Family Name: ....

Other Names: ....

Signature: ....

Student Number: ....

THE UNIVERSITY OF NEW SOUTH WALES
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Examination – Semester 2, October, 2010

COMP3431: Robot Architectures - EXAMPLE

Exam Duration: 3 hours **Total marks for this paper: 76**

Reading Time: 15 minutes

This paper has 9 pages including this cover page.

Authorised materials: ANY

Instructions to Invigilators:

- Please provide students with standard script books.
- Please **collect** exam papers at the conclusion of the exam.

Instructions to Students:

- This paper counts for 45% of your final grade.
- Answer in the **Script Books** provided.
- **DO NOT REMOVE QUESTION PAPER**
(Hand in this entire examination paper when you are finished).
- ALL questions are COMPULSORY.
- Start your answer to each question on a new page.
- There is no guarantee that all questions are solvable. Do **not** get stuck on one question to the detriment of later questions.

Architectures/Overview

Question 1.

(a) (3 marks): List two different ways of classifying an agent's environment. (Each way should be independent, so 'foo' and 'not foo' is one way of classifying an environment, not two.)

(b) (1 mark): Define 'Partially Observable'.

(c) (2 marks): List two different types of solution or plan specification

Bayesian Tracking

Question 2.

You have designed a mars rover. It has a sensor that can detect the density of some rock. Unfortunately the sensor is noisy. The number returned by the sensor is $3d + N(3, 100)$, where d is the actual density, and $N(\mu, \sigma^2)$ specifies a Gaussian distribution with mean μ and variance σ^2. You decide to use a Bayes' filter to help overcome the noise of the sensor and to help you detect which type of rock you are sensing. Here are rough numerical approximations to the sensor model in tabular form.

Sensor Model		Sensor Output				
		$x < 2$	$2 < x < 8$	$8 < x < 14$	$14 < x < 20$	$x > 20$
Rock Density	0	0.46	0.23	0.18	0.09	0.04
	2	0.24	0.22	0.23	0.17	0.14
	4	0.1	0.15	0.22	0.23	0.30
	6	0.03	0.07	0.15	0.22	0.53

(a) (2 marks): Derive Bayes' rule for two probabilities. $P(A|B) = ?$

(b) (4 marks): Which type of Bayes' filter is most appropriate for this problem: a Kalman filter, a particle filter or a table based Bayes' filter. Why?

For the next few parts of this question we will assume a table based filter (because it is easiest to ask questions about, don't assume it is the best for the question above simply because the following questions use it):

(c) (2 marks): What is your state representation for your table based filter?

(d) (1 mark): What is your initial state assuming you have no idea what sort of rock you are currently sensing?

Your sensor returns output of 3 when sensing a rock for the first time.

(e) (4 marks): What is the your new state?

(f) (1 mark): At this point, what is your best estimate as to what density of rock you are sensing?

You sense the same rock again, and this time get receive output of -1.

(g) (4 marks): What is your new state?

(h) (1 mark): At this point, what is your best estimate as to what density of rock you are sensing?

You have a friend who thinks Particle Filters are the way to go. He has implemented one, with five particles. You decide to help him debug his filter by looking at its execution during the observation update. The five particles his filter currently contains are: $d = 2, d = 2, d = 3, d = 3$ and $d = 4$.

(i) (1 mark): You notice that some particles are duplicates. Does this, by itself, indicate your friend has a bug? Why?

(j) (2 marks): The rover's sensor returns an output of 3, as above. What weights are assigned to the particles given this sensor reading?

(k) (2 marks): Given the weights from the previous part, what are the probabilities that you sample a particular particle during the re-sampling stage of the particle filter?

Question 3.

You want to make a cricket playing robot. This robot needs to track the ball through the air. You decide to use a state space of $x, y, z, \dot{x}, \dot{y}, \dot{z}$ to track the position and velocity of the ball in three dimensions. Your observations are taken from the position of the ball in a video camera image.

A friend suggests that position is independent of velocity, and so you can simplify things by using two Kalman filters, one over x, y, z and a second over $\dot{x}, \dot{y}, \dot{z}$.

(a) (2 marks): What effect will this simplification have on the Kalman filter?

Planning

Question 4.

All parts of this question refer to the following set of planning operators (for the snlp planner):

```
;; Define step for putting a block on the table.
(defstep :action '(newtower ?x)
  :precond '((on ?X ?Z) (clear ?X))
  :add '((on ?X Table) (clear ?Z)) :dele'((on ?X ?Z))
  :equals '((not (?X ?Z)) (not (?X Table)) (not (?Z Table))))

;; Define step for placing one block on another.
(defstep :action '(puton ?X ?Y)
  :precond '((on ?X ?Z) (clear ?X) (clear ?Y))
  :add   '((on ?X ?Y) (clear ?Z))
  :dele  '((on ?X ?Z) (clear ?Y))
  :equals '((not (?X ?Y)) (not (?X ?Z)) (not (?Y ?Z))
            (not (?X Table)) (not (?Y Table)))))
```

Given this start state:

`(on C A) (on A Table) (on B Table) (clear C) (clear B)`

and this goal state:

`(on A B) (on B C)`

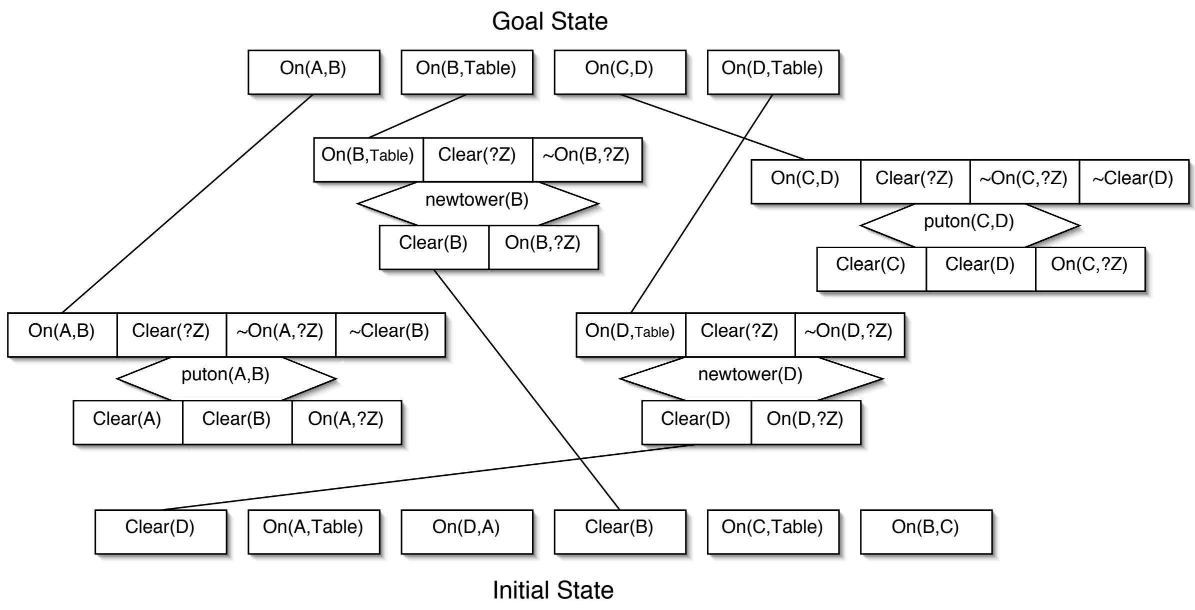
(a) (2 marks): Give a possible first step for a backward chaining (using regression) state space planner.

(b) (2 marks): Give a possible first step for a forward chaining (using progression) state space planner.

(c) (2 marks): Give a possible first step for a partial order plan space planner.

(d) (3 marks): In state space planning (and graph search) there are both forward and backward chaining planners. In plan space planning there are partial order planners. Do the concepts of forward and backward chaining make sense in a partial order setting? Explain.

(e) (4 marks): Given the following incomplete partial-order plan, list all the threats, and for each threat list all ways to resolve it. Note: this plan uses the same domain as above, but a different start state and goal.


Goal State
On(A,B)
On(B,Table)
On(C,D)
On(D,Table)
On(B,Table)
Clear(?Z)
~On(B,?Z)
newtower(B)
Clear(B)
On(B,?Z)
On(C,D)
Clear(?Z)
~On(C,?Z)
~Clear(D)
puton(C,D)
Clear(C)
Clear(D)
On(C,?Z)
On(A,B)
Clear(?Z)
~On(A,?Z)
~Clear(B)
puton(A,B)
Clear(A)
Clear(B)
On(A,?Z)
On(D,Table)
Clear(?Z)
~On(D,?Z)
newtower(D)
Clear(D)
On(D,?Z)
Clear(D)
On(A,Table)
On(D,A)
Clear(B)
On(C,Table)
On(B,C)
Initial State

Continuous Search

Question 5.

(a) (2 marks): In a Rapidly exploring Randomised Tree (RRT) search, there are two standard ways to choose a point to expand the tree. List them.

(b) (3 marks): A salesman comes to you with a search algorithm for optimising a function. It samples any given function, and is guaranteed to find a global minimum in a finite number of samples. Should you buy shares in this company? Why or why not?

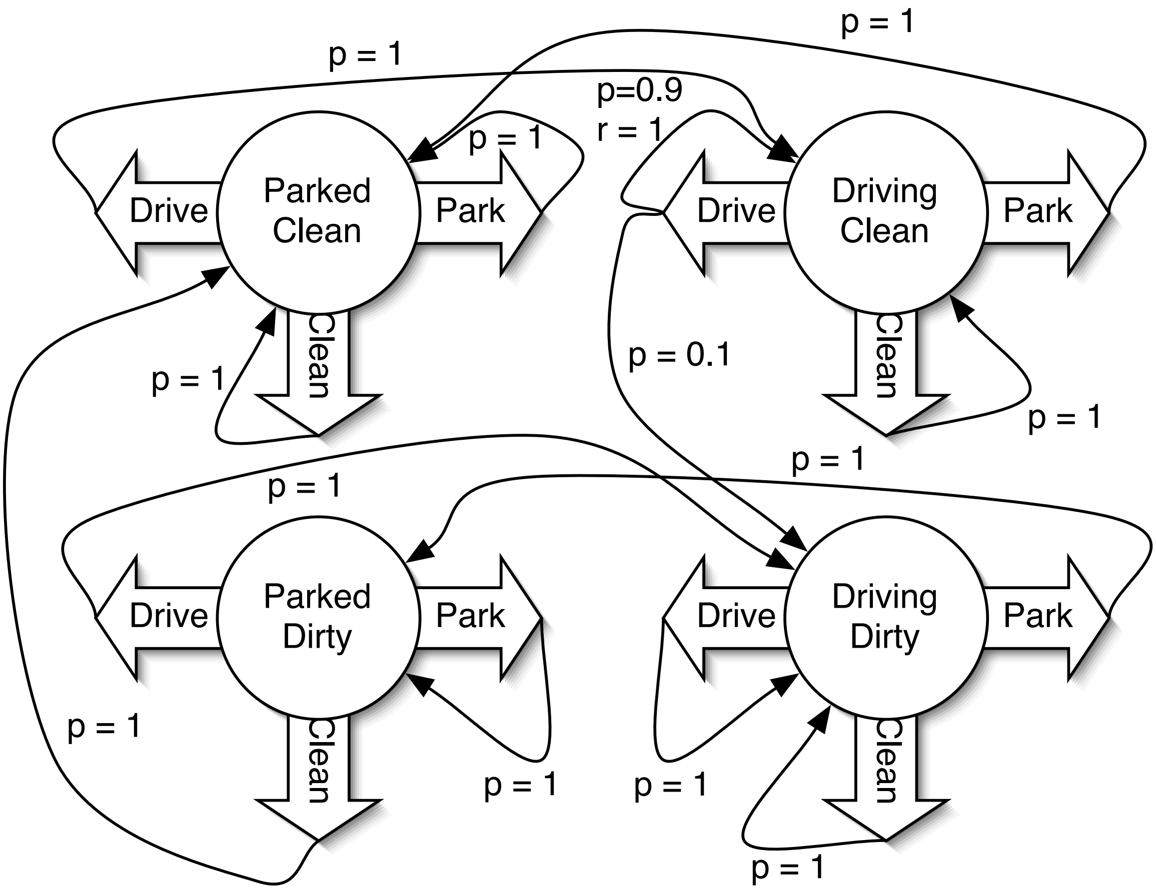
(c) (2 marks): In the lab you were given implementations of the Downhill Simplex and Conjugate Gradient optimisation methods. Imagine you were going to use one of them to optimise the AIBO's walk using the method described in class (i.e. minimise time to cross the field given a walk specified in a 24 dimensional space). Which method is likely to be more effective? Why?

(d) (2 marks): When discussing function minimisation, what does it mean for one direction to be conjugate to another?

Markov Decision Processes

Question 6.

You have the following MDP, with Q-values shown below. Note, the Q-values have not converged yet.



Rewards are 0 unless shown. Assume a discount factor of 0.9.

State	Action	Q-value
Parked Clean	Park	6.0
	Drive	4.0
	Clean	4.0
Parked Dirty	Park	6.0
	Drive	4.0
	Clean	4.0
Driving Clean	Park	6.0
	Drive	4.0
	Clean	4.0
Driving Dirty	Park	6.0
	Drive	4.0
	Clean	4.0

(a) (1 mark): Write the Bellman equations for model based Q-learning.

(b) (1 mark): If an agent starting with this Q-table, parks the dirty car it was driving and performs a single update based on its new state, how should the Q-values change?

(c) (1 mark): How does this change the value function?

(d) (6 marks): Assume that a prioritized sweeping system was initialised with the Q-values in table 1 and complete knowledge of the underlying transition and reward functions, that initially all priorities were 0, and that the agent then performed the one action above (parking a dirty car). What would the first three Q-function updates be?

(e) (3 marks): Which state-action pairs are left on the Priority Queue after the three updates, and what are their priorities?

Question 7.

(a) (4 marks): Give an example where Q-learning cannot solve a problem optimally because the problem is partially observable. Explain.

(b) (4 marks): Give an example of a partially observable domain where a simple Q-learner cannot solve the problem optimally it it only looks at the current observation, however if the Q-learner is learning its Q-function over the space of the current and previous observation then it can solve the problem optimally. Explain.

(c) (2 marks): For a fully observable domain, we use a policy that maps states to actions. For a partially observable domain, give two ways of defining a policy (e.g. a function from states to actions), each capable of representing an optimal policy.