Försättsblad till skriftlig tentamen vid Linköpings Universitet

Datum för tentamen	2012-10-23
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	TER2
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	Alexander Kleiner ankn 15 19/ Piotr Rudol 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	Hand calculator (miniräknare)
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
23 October 2012 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):

$$Lawyer(\mathsf{john}) \tag{1}$$
$$House(\mathsf{hus}, \mathsf{john}) \tag{2}$$
$$\forall p\ [Lawyer(p) \Rightarrow Rich(p)] \tag{3}$$
$$\forall p \exists h\ House(h, p) \tag{4}$$
$$\forall p \forall h\ House(h, p) \wedge Rich(p) \Rightarrow Big(h) \tag{5}$$
$$\forall h\ [(Big(h) \wedge \exists p\ House(h, p)) \Rightarrow Work(h)] \tag{6}$$

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:

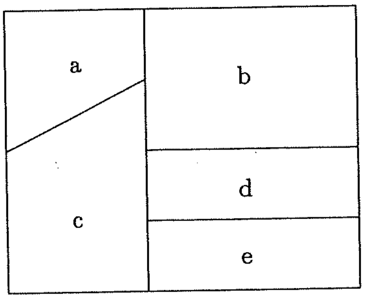
(a) Convert formulas (1) - (6) into clausal form with the help of appendix 1. **[2p]**

(b) Prove that $Work(\mathsf{hus})$ 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.

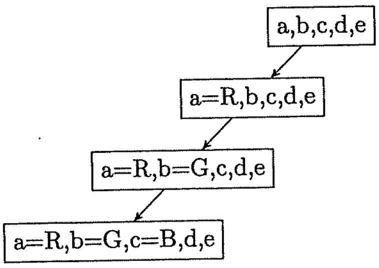
Question 2 (1+2+2)

Consider the following map-coloring problem. Five regions $\{a, b, c, d, e\}$ should be colored by three colors $\{R, G, B\}$. No two adjacent regions should have the same color.



(a) Draw a constraint graph using the regions as the nodes and the constraints as the arcs. **[1p]**

(b) Given the partial search tree (below) with the partial assignment $a = R, b = G, c = B$, expand the search tree further using backtracking search. Always try the values in the order R, G, B. Number the nodes according to the sequence in which they would be visited. **[2p]**



(c) Please illustrate the process of **forward checking** by filling the available values in the first three steps into the table. **[2p]**

step	assignment	a	b	c	d	e
0		R,G,B	R,G,B	R,G,B	R,G,B	R,G,B
1	a=R	R				
2	b=G	R	G			
3	c=B	R	G	B		

Question 3 (2+2)

$u = 2$	$u = 10$	$u = 4$
$u = -4$	$r = 2$	$u = 3$
$u = 6$	$u = 7$	$u = 11$

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 $\gamma = 0.5$ 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]**

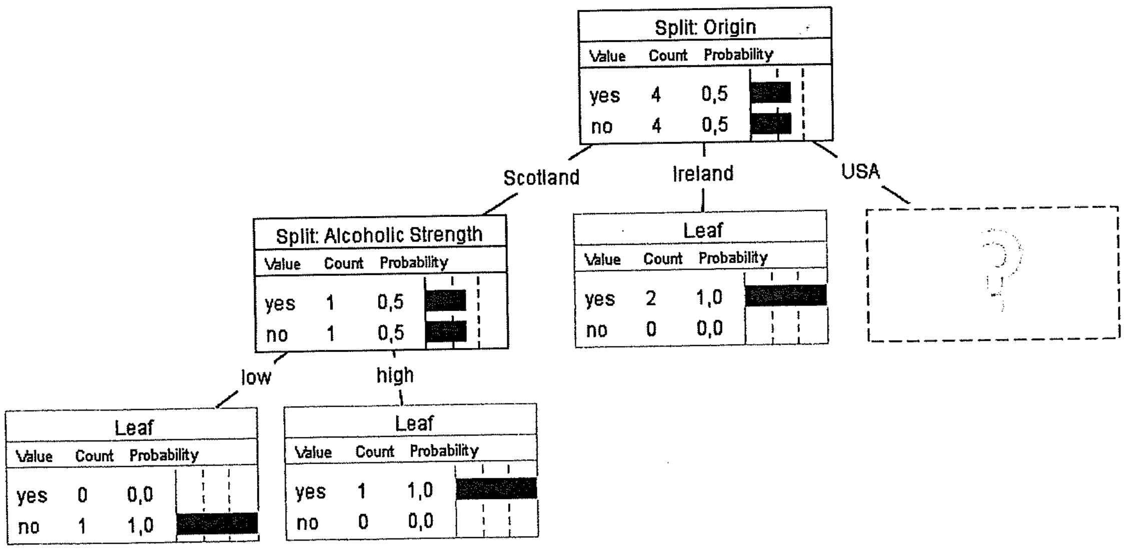
Question 4 (3+1)

No	Taste	Origin	Alcoholic Strength	High Quality?
1	dry	Ireland	high	yes
2	mild	USA	low	yes
3	dry	Scotland	low	no
4	dry	Ireland	high	yes
5	dry	USA	low	no
6	smoky	USA	low	no
7	dry	Scotland	high	yes
8	mild	USA	high	no

Consider the examples for properties of whisky brands as shown in the table above.

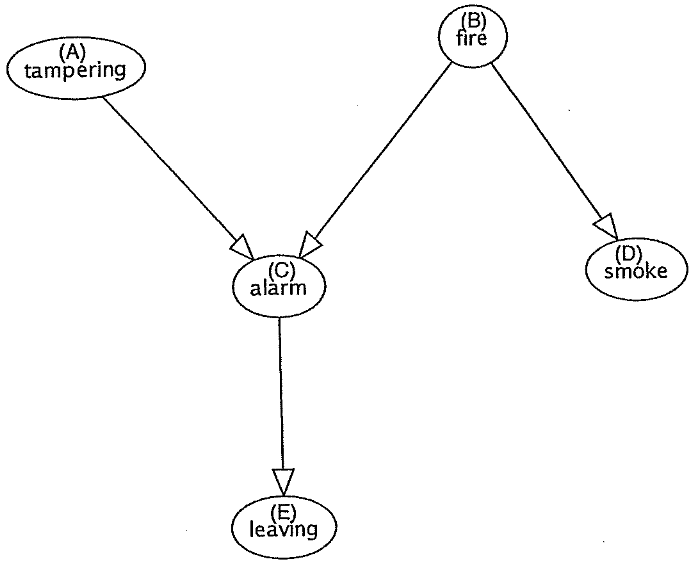
(a) Below is an incomplete decision tree which by means of the attributes *Taste*, *Origin*, and *Alcoholic Strength* attempts to correctly classify wether an example is of high quality or not. Value, Count and Probability in the boxes correspond to the values of the High-Quality target concept (yes/no), the number of examples at that point in the tree for each such value, and their proportions. Complete the right branch of the decision tree *down to its leaves* using the Information Gain strategy to decide which attribute to split on at each step (See Appendix 3). Show your calculations! **[3p]**

(b) Describe briefly (in at most two sentences), how overfitting can be reduced when learning descision trees. **[1p]**



Question 5 (1+1+2)

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.



A	P(A)
T	0.02
F	0.98

B	P(B)
T	0.01
F	0.99

B	D	$P(D \mid B)$
T	T	0.9
T	F	0.1
F	T	0.01
F	F	0.99

A	B	C	$P(C \mid A, B)$
T	T	T	0.5
T	T	F	0.5
T	F	T	0.85
T	F	F	0.15
F	T	T	0.99
F	T	F	0.01
F	F	T	0.1
F	F	F	0.9

C	E	$P(E \mid C)$
T	T	0.88
T	F	0.12
F	T	0.001
F	F	0.999

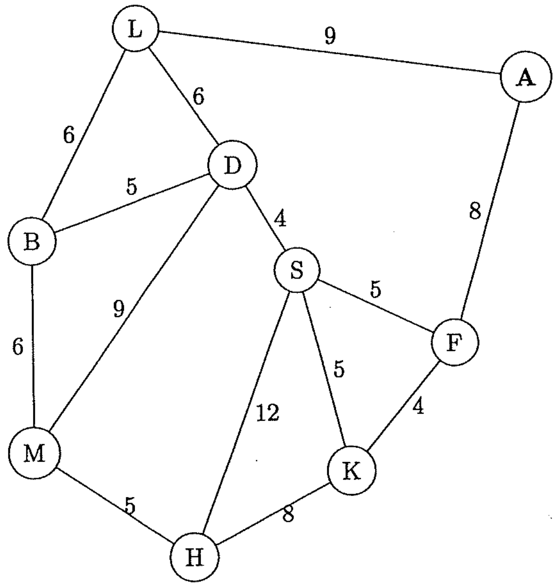
(a) Write the formula for the full joint probability distribution P(A, B, C, D, E) in terms of (conditional) probabilities derived from the shown bayesian network. **[1p]**

(b) Compute $P(a, \neg b, c, \neg d, e)$ **[1p]**

(c) Compute $P(e \mid a, c, \neg b)$ **[2p]**

Question 6 (2+1+3)

Consider the following road map:



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

city	**distance**
L	7
B	12
D	8
M	13
S	9
F	7
K	10
H	15

(a) Draw the **first expansion level** of the search tree generated by the A^* algorithm when searching for a shortest path from M to A, using as the heuristic the straight-line distance to A. 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) What is the advantage of A^* compared to greedy best-first search? **[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 (2+2+1)

Automated task planning is a central area in Artificial Intelligence.

(a) In partial-order planning, what is a *flaw* ? Name two distinct types of flaw and provide a clear explanation of each type. Illustrate each type of flaw using a small partial order plan that exhibits the flaw in question. [2p]

(b) Domain-independent heuristics play an important role in many planning algorithms. Some of these build on the use of a relaxation of the original planning problem. Explain the general concept of relaxation as well as the specific concept of delete relaxation. [2p]

(c) Either motivate clearly why a heuristic derived through relaxation is guaranteed to be admissible, or give a counterexample demonstrating why such a heuristic is sometimes inadmissible. [1p]

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.