IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2014

EEE/EIE PART III/IV: MEng, Beng and ACGI

ARTIFICIAL INTELLIGENCE

Corrected Copy

Thursday, 16 January 2:30 pm

Time allowed: 3:00 hours

There are SIX questions on this paper.

Answer FOUR questions.

All questions carry equal marks

Any special instructions for invigilators and information for candidates are on page 1.

Examiners responsible First Marker(s) : J.V. Pitt

Second Marker(s) : T-K. Kim

The Questions

1 a) Briefly describe the operation of the depth first, depth-limited depth first, and iterative deepening depth first search algorithms, and compare and contrast them with respect to appropriate criteria.

[6]

b) Write a Prolog program which, given a list, computes (on backtracking) all the different pairs in the list, as a 2-tuple, i.e.:

```
?- select2( [a,b,c,d], Pair ).
Pair = (a, b) ;
Pair = (a, c) ;
Pair = (a, d) ;
Pair = (b, c) ;
Pair = (b, d) ;
Pair = (c, d) ;
false.
```

You may assume that the list contains at least two elements.

[4]

c) Four people come to a river in the night. There is a narrow bridge, but it can only hold two people at a time. They have one torch and, because it's night, the torch has to be used when crossing the bridge. Person A can cross the bridge in one minute, B in two minutes, C in five minutes, and D in eight minutes. When two people cross the bridge together, they must move at the slower person's pace. However, the torch battery only has 15 minutes of life.

Formulate the problem, in Prolog or similar declarative notation, so that the General Graph Search (GGS) program can be used to find a plan whereby all four people can cross the river before the torch battery runs out.

You are required to specify:

i) The state representation;

ii) The start state;

iii) The goal state; and

iv) The state-change rules.

A solution to the problem is not required.

[10]

2 a) Briefly describe the operation of the uniform cost, best first and A* search algorithms, and compare and contrast them with respect to appropriate criteria.

[6]

b) Consider an explicit definition of a graph $G=\langle N, E, R\rangle$, where N is the set of nodes, E is the set of edges, and R is the incidence relation; and an implicit definition of a graph $G'=\langle S, Op\rangle$, where S is a node and Op is a set of operators (partial functions).

i) From an identified start node $S \in N$, give an inductive definition of the paths P_G in the graph G.

ii) Give an inductive definition of the paths P_G' defined by graph G'.

iii) State what requirement must be satisfied for $P_G = P_G'$.

[6]

c) Explain how the General Graph Search (GGS) program, using the uniform cost search algorithm, selects the paths in P_G' to search.

[4]

d) Given several heuristics for a problem-solving search using the A* algorithm, explain why, and under what conditions, one heuristic may be more 'efficient' than the others.

[4]

3. Consider a robot situated in a grid-like maze.

Assume that the maze (environment) is completely *unknown* to the robot, and that it has a requirement to traverse the maze from its current location, and find any goal locations that it must discover by exploring the maze.

a) Describe an algorithm for mapping the maze and learning the appropriate behaviour in a particular location.

In particular explain, with examples:

i) What data structures would be used;

ii) What exploration strategy would be used for path finding;

iii) What exploitation strategy would be used for path following;

iv) What strategy would be used for path learning.

[10]

b) Suppose that the robot *is not aware* of changes in the environment and must discover them by exploration. Explain how the algorithm of Part (a) could operate under these circumstances.

Suppose that the robot *is aware* of changes in the environment as they occur. Discuss the limitations of planning algorithms of this kind with respect to the rate of change of the environment.

[6]

c) Explain the differences between the MiniMax and AlphaBeta search algorithms for 2-player games.

[4]

4 a) In the context of automated reasoning, define the term *resolution*, specify the resolution rule, and explain its relevance to logic programming.

[4]

b) In the context of automated reasoning, define the term *unification*, specify a unification algorithm, and explain its relevance to logic programming.

[4]

c) Consider the following statements:

Anything with a long-nose can smell-better other things.

Anything with big-eyes can see-better other things.

If something can smell-better and see-better something else, then the first thing chases the second thing.

If something chases something else and the first thing runs-faster than the second-thing, then the first thing catches the second thing.

If a wolf catches a little-girl, then it eats her.

Express these five arguments as formulas of First Order Predicate Logic, and translate them into clausal form.

[6]

d) One unfortunate incident concerns the following facts:

Big-Bad-Woof is a wolf.
Big-Bad-Woof has a long-nose.
Big-Bad-Woof has big-eyes.
Red-Riding-Hood is a little-girl.
Big-Bad-Woof runs-faster than Red-Riding-Hood.

Express these facts in clausal form.

Using proof-by-refutation and the resolution inference rule (and showing the unifiers), prove that: *Big-Bad-Woof eats Red-Riding-Hood.*

[6]

5 a) Specify the rules of the KE calculus for the $\wedge, \vee, \rightarrow$ and $\leftrightarrow$ connectives.

Explain the PB rule.

Explain the closure rule.

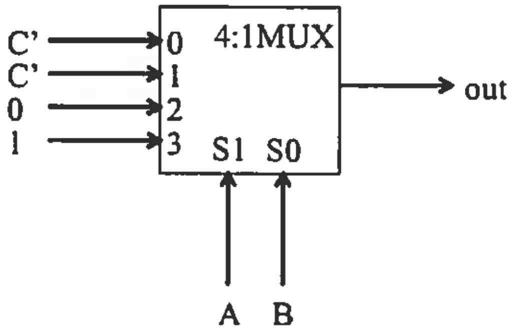
Explain the beta-simplification rule.

Explain why the restrictions on the closure rule often used in implementations of the KE calculus do not affect soundness and completeness.

[8]

b) A multiplexer can be used to implement random logic functions.

Consider the multiplexer shown in Figure 5.1.



A	B	C	out	
0	0	0	1	C'
0	0	1	0	
0	1	0	1	C'
0	1	1	0	
1	0	0	0	0
1	0	1	0	
1	1	0	1	1
1	1	1	1	

Figure 5.1: Multiplexer implementation of random logic function

Prove, using the KE proof procedure, that: $out \leftrightarrow ((A \wedge B) \vee (\neg A \wedge \neg C))$.

It is essential that the KE-tree is properly annotated to show the application of the rules in Part (a).

[12]

6 a) Define the syntax for well-formed formulas of propositional modal logic. Explain how a Kripke model is used to give a semantics for well-formed formulas of propositional modal logic.

[4]

b) Given the Kripke model M=<$\{a\}$, $\{\}$, $[[p]]=\{a\}$>, explain whether the formulas $\Box p$ and $\Diamond p$ are true or false at world a in model M.

[4]

c) Show that the axiom schema **D** does not hold in the class of all models. Show that the axiom schema **D** does hold in the class of all models that are serial.

[4]

d) Prove, using the KE calculus for propositional modal logic, that the following formula is not a theorem of modal logic **K**, but is a theorem of **S5**:

$$\Box p \rightarrow \Diamond p$$

Explain why it is not a theorem of **K**.

[4]

e) Demonstrate, using the KE calculus for propositional modal logic, that the axiom schema:

$$(p \leftrightarrow q) \rightarrow (\Box p \leftrightarrow \Box q)$$

is not provable in the modal logic **K**.

Construct a Kripke model to support the demonstration.

[4]