Försättsblad till skriftlig tentamen vid Linköpings universitet

Datum för tentamen	2016-08-22
Sal (2)	TER2 TERE
Tid	8-12
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	0703887122
Besöker salen ca klockan	ca kl. 10
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Anna Grabska Eklund, anna.grabska.eklund@liu.se, ankn. 2362
Tillåtna hjälpmedel	Miniräknare/Hand calculators
Övrigt	
Antal exemplar i påsen	

Försättsblad till skriftlig tentamen vid Linköpings universitet

Datum för tentamen	2016-08-22
Sal (2)	TER2 TERE
Tid	8-12
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	0703887122
Besöker salen ca klockan	ca kl. 10
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Anna Grabska Eklund, anna.grabska.eklund@liu.se, ankn. 2362
Tillåtna hjälpmedel	Miniräknare/Hand calculators
Övrigt	
Antal exemplar i påsen	

Linköpings Universitet
Institutionen för Datavetenskap
Patrick Doherty

Tentamen
TDDC17 Artificial Intelligence
22 August 2016 kl. 08-12

Points:

The exam consists of exercises worth 39 points.
To pass the exam you need 19 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 1-2.
Make reasonable assumptions when an exercise has been under-specified.
State these assumptions explicitly in your answer.
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 10.00.

1. Based on the material in Chapter 2 of the course book, define succinctly in your own words, the following terms:

    (a) Agent, Agent Function, Agent Program **[1p]**

    (b) Performance Measure, Rationality, Autonomy **[1p]**

    (c) Reflex Agent. Also provide a schematic diagram of such an agent. **[1p]**

    (d) Model-based Agent. Also provide a schematic diagram of such an agent. **[1p]**

    (e) Goal-based Agent. Also provide a schematic diagram of such an agent. **[1p]**

2. 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]**

    (c) Do you think the Physical Symbol System Hypothesis is true or false or somewhere in between? Provide reasonable justifications for your opinion. **[2p]**

3. Consider the following logical theory about elks (where x and y are variables and fred, ted and sam are constants):

$$Red(\mathsf{sam}) \tag{1}$$

$$White(\mathsf{fred}) \tag{2}$$

$$Likes(\mathsf{fred}, \mathsf{ted}) \tag{3}$$

$$Likes(\mathsf{ted}, \mathsf{sam}) \tag{4}$$

$$\neg(Red(\mathsf{ted}) \wedge White(\mathsf{ted})) \tag{5}$$

$$\neg(\neg Red(\mathsf{ted}) \wedge \neg White(\mathsf{ted})) \tag{6}$$

We would like to show using resolution that a red elk likes a white elk. To do this, answer the following questions:

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

(b) Prove that $\exists x \exists y\, (White(x) \wedge Red(y) \wedge Likes(x, y))$ is a logical consequence of (1) - (6) using the resolution proof procedure. **[3p]**

- Your answer should be structured using a resolution refutation tree (as used in the book or course slides).
- 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.

4. 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 or provide the following:

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

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

   (f) Provide a constraint graph that is arc consistent but globally inconsistent. **[1p]**

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.
   Provide an admissible heuristic for this problem. Explain why it is an admissible heuristic and 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 sufficiently rigid proof that A* is optimal if $h(n)$ is admissible. You need only provide a proof for either tree-search (seminar slides) or graph-search (in course book). If possible, use a diagram to structure the contents of the proof to make it more readable. **[2p]**

6. The following questions pertain to automated planning.

   (a) State three distinct and important reasons why planning should be automated. In other words, why do we want *computers* to create plans, rather than simply creating the plans ourselves? **[1p]**

   (b) What is *satisficing* planning? **[1p]**

   (c) Recall that the two heuristic functions $h_1(n)$ and $h_{\mathsf{add}}(n)$ are very similar, but that their definitions have one crucial difference that results in significant differences in their behavior. What is this crucial difference? Which of the functions is inadmissible, and why? Which function provides more information, and why? **[2p]**

   (d) Plan-space search begins with an *initial plan* and iteratively modifies this plan in order to repair *flaws*. Name the two distinct *types* of flaw that exist. Explain clearly how these flaw types can be identified in a plan. For the explanation to be sufficiently clear, you may need to provide an example plan drawn in the same way as during the lectures. **[2p]**

7. The following questions pertain to machine learning. Give detailed answers.

   (a) Explain supervised and reinforcement learning in terms of the input and output of the respective types of algorithms, highlighting their differences. **[2p]**

   (b) Explain the concept of *overfitting* and a princpled way to detect it. **[1p]**

   (c) Explain when and why *exploration* is needed and outline an example algorithm. **[1p]**

8. 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. If you do not have a hand-held calculator with you, make sure you set up the solution to the problems appropriately for partial credit.

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

   (c) What is $P(d \mid a, c, \neg b)$? **[2p]**

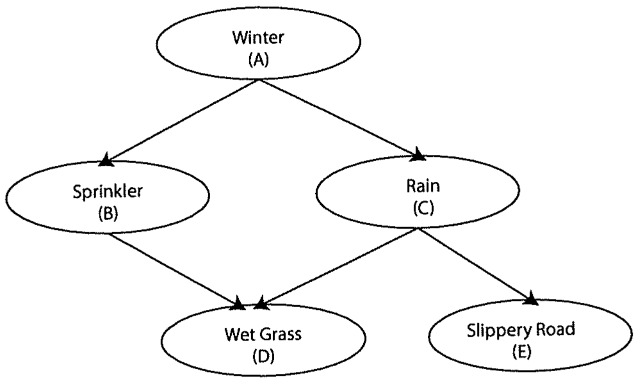

Figure 1: Bayesian Network Example

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

Appendix 1

Converting arbitrary wffs to CNF 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.

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.

Useful Equivalences

$\neg \forall x\, \omega(x) \equiv \exists x\, \neg \omega(x)$
$\neg \exists x\, \omega(x) \equiv \forall x\, \neg \omega(x)$

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:

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