Försättsblad till skriftlig tentamen vid Linköpings Universitet

Datum för tentamen	2011-08-19
Sal (1) Om tentan går i flera salar ska du bifoga ett försättsblad till varje sal och ringa in vilken sal som avses	TER3
Tid	14-18
Kurskod	TDDC17
Provkod	TEN1
Kursnamn/benämning **Provnamn/benämning**	Artificiell intelligens En skriftlig tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	8
Jour/Kursansvarig Ange vem som besöker salen	Mariusz Wzorek
Telefon under skrivtiden	0703-88 71 22
Besöker salen ca kl.	ca. kl. 15:00
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	
Antal exemplar i påsen	

Linköpings Universitet
Institutionen för Datavetenskap
Patrick Doherty

Tentamen
TDDC17 Artificial Intelligence
19 august 2011 kl. 14-18

Points:

The exam consists of exercises worth 33 points.
To pass the exam you need 17 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: Mariusz Wzorek, 0703887122. Mariusz will arrive for questions around 15.00.

1. Consider the following theory (where x, y and z are variables and history, lottery and john are constants):

$$\forall x([Pass(x, \mathsf{history}) \wedge Win(x, \mathsf{lottery})] \Rightarrow Happy(x)) \quad (1)$$

$$\forall x \forall y([Study(x) \vee Lucky(x)] \Rightarrow Pass(x, y)) \quad (2)$$

$$\neg Study(\mathsf{john}) \wedge Lucky(\mathsf{john}) \quad (3)$$

$$\forall x(Lucky(x) \Rightarrow Win(x, \mathsf{lottery})) \quad (4)$$

   (a) Convert formulas (1) - (4) into clause form. **[1p]**

   (b) Prove that *Happy*(john) is a logical consequence of (1) - (4) 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.

2. Constraint satisfaction problems consist of a set of variables, a value domain for each variable and a set of constraints. A solution to a CS problem is a consistent set of bindings to the variables that satisfy the contraints. A standard backtracking search algorithm can be used to find solutions to CS problems. In the simplest case, the algorithm would choose variables to bind and values in the variable's domain to be bound to a variable in an arbitrary manner as the search tree is generated. This is inefficient and there are a number of strategies which can improve the search. Describe the following three strategies:

   (a) Minimum remaining value heuristic (MRV). **[1p]**

   (b) Degree heuristic. **[1p]**

   (c) Least constraining value heuristic. **[1p]**

   Constraint propagation is the general term for propagating constraints on one variable onto other variables. Describe the following:

   (d) What is the Forward Checking technique? **[1p]**

   (e) What is arc consistency? **[1p]**

3. The following questions pertain to the course article by Newell and Simon entitled *Computer Science as an Empirical Enquiry: Symbols and Search.*.

   (a) What is a *physical symbol system* (PSS) and what does it consist of? **[2p]**

   (b) What is the Physical Symbol System Hypothesis? **[1p]**

4. Use the Bayesian network in Figure 1 together with the conditional probability tables below to answer the following questions. Appendix 2 may be helpful to use.

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

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

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

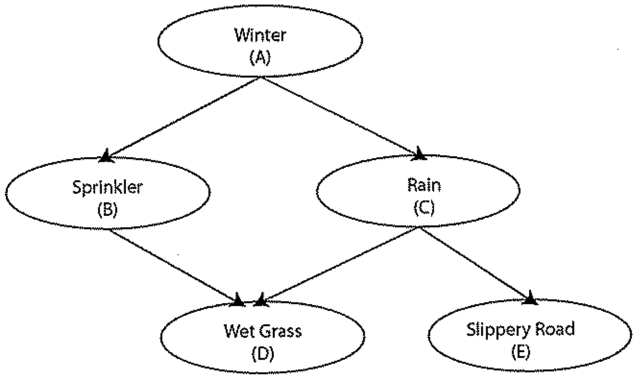

Figure 1: Bayesian Network Example

A	P(A)
T	.6
F	.4

A	B	$P(B \mid A)$
T	T	.2
T	F	.8
F	T	.75
F	F	.25

A	C	$P(C \mid A)$
T	T	.8
T	F	.2
F	T	.1
F	F	.9

B	C	D	$P(D \mid B, C)$
T	T	T	.95
T	T	F	.05
T	F	T	.9
T	F	F	.1
F	T	T	.8
F	T	F	.2
F	F	T	0
F	F	F	1

C	E	$P(E \mid C)$
T	T	.7
T	F	.3
F	T	0
F	F	1

5. A* search is the most widely-known form of best-first search. The following questions pertain to A* search:

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

   (b) Suppose a robot is searching for a path from one location to another in a rectangular grid of locations in which there are arcs between adjacent pairs of locations and the arcs only go in north-south (south-north) and east-west (west-east) directions. Furthermore, assume that the robot can only travel on these arcs and that some of these arcs have obstructions which prevent passage across such arcs.

   The *Mahattan distance* between two locations is the shortest distance between the locations ignoring obstructions. Is the Manhattan distance in the example above an admissible heuristic? Justify your answer explicitly. **[2p]**

   (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 n_0 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. **[2p]**

6. The following question pertains to Decision Tree Learning. Use the definitions and data table in Figure 3, Appendix 3 to answer this question. Figure 2 shows a partial decision tree for the Table in Figure 3 with target attribute *PlayTennis*.

   (a) What attribute should be tested in the box with the question mark on the right branch of the decision tree in Figure 2? Justify your answer by computing the information gain for the appropriate attributes in the Table in Figure 3 in Appendix 3. **[3p]**

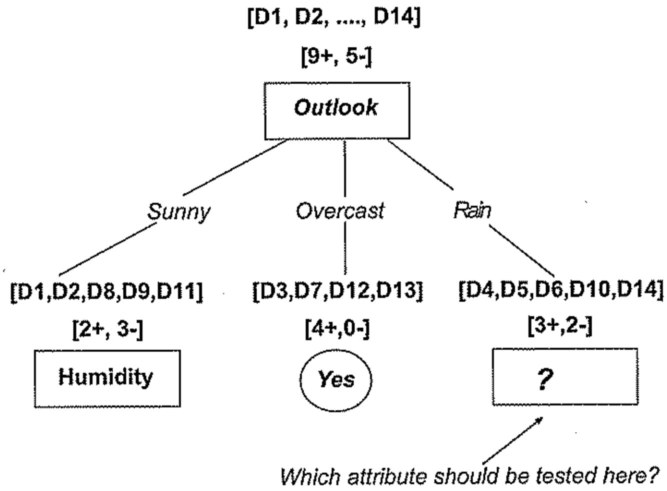

Figure 2: Partial Decision Tree for PlayTennis

7. The following questions pertain to partial-order planning:

   (a) What is partial-order planning and how does it differ from STRIPS-based planning in terms of search space and output of the respective planning algorithms?**[3p]**

   (b) Describe the four basic components in a partial-order plan according to Russell/Norvig. **[2p]**

8. Modeling actions and change in incompletely represented, dynamic worlds is a central problem in knowledge representation. The following questions pertain to reasoning about action and change.

   (a) What is Temporal Action Logic? Explain by describing the ontology used in the formalism, that is, what is taken to exist, and what notation is used in the logical language to represent those things that are taken to exist. **[2p]**

   (b) What is the frame problem? Use the Wumpus world to provide a concrete example of the problem. Represent the problem by representing an initial timepoint, an action and the result of the action using the TAL notation (either with macros or without). **[2p]**

   (c) What is nonmonotonic logic? How can it be used to provide solutions to the frame problem? **[2p]**

Appendix 1

Converting arbitrary wffs to clause form:

1. Eliminate implication signs.
2. Reduce scopes of negation signs.
3. Standardize variables within the scopes of quantifiers (Each quantifier should have its own unique variable).
4. Eliminate existential quantifiers. This may involve introduction of Skolem constants or functions.
5. Convert to prenex form by moving all remaining quantifiers to the front of the formula.
6. Put the matrix into conjunctive normal form. Two useful rules are:
   - $\omega_1 \vee (\omega_2 \wedge \omega_3) \equiv (\omega_1 \vee \omega_2) \wedge (\omega_1 \vee \omega_3)$
   - $\omega_1 \wedge (\omega_2 \vee \omega_3) \equiv (\omega_1 \wedge \omega_2) \vee (\omega_1 \wedge \omega_3)$
7. Eliminate universal quantifiers.
8. Eliminate $\wedge$ symbols.
9. Rename variables so that no variable symbol appears in more than one clause.

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:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid x_{i-1}, \ldots, x_1) \tag{5}$$

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

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i)), \tag{6}$$

where $parents(X_i)$ denotes the specific values of the variables in $Parents(X_i)$.

Recall the following definition of a conditional probability:

$$\mathbf{P}(X \mid Y) = \frac{\mathbf{P}(X \wedge Y)}{\mathbf{P}(Y)} \tag{7}$$

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:

$$\mathbf{P}(X \mid \mathbf{e}) = \alpha \mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y}) \tag{8}$$

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:

$$\mathbf{P}(X \mid \mathbf{e}) = \frac{\mathbf{P}(X, \mathbf{e})}{\mathbf{P}(\mathbf{e})} = \frac{\sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})}{\sum_{\mathbf{x}} \sum_{\mathbf{y}} \mathbf{P}(\mathbf{x}, \mathbf{e}, \mathbf{y})} \tag{9}$$

Appendix 3

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., $S_v = \{s \in S \mid A(s) = v\}$).

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Figure 3: Sample Table for Target Attribute PlayTennis

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 $0log0$ to be 0.