CS486

OLLSCOILL NA hÉIREANN
THE NATIONAL UNIVERSITY OF IRELAND
COLÁISTE NA hOLLSCOILLE, CORCAIGH
UNIVERSITY COLLEGE, CORK

**Sample Examination**

B.Sc. Honours in Computer Science 1998

Knowledge Based Systems

Time allowed: **Two hours**

Answer **all** questions.

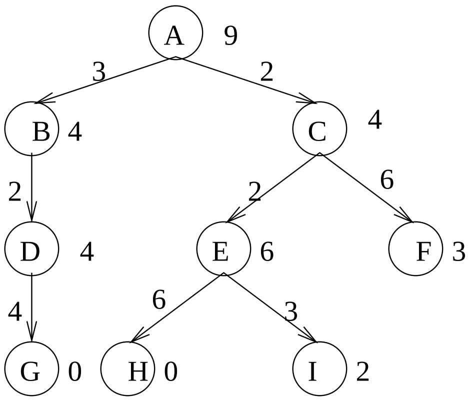
1 (10 marks)

(i) [2 marks]

(a) [1 mark] Give one advantage of depth-first search over breadth-first search.

(b) [1 mark] Give one advantage of iterative-deepening search over depth-first search.

(ii) [5 marks] Consider the following state space in which the states are shown as nodes labeled A through I. A is the initial state, and G and H are the goal states. The numbers alongside the edges represent the costs of moving between the states. To the right of every state is the estimated cost of the path from the state to the nearest goal.




Show how each of the following search strategies finds a solution in this state space by writing down, in order, the names of the nodes removed from the agenda. Assume the search halts when a goal state is removed from the agenda. (In some cases, multiple answers are possible. You need give only one such answer in each case.)

(a) Breadth-first;

(b) Depth-first;

(c) Iterative-deepening;

(d) Least-cost search; and

(e) Heuristic search using $f(n) = g(n) + h(n)$ as the heuristic function, where $g(n)$ is the cost of the path to node $n$, and $h(n)$ is the estimated cost of the path from node $n$ to the nearest goal.

(iii) [3 marks] What does it mean for an heuristic function to be *admissible*?

Is $h(n)$ of part (ii) admissible?

Hence, what do we know about the search strategy of part (ii)(e)?

2 (10 marks)

This question, and question 3, concern a robot which must collect a dangerous chemical and place it in a cupboard. The question uses the following 'key':

| Predicate symbols | |
|---|---|
| *LOCNS(w,x,y,z)* | In state $z$, the location of the objects are: the robot is at $w$, the chemical is at $x$ and the cupboard is at $y$. |
| *SAFE(z)* | In state $z$, the chemical is safely stored away. |

| Function symbols | |
|---|---|
| *move(x,y,z)* | *move* denotes a function that returns the new world state attained if the world is in state $z$ and the robot moves from $x$ to $y$. |
| *carry(x,y,z)* | *carry* denotes a function that returns the new world state attained if the world is in state $z$ and the robot moves from $x$ to $y$ carrying the chemical with it. |
| *stash(z)* | *stash* denotes a function that returns the new world state attained if the world is in state $z$ and the robot unlocks the cupboard and places the chemical inside. |

| Constant symbols | |
|---|---|
| $a$ | the initial location of the robot |
| $b$ | the initial location of the chemical |
| $s_0$ | the initial world state |

$v, w, x, y$ and $z$ (possibly subscripted) are used for variables.

You will use the following axioms (already in clausal form):

1. $\neg LOCNS(v_1, x_1, y_1, z_1) \vee LOCNS(w_1, x_1, y_1, move(v_1, w_1, z_1))$
2. $\neg LOCNS(w_2, w_2, y_2, z_2) \vee LOCNS(x_2, x_2, y_2, carry(w_2, x_2, z_2))$
3. $\neg LOCNS(y_3, y_3, y_3, z_3) \vee SAFE(stash(z_3))$
4. $LOCNS(a, b, c, s_0)$

Use *resolution refutation* theorem-proving to answer the question: how can we reach a safe state, i.e. in first-order predicate calculus:

$$\exists z SAFE(z)$$

Show your working and present your proof in the form of a *refutation tree.*

(**Hint:** Among the changes you will make to the goal is the introduction of a literal $ANS(z)$.)

3 (10 marks)

Question 2 showed that, if actions, states and goals are suitably axiomatised, resolution-refutation theorem-proving can be used to solve planning problems. Call such a planning system $P$.

(i) [2 marks] Explain the difference between *regression planning* and *progression planning.*

Explain **precisely** how we can ensure that $P$ is a regression planner.

(ii) [3 marks] Explain the difference between planners that build *linear plans* (also called *totally-ordered plans*) and those that build *non-linear plans* (also called *partially-ordered plans*).

Explain **precisely** why the representation that $P$ uses for its plans (i.e. first-order terms) means that $P$ cannot easily be changed to give a non-linear planner.

(iii) [3 marks] Using symbols that are *similar* to those used in question 2, translate the four axioms in question 2 into the STRIPS language.

(**Hint:** States are explicit arguments in question 2 but will be implicit in the STRIPS language representation.)

(iv) [2 marks] When axiomatising actions for a planning system, what is meant by the *qualification problem*? Give an example for the domain described in question 2.

4 (10 marks)

A taxonomy of objects, classes and their properties has been represented in logic as follows:

$$\begin{array}{l}
\forall x(seat(x) \rightarrow furniture(x)) \\
\forall x(table(x) \rightarrow furniture(x)) \\
\forall x(chair(x) \rightarrow seat(x)) \\
\forall x(sofa(x) \rightarrow seat(x)) \\
\forall x(seat(x) \rightarrow hasLegs(x, 4)) \\
\forall x(chair(x) \rightarrow hasPlaces(x, 1)) \\
\forall x(sofa(x) \rightarrow hasPlaces(x, 2)) \\
chair(myChair) \\
sofa(myBlueSofa) \\
sofa(myGreenSofa)
\end{array}$$

where the predicate symbols $furniture, seat, table, chair$ and $sofa$ denote classes, the predicate symbols $hasLegs$ and $hasPlaces$ denote properties, and the constant symbols denote objects.

(i) [2 marks] Show this diagrammatically as a *semantic net*. Include a key to the symbols used in your diagram.

(ii) [4 marks] Give an alternative axiomatisation in logic in which classes and properties have been *reified* into objects. Include axioms that give the effect of inheritance so that we can derive from your new axiomatisation formulae that correspond to those that can be derived from the axiomatisation above.

(iii) [4 marks] Stool is a subclass of seat but stools have only 3 legs. I have two stools, one blue one and one green one, but the green one is broken: it has only two legs.

Revise **one or other** of the axiomatisations used in this question to handle this information about stools and the over-riding of inheritance in general.

5 (10 marks)

A rule-based expert system has been developed to assist in the packing of household possessions when moving home. There are 10 rules as follows:

```
R1:  IF   Step is box-fragile-items AND
          There is a fragile item to box AND
          (There is an empty box OR
           There is a box containing only fragile items) AND
          The fragile item hasn't been padded
     THEN Pad the fragile item

R2:  IF   Step is box-fragile-items AND
          There is a fragile item to box AND
          (There is an empty box OR
           There is a box containing only fragile items) AND
          The fragile item has been padded AND
          The box's free space >= the item's size
     THEN Put this item into the box AND
          Decrease the box's free space appropriately

R3:  IF   Step is box-fragile-items AND
          There is a fragile item to box
     THEN Start a fresh box

R4:  IF   Step is box-fragile-items
     THEN Discontinue box-fragile-items AND
          Start box-heavy-items

R5:  IF   Step is box-heavy-items AND
          There is a heavy item to box AND
          (There is an empty box OR
           There is a box with no fragile items) AND
          The box's free space >= the item's size
     THEN Put the item into the box AND
          Decrease the box's free space appropriately

R6:  IF   Step is box-heavy-items AND
          There is a heavy item to box
     THEN Start a fresh box
```

```
R7:  IF   Step is box-heavy-items
     THEN Discontinue box-heavy-items AND
          Start box-light-items

R8:  IF   Step is box-light-items AND
          There is a light item to box AND
          (There is an empty box OR
           There is a box with no fragile items) AND
          The box's free space >= the item's size
     THEN Put the item into the box AND
          Decrease the box's free space appropriately

R9:  IF   Step is box-light-items AND
          There is a light item to box
     THEN start a fresh box

R10: IF   Step is box-light-items
     THEN Discontinue box-light-items AND
          Halt
```

(i) [1 mark] Explain whether the problem being solved here is primarily an *analysis* task or primarily a *synthesis* task.

(ii) [3 marks] The reasoning strategy in this task is to pad fragile items, then to box them, then to box heavy items and finally to box light items. The system will behave in this way only if the inference engine works in a particular way.

Explain how the inference engine must work to get this behaviour.

(iii) [3 marks] One way to make the reasoning strategy mentioned in part (ii) more explicit would be the use of *meta-rules*.

Explain what these are and how they would be used in this situation.

(iv) [3 marks] Another way to make the reasoning strategy more explicit would be the use of *task hierarchies*.

Explain what these are and how they would be used in this situation.

$$$$