UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1997

MSc Degree in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the
Diploma of Membership of Imperial College

PAPER A4.94

INTRODUCTION TO ARTIFICIAL INTELLIGENCE
Tuesday, April 22nd 1997, 2.00 - 4.00

Answer THREE questions

For admin. only: paper contains 4 questions

The following questions are about four different ways of representing the robot path finding problem: A robot is in a building and has the problem of finding a path from one room to another. The layout of the building is described by a predicate

connected(room$_1$, door, room$_2$)

which holds when *room$_1$* is directly connected by *door* to *room$_2$*. The predicate *connected* is not necessarily symmetric in its first and third arguments, i.e. doors may be one way.

The state of the building can be changed by two actions

open(door) or *move(room$_1$, door, room$_2$).*

The action *open(door)* is possible if the robot is in a room which is directly connected by *door* to another room and *door* is closed. The result of the action is that *door* changes from *closed* to *open.*

The action *move(room$_1$, door, room$_2$)* is possible if the robot is in *room$_1$* and *room$_1$* is directly connected to *room$_2$* by *door* and *door* is *open.* The result of the action is that the location of the robot changes from *room$_1$* to *room$_2$*.

1a Give a set of non-recursive condition-action rules which will enable the robot to explore the building, repeatedly moving from one room to another either by moving directly through a door which is already open or by first opening a door which is closed. The rules should be formulated in such a way that all conditions, except for those of the form *connected(room$_1$, door, room$_2$)* are verified simply by matching them against observations made in the environment. For this purpose, you should assume that the robot can observe what room it is in and the status of any doors directly connecting that room to other rooms.

b Show how a condition-action rule interpreter might execute the condition-action rules of part a, given a building described by

connected(r_1, d_1, r_2)
connected(r_1, d_2, r_3)
connected(r_2, d_3, r_4)
connected(r_3, d_4, r_4)

and an initial state in which the robot is in room r_1, doors d_1 and d_4 are open and d_2 and d_3 are closed. Assume all actions are executed successfully.

c Modify the rules of part a so that the robot need not have any facts of the form *connected(room$_1$, door, room$_2$)* in its knowledge base. For this purpose you may need to change the form of the actions the robot can perform.

d Suppose that the goal of going to a particular room, *room*, is represented by a fact *go(room)* in the robot's knowledge base. Modify the rules of part a so they are "goal-oriented", in the sense that the robot will stop exploring the building when it has reached the goal *room.*

e What are the advantages and disadvantages of the condition-action rule approach to path-finding compared with heuristic search or planning.

The five parts are of equal weight.

2 a Formulate the robot path finding problem described above in the situation calculus.

b Now consider a modification of the robot path finding problem in which the robot cannot open closed doors, and therefore the only action the robot can perform is *move(room$_1$, door, room$_2$)* described above. Assume that the building layout is described by:

holds(connected(r_1, d_1, r_2*), S)*
holds(connected(r_1, d_2, r_3*), S)*
holds(connected(r_2, d_3, r_4*), S)*
holds(connected(r_3, d_4, r_4*), S)*
holds(open(d_2*), S)*
holds(open(d_4*), S)*
holds(closed(d_1*), S)*
holds(closed(d_3*), S)*

Use goal-reduction, employing the situation calculus formulation of part a (without the action of opening closed doors), to solve the goal of the robot being in room r_4, given an initial state in which the robot is in room r_1. Give only a proof and not the whole search space.

HINT: You may wish first to simplify the formulation of part a (without the action of opening closed doors) by using partial evaluation.

The two parts are of equal weight.

3 a Describe informally and in general terms how an expert system approach to problem solving differs from problem solving from "first principles".

b Now consider a modification of the robot path finding problem in which doors connect spaces, which can be rooms, hallways or stairwells. Assume also that doors are two-way (symmetric) connections.

Describe informally the top-most level of an expert system for going from one room in a building to another room in the same building. You might find it useful to use such concepts as whether a space is a room, hallway or stairwell and what floor a room or hallway is on.

You may assume that

every room and every hallway is on only one floor,
every room is directly connected to a hallway,
all doors are open,
for every stairwell and every floor, there is some hallway on the floor to which the stairwell is directly connected.

You may also assume that there already are methods available for

going from any space to any other space directly connected to it by a door and
going from any hallway to any other hallway on the same floor,

c Formalise the top-level expert system of part b. You may use any formalism which is convenient.

The three parts carry, respectively, 30%, 30% and 40% of the marks.

Turn over...

4 a i What is the *admissibility* property of a search algorithm for finding a least cost solution path in a problem solving search graph?

ii What value is associated with each node N on a search graph by the A* graph search algorithm? How is this value computed? What node does A* select for expansion?

iii What are the two conditions on the use of the A* algortihm that guarantee admissibility?

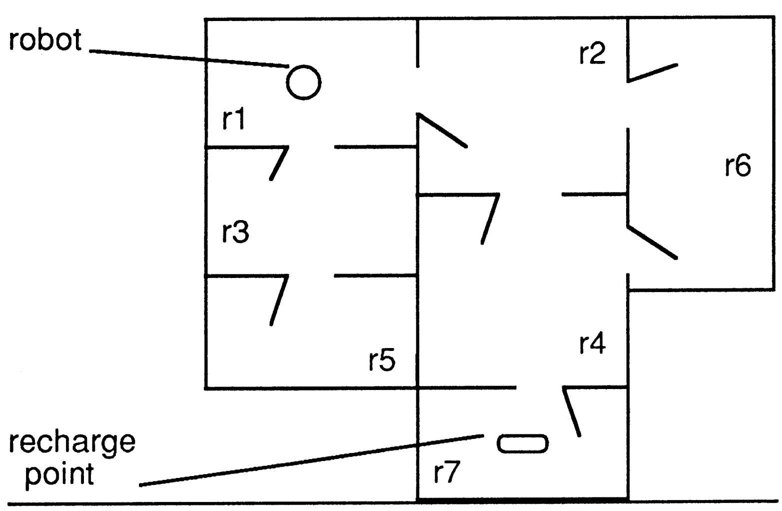
b Suppose a robot has to get to a room that has the recharge point in it and that it decides to find the minimum cost route from its initial position to the recharge point, before it starts out, by using the A* algorithm. It has access to data telling it in which room it is initially located, in which room the power supply is located, and which room is directly connected to which via a door. It also has access to two data tables, one giving the actual distance between the mid-points of two directly connected rooms, the other giving the straight line distance from the center of any room to the recharge point. The straight line distance may go through one or more walls.

Assume also that doors are always open and are two-way (symmetric) connections.

Suppose that each move that the robot makes from one room to another by going through a door has an associated cost that is the distance between the mid-points of the two rooms.

i What heuristic function would you use to apply the A* algorithm to this search problem? Justify your choice.

ii For the room layout and initial state represented by the picture:



draw the search graph of all the rooms that the robot might be in after *any* three actions of moving from one room to another with which it has a direct connection.

iii Assume that the distance between the mid-points of each pair of directly connected rooms is the same, namely 1, and the table giving the straight line distance from the mid-point of each room to the power point is:

Room	Straight Line Distance
r1	2.7
r2	2.0
r3	1.8
r4	1.0
r5	1.1
r6	1.9

Separately draw the sub-tree of this graph that will be generated by application of the A* algorithm using your heuristic function. On the search tree label every arc connecting one state to another with the number of the step in the A* algorithm at which the arc is added to the tree. Thus all the arcs descending from the initial state r1 will be numbered 1. Assume that A* checks for repeated states and prunes them from the search tree. Attach to each state on your search tree the value of the state as computed by the A* algorithm.

The two parts of this question carry respectively 40%, 60% of the marks.

End of paper

