Exam Introduction Intelligent Systems, January 7th, 2011, 14.00 – 17.00

There are 9 questions that together are worth 90 points. The first 10 points you get for free.

Excercise	1	2	3	4	5	6	7	8	9
Points	8	10	10	12	12	8	10	10	10

Always explain your answers. Good luck!

1. First-order logic (FOL) is more expressive than propositional logic (PL).
   (a) Explain what it is, that can be expressed in FOL, but not in PL.
   (b) Under certain conditions, the role of first-order quantifiers can be simulated in propositional logic. Explain what the conditions are and how the simulation works.

2. Translate the following sentences into First-Order Logic:
   (a) If two persons with different religions sleep in one bed, the devil sleeps between them (Dutch saying).
   (b) Jan sees the mote in his brother's eye but does not see the beam in his own (Dutch / English saying).

3. Two well-known problems with the logic representation of actions are the *frame problem*, and the *qualification problem*.
   (a) Explain these problems briefly.
   (b) Russell and Norvig describe a solution to the representational frame problem in the situation calculus. Explain this solution, using an example and (if you can) formulas.

4. (a) Explain the difference between logic programming and Prolog.
   (b) Explain the difference between a green and a red cut.
   (c) What will be the result if the following Prolog program is asked whether **d** is a scsi drive?

```
drive(d).
scsi_drive(X) :- drive(X), not(ide_drive(X)).
ide_drive(X) :- drive(X), not(scsi_drive(X)).
```

5. A palindrome is a word that reads the same in reverse direction. Examples are `['l','e','p','e','l']` and `['p','a','r','t','e','r','e','t','r','a','p']`.
   Blackburn gives the following program to check if a word is a palindrome:

```
palindrome(List) :- reverse(List,List)
```

   There are also solutions not using **`reverse/2`**. One solution uses recursion and **`append/3`** (append two lists together to yield a third list). Give this solution. If you cannot do so, but do know how to explain the solution in words, this might also give you some points.

6. A plan library, as used for HTN planning, contains knowledge concerning the decomposition of plans. Does this knowledge put a constraint on the possible solutions to the planning problem in the sense that every plan has to comply with at least one of the decompositions in the plan library? If so, explain how a planner finds this decomposition. If not, give an alternative explanation of the role of the knowledge in the plan library.

see the other side

7. A planning problem is described using STRIPS in the following way:

- Initial siutation: `A ∧ C ∧ D`
- The goal: `B`
- The actions:

```
Act1                     Act2
PRECOND: D               PRECOND: A ∧ C
EFFECT: C ∧ ¬E           EFFECT: E ∧ ¬A

Act3                     Act4
PRECOND: C ∧ E ∧ F       PRECOND: D
EFFECT: B                EFFECT: F ∧ ¬C
```

(a) Solve this planning problem.
(b) Mention the restrictions of the STRIPS representation formalism.
(c) STRIPS uses the Closed World Assumption. Explain for each of the aspects 'initial situation', 'goal', 'precondition' and 'effect' of a STRIPS description whether or not STRIPS assumes the CWA or not.

8. Your doctor has bad news. You tested positive for a serious disease and the test is 98% accurate. The doctor says that the 98% accuracy means that the probability of testing positive if you have the disease is 0.98, and that the probability of testing negative if you do not have the disease is also 0.98. The doctor says there might also be good news: the disease is rare, striking only 1 out of 100000 people of your age.
(a) You claim that the good news is no good news since the test has a known accuracy. Who is right, and why?
(b) Is the information "the probability of testing negative if you do not have the disease is also 0.98" relevant for determining your chance of having the disease? Explain your answer.
(c) Give the calculation for the chance that you actually have the disease.

9. (a) Explain the difference between value iteration and policy iteration.
Below two pictures from the book concerning an optimal policy for a Markov decision process where the (dis)reward in each non-terminal state is -0.04. The left picture shows the optimal policy when the chances that as a result of an action one ends up in the room one is heading for are 0.8, and the chances that one ends up in a room to the left or to the right are 0.1 for both possibilities. Bumping into a wall means that one stays in the same room. The rooms with rewards +1 and -1 are terminal states. The right picture shows the utilities associated with the states, for the optimal policy of the left picture.
(b) In what way do we have to adapt the dis-reward in order to let the agent decide in (3.1) not to take the 'detour' via (1,1) and (1,3)?