct one?

2. In the course, we used hidden Markov models to carry out part-of-speech tagging. We will examine here how to use discriminant classification methods, like decision trees or logistic regression, instead.

   (a) For a given word, w_i, in a sentence preceded by one word, w_{i-1}, and followed by one word, w_{i+1}, define part-of-speech tagging in terms of classes and attributes (features):

   - What would be the class?
   - What could be the attributes (features)?

   (b) Give an example of such a class and attributes for the word *table* in the sentence *That table collapsed.*

   (c) Formulate part-of-speech tagging as a classification problem.

   (d) Should you use decision trees, create a training set of two lines, that you will extract from this except:

   ```
   27      the         DT
   28      collapsed   VBN
   29      UAL         NNP
   30      Corp        NNP
   ```

   for the words *collapsed* and *UAL.* Use the ARFF format.

3. Should you use logistic regression, how would your transform the nominal data, words and parts of speech, into numbers?

Good Luck!