Försättsblad till skriftlig tentamen vid Linköpings Universitet

Datum för tentamen	2013-01-12
Sal (1) Om tentan går i flera salar ska du bifoga ett försättsblad till varje sal och <u>ringa in</u> vilken sal som avses	KÅRA
Tid	14-18
Kurskod	TDDC17
Provkod	TEN1
Kursnamn/benämning **Provnamn/benämning**	Artificiell intelligens En skriftlig tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	7
Jour/Kursansvarig Ange vem som besöker salen	Alexander Kleiner/ Piotr Rudol
Telefon under skrivtiden	Piotr: 0703167242
Besöker salen ca kl.	ca: 16:15
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Anna Grabska Eklund, ankn. 2362, anna.grabska.eklund@liu.se
Tillåtna hjälpmedel	Miniräknare/Hand calculators.
Övrigt	
Vilken typ av papper ska användas, rutigt eller linjerat	Valfritt
Antal exemplar i påsen	

Linköpings Universitet
Institutionen för Datavetenskap
Alexander Kleiner

Tentamen
TDDC17 Artificial Intelligence
12 January 2013 kl. 14-18

Points:

The exam consists of exercises worth 32 points.
To pass the exam you need 16 points.

Auxiliary help items:

Hand calculators.

Directions:

You can answer the questions in English or Swedish.
Use notations and methods that have been discussed in the course.
In particular, use the definitions, notations and methods in appendices 1-3.
Make reasonable assumptions when an exercise has been under-specified.
Begin each exercise on a new page.
Write only on one side of the paper.
Write clearly and concisely.

Jourhavande: Piotr Rudol, 0703167242. Piotr will arrive for questions around 16:15.

Question 1 (2+2)

Consider the following logical theory (where h and p are variables and john, hus are constants):

$$Lawyer(\text{john}) \quad (1)$$

$$House(\text{hus}, \text{john}) \quad (2)$$

$$\forall p\ [Lawyer(p) \Rightarrow Rich(p)] \quad (3)$$

$$\forall p \exists h\ House(h, p) \quad (4)$$

$$\forall p \forall h\ House(h, p) \wedge Rich(p) \Rightarrow Big(h) \quad (5)$$

$$\forall h\ [(Big(h) \wedge \exists p\ House(h, p)) \Rightarrow Work(h)] \quad (6)$$

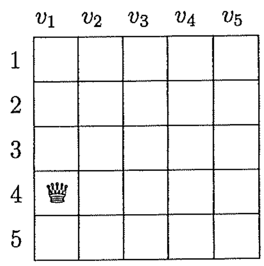
This theory asserts that john is a lawyer; john lives in a house (hus) ; lawyers are rich; any person has a house; rich people have big houses; and big houses are a lot of work to maintain. We would like to show using resolution that John's house is a lot of work to maintain. To do this, answer the following questions:

(a) Convert formulas (1) - (6) into clausal form with the help of appendix 1. **[2p]**

(b) Prove that $Work(\text{hus})$ is a logical consequence of (1) - (6) using the resolution proof procedure. **[2p]**

- Your answer should be structured using a resolution refutation tree (as used in the book).
- Since the unifications are trivial, it suffices to simply show the binding lists at each resolution step. Don't forget to substitute as you resolve each step.

Consider the 5-queens problem, where 5 pieces have to be placed on a size 5×5 board in such a way that no two queens are on the same horizontal, vertical, or diagonal line. The Variables $v_i \in V = \{v_1 \dots v_5\}$ indicate the position of the i–th queen with domain $dom(v_i) = 1, \dots, 5$ for all variables $v_i \in V$. Consider now state $\alpha = \{v_1 \mapsto 4\}$ shown below.



(a) Apply forward-checking in α. Fill the table below with the domains of the missing variables **[1p]**.

	v_1	v_2	v_3	v_4	v_5
init	$\{4\}$	$\{1,2,3,4,5\}$	$\{1,2,3,4,5\}$	$\{1,2,3,4,5\}$	$\{1,2,3,4,5\}$
FC	$\{4\}$				

(b) Enforce arc consistency in α. Specify the domains of the variables after applying arc consistency to all constraints belonging to one variable (E.g. in the first row you should depict the domains after enforcing arc consistency for all constraints containing v_2.) **[4p]**.

	v_1	v_2	v_3	v_4	v_5
init	$\{4\}$	$\{1,2,3,4,5\}$	$\{1,2,3,4,5\}$	$\{1,2,3,4,5\}$	$\{1,2,3,4,5\}$
AC - v_2	$\{4\}$				
AC - v_3	$\{4\}$				
AC - v_4	$\{4\}$				
AC - v_5	$\{4\}$				
	$\{4\}$				
	$\{4\}$				
	$\{4\}$				
	$\{4\}$				

Question 3 (2+2)

$u = 8$	$u = 13$	$u = 12$
$u = 2$	$r = 2$	$u = 10$
$u = 6$	$u = 12$	$u = 11$

Consider the grid world given in the diagram above. The u values specify the utilities after value iteration has been run to convergence, but the utility of the center state has been removed and instead you only know the reward r that was associated with that state. Assume that $\gamma = 0.7$ and that an agent can perform four possible actions: **North**, **South**, **East**, and **West**. With probability 0.8 the agent reaches the intended state, with probability 0.1 it moves to the right or to the left of the intended direction at right angles. For example, if it attempts to move East there is a 0.1 probabiliy that it ends up in the square to the north and south respectively. **Hint:** Make use of Appendix 4.

(a) What is the best action an agent can execute if it is currently in the center state of the grid world? Justify your answer. **[2p]**

(b) Compute the utility of the center state. Remember that the other states are already converged. **[2p]**

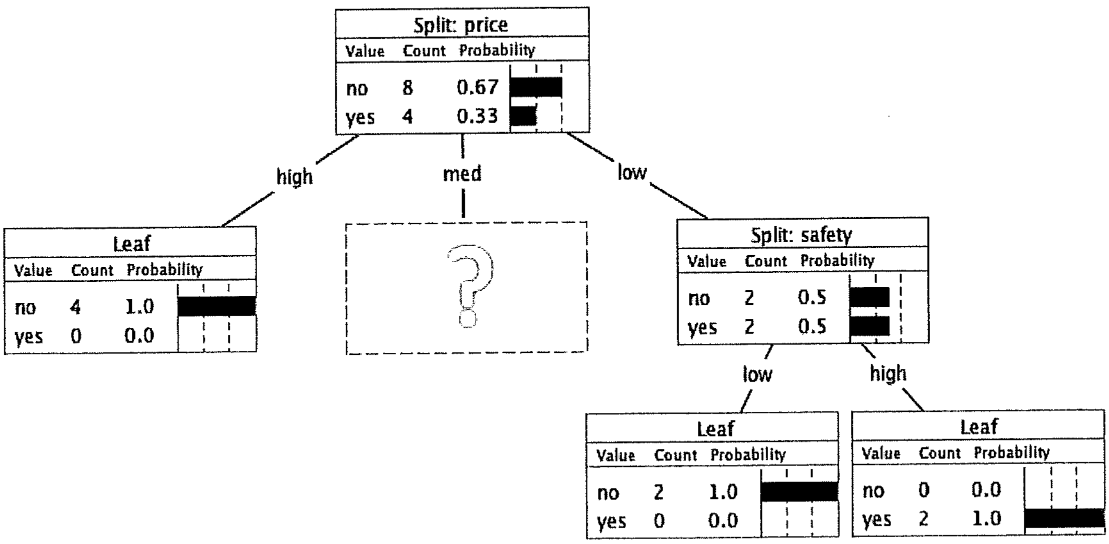
Question 4 (3+1)

No	Price	Maintenance Cost	Safety	Acceptable
1	high	med	low	no
2	high	med	high	no
3	high	high	high	no
4	high	med	high	no
5	med	med	high	yes
6	med	high	high	no
7	med	med	low	no
8	med	low	high	yes
9	low	high	low	no
10	low	high	high	yes
11	low	low	high	yes
12	low	low	low	no

Assume that a car dealership collects data on which cars a certain type of customer prefers. A small subset of this data is found in the table above.

(a) Below is an incomplete decision tree which by means of the attributes *Price*, *Maintenance Cost*, and *Safety* attempts to correctly classify wether such a customer will find a car acceptable or not. Value, Count and Probability in the boxes correspond to the values of the "Acceptable" target concept (yes/no), the number of examples at that point in the tree for each such value, and their proportions. Complete the middle branch of the decision tree *down to its leaves* using the Information Gain strategy (See Appendix) to decide which attribute to split on at each step. Show your calculations! **[3p]**

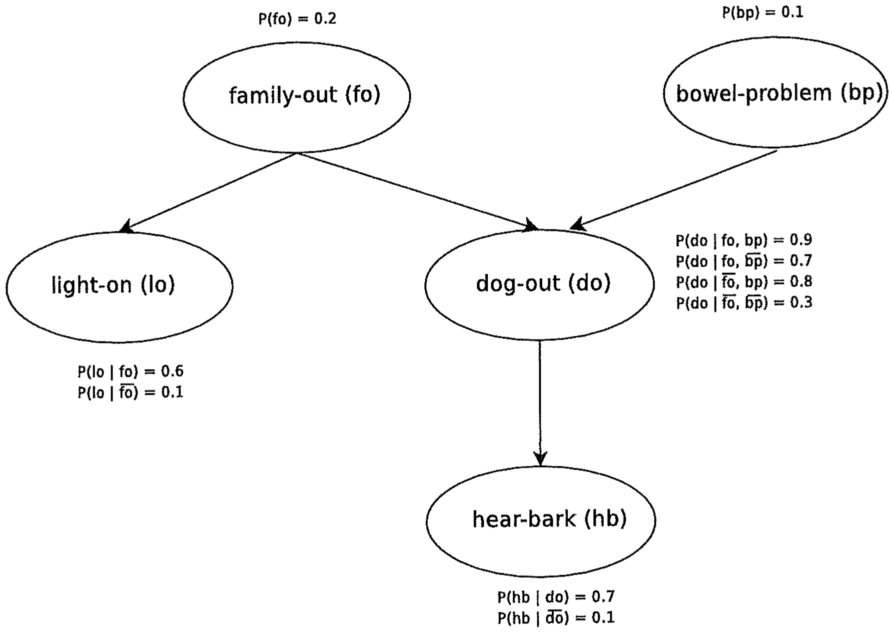
(b) When you attempt to learn a decision tree this way from a small number of examples will it always generalize well to new instances? Explain this phenomena. **[1p]**



Question 5 (0.5+0.5+3) BAYESIAN NETWORKS

Use the Bayesian network in the following figure together with the conditional probability tables below to answer the following questions. Appendix 2 may be helpful to use. Here,

- **lo** denotes the event that the outdoor light of the house is on.
- **fo** denotes the event that the family is not at home.
- **bp** denotes the event that the dog of the family has bowel problems.
- **do** denotes the event that the dog is in front of the door.
- **hb** denotes the event that one can hear barking in the vicinity of the house.



fo	P(fo)
T	0.2
F	0.8

bp	P(bp)
T	0.1
F	0.9

fo	lo	$P(lo \mid fo)$
T	T	0.6
T	F	0.4
F	T	0.1
F	F	0.9

fo	bp	do	$P(do \mid fo, bp)$
T	T	T	0.9
T	T	F	0.1
T	F	T	0.7
T	F	F	0.3
F	T	T	0.8
F	T	F	0.2
F	F	T	0.3
F	F	F	0.7

do	hb	$P(hb \mid do)$
T	T	0.7
T	F	0.3
F	T	0.1
F	F	0.9

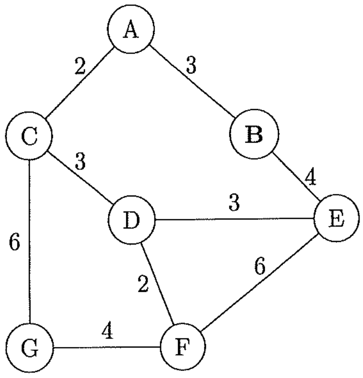
(a) Write the formula for the full joint probability distribution P(lo, fo, bp, do, hb) in terms of (conditional) probabilities derived from the shown bayesian network. **[0.5p]**

(b) Compute $P(lo, \neg fo, bp, \neg do, hb)$ **[0.5p]**

(c) Compute the probability that the family is out given the light is on and one can hear the dog barking $P(fo \mid lo, hb)$. **[3p]**

Question 6 (2+1+3)

Consider the following road map:



The straight-line distances between **F** and the other cities are given in the following table:

city	distance
A	6
B	5
C	4
D	1
E	4
G	3

(a) Draw the search tree generated by the A^* algorithm when searching for a shortest path from A to F, using as the heuristic the straight-line distance to F. Include all nodes in your drawing for which A^* would calculate f, g and h. Indicate in which order the nodes are expanded (expanding a node = applying the goal-test and adding it's children to the priority queue) and annotate each node with its f, g, and h value. **[2p]**

(b) Explain what an admissible heuristic function is using the notation and descriptions in (c). **[1p]**

(c) Let $h(n)$ be the estimated cost of the cheapest path from a node n to the goal. Let $g(n)$ be the path cost from the start node to n. Let $f(n) = g(n) + h(n)$ be the estimated cost of the cheapest solution through n. Provide a general proof that A* using tree-search is optimal if $h(n)$ is admissible. If possible, use a diagram to structure the proof. **[3p]**

Question 7 (1+2+2) PLANNING

Automated task planning is a central area in Artificial Intelligence.

(a) Heuristic functions can yield *plateaus* in the search space. What is a plateau? Visualize one using part of a search space: Show a set of search nodes, a number of possible transitions between nodes, and the type of heuristic values that characterize a plateau. Show which nodes are part of the plateau and explain in words *why* they form a plateau. **[1p]**

(b) In sequential planning with complete information, each search node can simply consist of a completely specified world state. The initial node in the search space then corresponds directly to the initial state specified in the planning problem description.

In contrast, *partial order causal link* (POCL) planning has no "current state". What does the initial search node in the POCL search space consist of? Illustrate its structure using a simple example. **[2p]**

(c) Explain in general terms how *planning graphs* can assist a backwards search procedure in efficiently finding a way from a goal state back towards the initial state. By "general terms" we mean that while you must clearly explain the ideas behind the graphs and show intuitively how they can improve backward search, you do not have to show all details in their structure or discuss the algorithms involved in actually constructing the graphs. **[2p]**

Appendix 1

Converting arbitrary wffs to clause form:

(a) Eliminate implication signs.

(b) Reduce scopes of negation signs.

(c) Standardize variables within the scopes of quantifiers (Each quantifier should have its own unique variable).

(d) Eliminate existential quantifiers. This may involve introduction of Skolem constants or functions.

(e) Convert to prenex form by moving all remaining quantifiers to the front of the formula.

(f) Put the matrix into conjunctive normal form. Two useful rules are:

- $\omega_1 \vee (\omega_2 \wedge \omega_3) \equiv (\omega_1 \vee \omega_2) \wedge (\omega_1 \vee \omega_3)$
- $\omega_1 \wedge (\omega_2 \vee \omega_3) \equiv (\omega_1 \wedge \omega_2) \vee (\omega_1 \wedge \omega_3)$

(g) Eliminate universal quantifiers.

(h) Eliminate $\wedge$ symbols.

(i) Rename variables so that no variable symbol appears in more than one clause.

Skolemization

Two specific examples. One can of course generalize the technique.
$\exists x P(x)$:
Skolemized: $P(\mathsf{c})$ where c is a fresh constant name.
$\forall x_1, \ldots, x_k, \exists y P(y)$:
Skolemized: $P(\mathsf{f}(x_1, \ldots, x_k))$, where f is a fresh function name.

Appendix 2

A generic entry in a joint probability distribution is the probability of a conjunction of particular assignments to each variable, such as $P(X_1 = x_1 \wedge \ldots \wedge X_n = x_n)$. The notation $P(x_1, \ldots, x_n)$ can be used as an abbreviation for this.

The chain rule states that any entry in the full joint distribution can be represented as a product of conditional probabilities:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid x_{i-1}, \ldots, x_1) \tag{7}$$

Given the independence assumptions implicit in a Bayesian network a more efficient representation of entries in the full joint distribution may be defined as

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid \mathit{parents}(X_i)), \tag{8}$$

where $\mathit{parents}(X_i)$ denotes the specific values of the variables in $\mathit{Parents}(X_i)$.

Recall the following definition of a conditional probability:

$$\mathbf{P}(X \mid Y) = \frac{\mathbf{P}(X \wedge Y)}{\mathbf{P}(Y)} \tag{9}$$

The following is a useful general inference procedure:

Let X be the query variable, let $\mathbf{E}$ be the set of evidence variables, let $\mathbf{e}$ be the observed values for them, let $\mathbf{Y}$ be the remaining unobserved variables and let α be the normalization constant:

$$\mathbf{P}(X \mid \mathbf{e}) = \alpha \mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y}) \tag{10}$$

where the summation is over all possible $\mathbf{y}$'s (i.e. all possible combinations of values of the unobserved variables $\mathbf{Y}$).

Equivalently, without the normalization constant:

$$\mathbf{P}(X \mid \mathbf{e}) = \frac{\mathbf{P}(X, \mathbf{e})}{\mathbf{P}(\mathbf{e})} = \frac{\sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})}{\sum_{\mathbf{x}} \sum_{\mathbf{y}} \mathbf{P}(\mathbf{x}, \mathbf{e}, \mathbf{y})} \tag{11}$$

Appendix 3: Decision Tree Learning

Definition 1
Given a collection S, containing positive and negative examples of some target concept, the entropy of S relative to this boolean classification is

$$Entropy(S) \equiv -p_{\oplus} log_2 p_{\oplus} - p_{\ominus} log_2 p_{\ominus},$$

where $p_{\oplus}$ is the proportion of positive examples in S and $p_{\ominus}$ is the proportion of negative examples in S.

Definition 2
Given a collection S, containing positive and negative examples of some target concept, and an attribute A, the information gain, $Gain(S, A)$, of A relative to S is defined as

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v),$$

where $values(A)$ is the set of all possible values for attribute A and S_v is the subset of S for which the attribute A has value v (i.e., $S_v = \{s \in S \mid A(s) = v\}$).

For help in converting from one logarithm base to another (if needed):

$$log_a x = \frac{log_b x}{log_b a}$$

$$ln\, x = 2.303\, log_{10}\, x$$

Note also that for the example, we define $0 log 0$ to be 0.

Appendix 4: Value Iteration

Value iteration is a way to compute the utility $U(s)$ of all states in a *known* environment (MDP) for the optimal policy $\pi^*(s)$.

It is defined as:

$$U(s) = R(s) + \gamma \max_{a \in \mathcal{A}} \sum_{s'} P(s'|s, a)U(s') \tag{12}$$

where $R(s)$ is the reward function, s and s' are states, γ is the discount factor. $P(s'|s, a)$ is the state transition function, the probability of ending up in state s' when taking action a in state s.

Value iteration is usually done by initializing $U(s)$ to zero and then sweeping over all states updating $U(s)$ in several iterations until it converges to the utility of the optimal policy $U_{\pi^*}(s)$.

A policy function defines the behavior of the agent, the action a it will take in each state s. Once we have computed the utility function of the optimal policy as above, $U_{\pi^*}(s)$, we can easily extract the optimal policy $\pi^*(s)$ itself by simply taking the action that leads to the highest expected utility in each state

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} \sum_{s'} P(s'|s, a)U_{\pi^*}(s') \tag{13}$$

where „arg max" means selecting the action that maximizes the expression.