

Family Name:

Other Names:

Signature:

Student Number:

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

Examination – Semester 2, November, 2006

COMP3431: Robot Architectures - EXAMPLE

Exam Duration: 3 hours

Total marks for this paper: 113

Reading Time: 15 minutes

This paper has 10 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.

Bayesian Tracking

Question 1.

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 (starting with a Venn diagram).
 $P(A|B) = ?$

(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 rest 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?

Architectures

Question 2.

- (a) (3 marks): List three different types of goal specification
- (b) (2 marks): List two different types of solution or plan specification

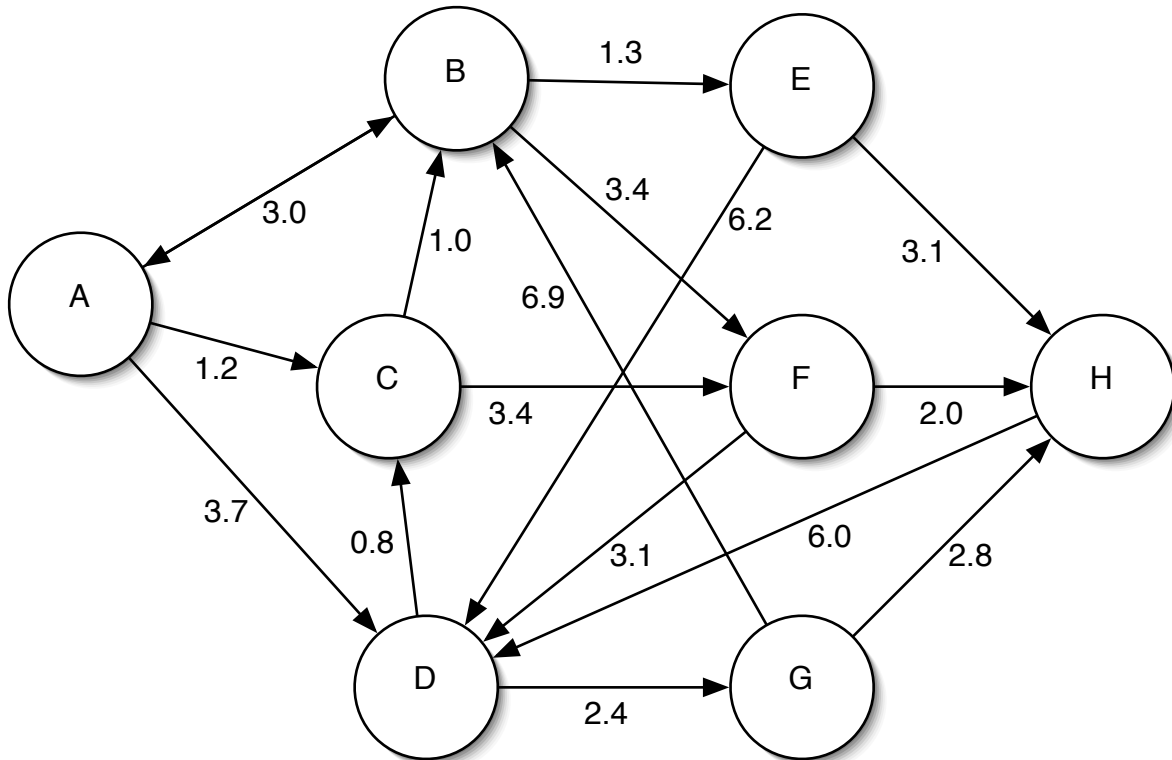
Question 3.

- (a) (2 marks): What is the defining characteristic of the lowest layer of a three layer robot architecture?
- (b) (2 marks): What is the defining characteristic of the middle layer of a three layer robot architecture?
- (c) (2 marks): What is the defining characteristic of the top layer of a three layer robot architecture?

Graph Search

Question 4.

All parts of this question refer to the following graph:



With the following heuristic distances to node H:

Node	Distance
A	6.0
B	4.0
C	5.0
D	4.5
E	3.0
F	2.0
G	2.5
H	0.0

(a) (8 marks): List the set of open nodes after each step of best first search starting from node A with a goal of node H.

(b) (8 marks): List the set of open nodes after each step of A* search starting from node A with a goal of node H.

Question 5.

(a) (3 marks): Dijkstra's algorithm for finding distances in a graph is the same as A* with a heuristic value of 0 at all nodes. Dijkstra's algorithm is known to give incorrect distances sometimes when there are edges with negative cost in the graph. Why does the optimality proof of A* not apply here?

(b) (3 marks): In normal A* search, the priority of a node, n , is: $f(n) = g(n) + h(n)$. If we add a constant coefficient (greater than one) to the heuristic, we get $f(n) = g(n) + \alpha h(n)$. If the optimal path cost is C^* , can you write a bound on the cost of the path found by our new search algorithm? If so, what is it?

(c) (3 marks): When might the modified A* in part B be a useful search algorithm?

(d) (4 marks): Imagine a keen young student invents a new search algorithm to compare with A*. Furthermore, this student has a proof that his algorithm expands fewer nodes than A* in some cases. Both algorithms are searching the same graph, and both have access to the same admissible heuristic function.

The student's proof shows that A* expands all nodes with $f(n) = g(n) + h(n) < C^*$, where C^* is the cost of the optimal path. He then shows that his algorithm sometimes manages to find a solution without expanding some nodes that have $f(n) < C^*$. In particular, the student shows an example where there is a node, q , where $f(q) < C^*$, and yet q is not expanded by his algorithm when it returns a path to the goal. In the example the student shows you, the path his algorithm finds is optimal.

Show that the student's algorithm may sometimes produce a non-optimal solution.

(e) (2 marks): In the design of a heuristic, what does it mean to *relax* a problem?

Planning

Question 6.

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))))
```

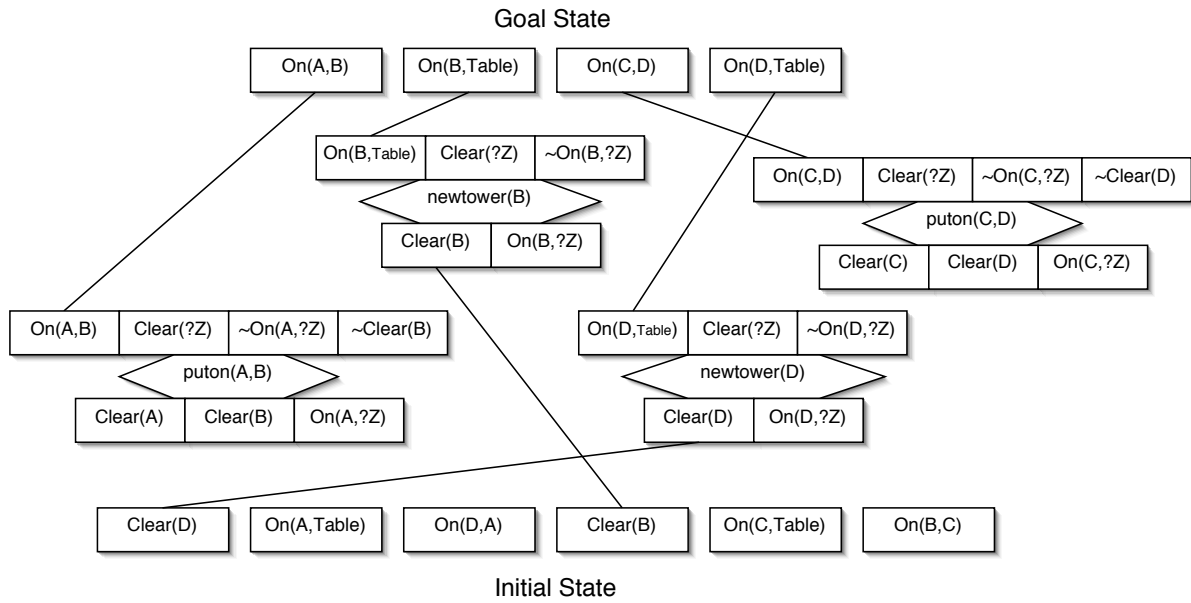
Given this start state:

```
(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 state space planner.
- (b) **(2 marks):** Give a possible first step for a forward chaining 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.



(f) (2 marks): Describe a common search heuristic for state space planners.

Continuous Search

Question 7.

(a) (3 marks): In a Rapidly exploring Randomised Tree search, describe the effect of changing the probability of going directly to the goal on the time required to solve a planning problem.

Question 8.

(a) (3 marks): A salesman comes to you with a search algorithm for optimizing 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?

Question 9.

You decide to try using a genetic algorithm for path planning through a simple two dimensional space with obstacles.

(a) (2 marks): Give an example evaluation function (or other method of comparing paths so determine which paths should survive to the next generation).

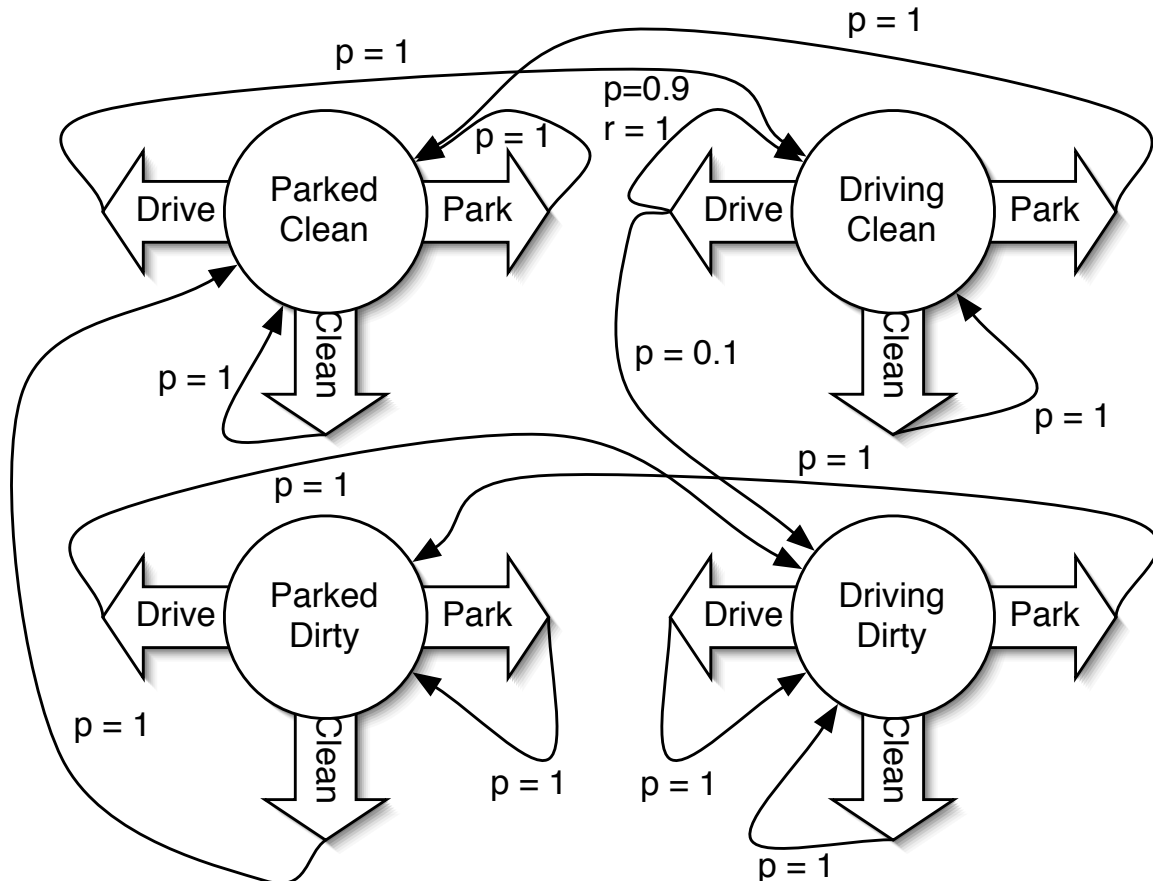
(b) (3 marks): Give an example mutation operator.

(c) (3 marks): Give an example cross-over operator.

Markov Decision Processes

Question 10.

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 11.

- (a) (4 marks): Give an example of there 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 if 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) (3 marks): Is there any relation between the value function of a reinforcement learning system and the heuristic function for an A* search algorithm? Explain.