UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

B. Sc. Examination 2011

Computing

IS53024A Artificial Intelligence

Duration: 2 hours and 15 minutes

Date and time:

There are five questions in this paper. You should answer no more than three questions. Full marks will be awarded for complete answers to a total of three questions. Each question carries 25 marks. The marks for each part of a question are indicated at the end of the part in [.] *brackets.*

There are 75 marks available on this paper.

Electronic calculators must not be programmed prior to the examination. Calculators which display graphics, text or algebraic equations are not allowed.

THIS PAPER MUST NOT BE REMOVED
FROM THE EXAMINATION ROOM

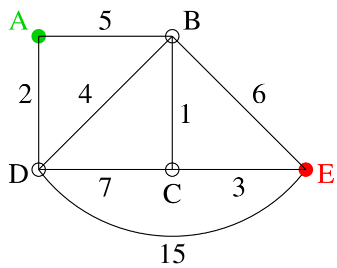
Question 1

(a) Explain the four common measures for performance evaluation of a search algorithm in the context of Artificial Intelligence. Describe the question that is addressed by each of the four measures. [4]

(b) Consider the two named search algorithms and the simple graph below. Assume that, starting from vertex A, an agent searches for a shortest path from A to E. Demonstrate how each of the two search algorithms work by tracing the vertices in order of being visited. Show all your work. Outline each search algorithm in your own words and comment on the completeness and optimality. [12]

i. The Breadth-First Search algorithm

ii. The A* Search algorithm. Assume the straight-line distance to E are from A:10, from B:3, from C:3 and from D:9.



(c) Consider the trivial agent program TABLE-DRIVEN-AGENT below. Describe briefly how this table-driven program may be used. Explain, with the aid of an example, why the table-driven approach for agent construction tends to be unsuccessful. Give a quantitative justification to support each of your arguments. [9]

function TABLE-DRIVEN-AGENT (percept)
percepts: a sequence of percepts, initially empty;
actions: a table of actions, initially fully specified.

```
append percept to the end of percepts
action ← LOOKUP(percepts, table)
return action
```

Question 2

Consider a simple game for two players A and B. Initially, player A is at location 1, and player B is at location 4, as shown in the figure below.

1	2	3	4
A			B

Player A moves first. Players A and B then move alternately, one at a time. A player must move his piece to an open adjacent space in either left or right direction (Passes are not permitted). If the opponent occupies an adjacent space, then a player may jump over the opponent to the next open space. The game ends when one player reaches the opposite end of the array. That is, if A reaches location 4 first, then the score of the game to player A is +1. If B reaches location 1 first, then the score of the game to player A is -1.

(a) Draw a game tree for the game, and highlight the redundant nodes using round brackets "()". [8]

(b) Draw a state graph for the game. [6]

(c) Conduct a graph search on the state graph derived in part (b) using the GRAPH-SEARCH algorithm. Demonstrate the separation property of the algorithm with the aid of a diagram. [5]

(d) Explain and demonstrate how the Minimax algorithm can be applied to find the optimal solution. [6]

Question 3

(a) Name three main uninformed search techniques used in the field of artificial intelligence. [3]

(b) Explain briefly the most common rule-based system modes of operation: stimulus-driven and goal directed, giving also their alternative names. [6]

(c) What are the four shortcomings in the conventional programming technology addressed by the rule-based systems? [8]

(d) Draw a graphical representation of a sophisticated rule-based system. Give the name of each building block in such a system, as well as the input and output data for each building block. [8]

Question 4

(a) i. What are the four main elements of a search-based planner? Explain each of them briefly. [8]

ii. What descriptive means should a language for planning operators provide? [5]

(b) Plan the actions of a person who needs to have a hot drink using a rule-based inference engine operating with forward chaining. Develop the plan by interpreting the performance of the engine showing the sequence of rule firings as well as the variable bindings. Usethe following rules: [12]

```
(RULE 1 "Want a hot drink"
    (IF   ( ?x at home ) & ( ?x needs hot drink ) & ( ?x has tea ))
    (THEN (DELETE ( ?x needs hot drink )) & (ADD ( ?x makes tea ))))

(RULE 2 "Go to shop for tea"
    (IF   ( ?x arrives at shop ) & ( ?x has credit card ))
    (THEN (DELETE ( ?x arrives at shop )) & (ADD ( ?x looks for tea ))))

(RULE 3 "Take the bus"
    (IF   ( ?x has coins ) & ( ?x has energy ) & ( ?x close to bus ))
    (THEN (ADD ( ?x takes bus to shop ))))

(RULE 4 "Purchase tea"
    (IF   ( ?x looks for tea ) & ( shop has tea ))
    (THEN (DELETE ( ?x looks for tea )) & (ADD ( ?x buys tea ))))

(RULE 5 "Enter shop"
    (IF   ( ?x takes bus to shop ))
    (THEN (ADD ( ?x arrives at shop ))))

(RULE 6 "Tea available"
    (IF   ( ?x buys tea ))
    (THEN (ADD ( ?x has tea ))))
```

Assume that the working memory is initialized with the following facts:

```
WM=[ ( John at home )( John has credit card )( shop has tea )
      ( John has coins )( John has energy )( John close to bus )
      ( John needs hot drink ) ]
```

Question 5

(a) i. Explain the procedure for implementing the cover predicate in the context of version space learning. [3]

ii. Give the substeps from the candidate elimination algorithm for processing a negative training example. [7]

(b) Consider a concept description language with three attributes predefined as follows:

```
 attribute1        attribute2         attribute3
 ----------        ------------       ------------
|          |      |    |      |      |            |
a          b      c    d      e      f            g
```

Demonstrate version space learning using the following positive and negative training examples:

```
1.  ( a c f ) +)
2.  ( b c f ) +)
3.  ( a e g ) -)
4.  ( a c g ) -)
5.  ( b d f ) -)
```

Show how the candidate elimination algorithm changes the boundary sets after processing each example. [15]