UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2000

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

# PAPER A494

# ARTIFICIAL INTELLIGENCE

Friday 12 May 2000, 14:30
Duration: 120 minutes

*Answer THREE questions*

Paper contains 4 questions

1 a i) What is meant by saying that a search algorithm for finding a minimum cost solution path in a problem solving search graph is *admissible*?

ii) What value is associated with each node N of the exploration of a search graph by the A* graph search algorithm? What node does A* select for expansion at each step in the construction of its search tree.

iii) What are the two conditions on the use of the A* algorithm that guarantee admissibility? Using one of these conditions, prove that any solution path found by A* will be a minimum cost path.

b The following is a simplified generic search program written in Prolog. It is simplified because at each step in the search it only retains a list of the leaf nodes of the part of the search graph that it has constructed up to that point. Each leaf node is just a problem state reachable from the start state by some sequence of operations. The program does not retain information about the path from the start node to each leaf, nor does it check for repeated states. The program is invoked with a query of the form:

search([**start**],Goal)

where **start** is the start state for the search. It will terminate, if at all, with Goal bound to a goal state that is reachable from **start**.

```
search(LeafNodes,GNode) :-
        choose(GNode,LeafNodes,RemLeafNodes),
        goal(GNode).
search(LeafNodes,GNode) :-
        choose(Node,LeafNodes,RemLeafNodes),
        next_nodes(Node,NextNodes),
        add_to_leaves(NextNodes,RemLeafNodes,NewLeafNodes),
        search(NewLeafNodes,GNode).
```

Assuming that the search problem is defined by a set of clauses for the relation:

a_next_state(State,NextState): NextState is a state reachable by the application of one operation from State

Give:

i) a suitable Prolog definition for next_nodes/2.

and give suitable definitions for choose/3 and add_to_leaves/3 so that the general program implements:

ii) depth first search
iii) breadth first search

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

2 a Briefly compare the heuristic search approach to problem solving with the planning approach using STRIPS style action descriptions.

b What is the STRIPS `divide and conquer' planning algorithm? What is its major drawback?

c A multi-armed robot cook needs to plan its cooking.

i) Give the STRIPS style action descriptions for the following robot actions and their effects:

Cooking an omelette requires that the robot have eggs, a frying pan and butter and be at the cooker. The action results in the robot having a plain omelette but no longer having the eggs or butter. It also results in the frying pan being dirty.

The robot can obtain any item from a storage cabinet by the action of taking it from the cabinet, providing it is at the cabinet.

The robot can move from any kitchen location to another kitchen location providing the path is clear between the two.

ii) Give a set of facts that would be a suitable initial description of a robot's world that will enable the robot to construct a plan for cooking an omelette. Assume that the robot knows that eggs and butter are kept in the refrigerator, that a frying pan is kept in the cupboard, that the cooker, the refrigerator and the cupboard are storage cabinets and kitchen locations. Also assume that the robot is initially at the sink, which is a kitchen location, and that there is a clear path between the sink and the cupboard, between the cupboard and the refrigerator, and between the refrigerator and the cooker. What are the fluent predicates? What is the goal state for such a plan?

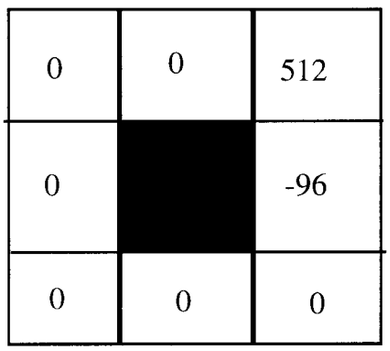
iii) Give the sequence of actions of such a plan. You do not need to show how it might be generated.

*The three parts carry respectively 20%,20%,60% of the marks.*

3 a Briefly describe the following *frame knowledge representation* concepts, and for each one mention a related concept of object oriented programming.

i) a frame definition
ii) a frame instance
iii) an inherited value for a frame slot

b Consider an agent moving in a deterministic grid world with the following rewards:




The black square in the centre is inaccessible to the agent. From the upper right-hand corner, there is only one possible action, the *halt* action which terminates the agent and gives the reward 512. From every other state (including the -96 state), there are two actions available, which allow the agent to move to either of the two adjacent states.

i) Using the formula:

$$U(i) = R(i) + \gamma \max_a U(\delta(i, a))$$

(where $R(i)$ is the reward associated with state $i$, $\delta(i, a)$ is the transition function giving the state reached after performing action $a$ in state $i$, and $\gamma$ is the discount factor), calculate the utility $U$ for each state in the grid world and show these utilities with a diagram. Take $\gamma = 0.5$.

ii) Using the formula:

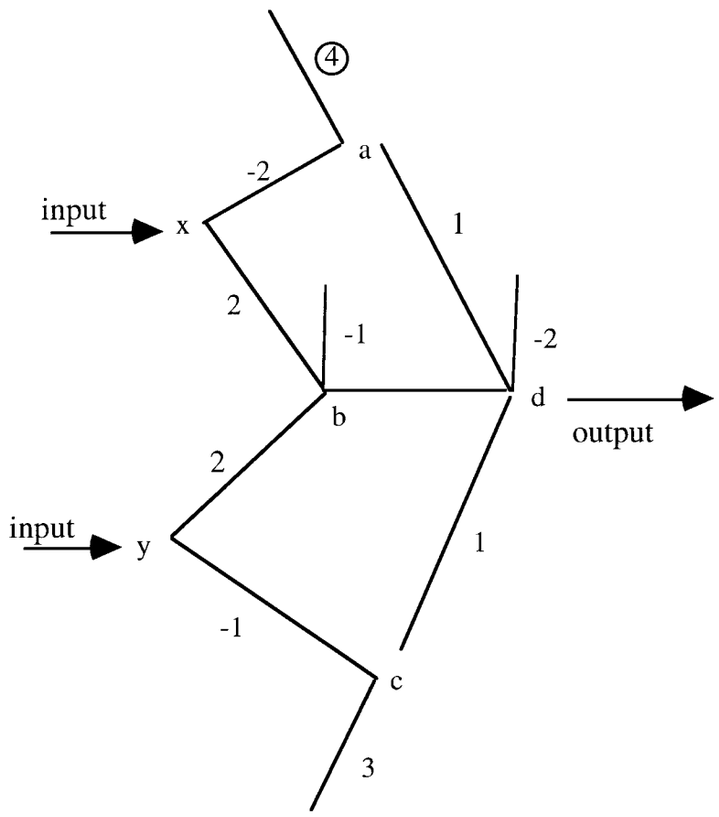
$$Q(a,i) = R(i) + \gamma \max_{a'} Q(a',j) \qquad \text{where } j = \delta(i, a')$$

draw a diagram showing the Q-values for each possible action from each state in the grid world. Again take $\gamma = 0.5$.

iii) Draw the optimal policy for the agent.

*The two parts carry respectively 30% and 70% of the marks*

4 a Consider the following neural network:




The nodes labelled $x$ and $y$ are inputs into the network, and the short steep lines represent constant offsets. The nodes $a$, $b$, $c$, $d$ represent perceptron units using the step activation function:

$$f(net) = \begin{array}{l} 1 \text{ if } net > 0 \\ 0 \text{ otherwise} \end{array}$$

i) What is the output of the network given the inputs $x$=1, $y$=2?

ii) Draw a sketch of the $xy$ plane showing the region of inputs where the network would produce an output of 1.

iii) Taking a Boolean logic interpretation in which 0=*false* and 1=*true*, what logical expression can the network calculate on inputs x,y $\in$ {0,1}?

iv) If the circled weight were changed to 1, what expression would the new network calculate?

b The version space learning algorithm can be used to induce a concept description for consumers who miss credit card payments. As in lecture, the target description language is an attribute-valued logic containing expressions which are conjunctions of attribute assignments. If an attribute is not listed, it is allowed to take on any value. Some example expressions:

$attribute_2 = value_1$

$attribute_2 = value_1 \wedge attribute_3 = value_5$

*true*

For convenience, we write these expressions in an abbreviated tuple notation as:

(?,$value_1$, ?, ?, ?)
(?,$value_1$, $value_5$, ?, ?)
(?, ?, ?, ?, ?)

Disjunctions between two expressions, or between two values for the same attribute, are not allowed.

The attributes and their permitted values are as follows:

| | |
|---|---|
| age | <30, 30-39, 40-49, 50+ |
| sex | M, F |
| income | low, middle, high |
| occupation | management, research, service, clerical, manufacturing |
| status | single, married, divorced, cohabiting |

i) Using the tuple notation, write an expression for the class of consumers who are single female managers.

ii) Give the set of hypotheses in this language which are the immediate (i.e., can be reached in one step) generalisations of the expression in part i).

iii) Give the set of hypotheses which are the immediate specialisations of the expression in part i).

iv) Suppose that after a few iterations of the version space algorithm we obtain the sets:

G = {(<30, ?, ?, ?, ?), (?, ?, low, ?, ?), (?, ?, ?, research, ?)}
S = {(<30, ?, low, research, single)}

After the positive example (<30, F, low, manufacturing, single) is presented, which consistent hypotheses remain in the version space?

*The two parts of this question carry equal marks*