

**NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY**  
FACULTY OF APPLIED SCIENCE  
**COMPUTER SCIENCE DEPARTMENT**  
DECEMBER EXAMINATIONS 2005

**SUBJECT:** ARTIFICIAL INTELLIGENCE  
**CODE:** SCS4101

**INSTRUCTION TO CANDIDATES**

Answer all questions in section A.  
Answer any 3 questions in Section B.

**Time: 3 hours**

**Section A**

**QUESTION ONE**

- a) List 5 areas Prolog can be used in as a system development language [5]
- b) Euclid's algorithm to calculate the greatest common divisor of two numbers tells us that:

$$gcd(x, y) = \begin{cases} x, & \text{when } x = y \\ gcd(x, y - x), & \text{when } x < y \\ gcd(x - y, y), & \text{when } x > y \end{cases}$$

Express this in Prolog as a recursive predicate where  $gcd(X, Y, Z)$  means "the greatest common divisor of X and Y is Z". [5]

- c) A palindrome is a word that reads the same is reverse e.g madam. Write a predicate  $pal(L1)$  which is true is if the list L1 is a palindrome [5]
- d) Suppose that `mystery` is queried with its first argument a bound list and its second argument an unbound list. Describe what `mystery` does in a sentence of 20 or less English words.

`mystery([],[]).`

`mystery([X],[X]).`

`mystery([X.Y|L],[X.censor|M]):-mystery(L,M).` [5]

## QUESTION TWO

- a) What is meant by the statement: "Prolog is a declarative and relational programming language". [2]
- b) Given the following program:
- ```
likes(ann,X) :- toy(X), plays(ann,X).
likes(john,Y) :- likes(ann,Y).
toy(doll).
toy(train).
plays(ann,train).
```
- Write queries to find out the following:
- i) Does Ann like dolls? [1]
- ii) Who likes trains? [1]
- iii) What does John like? [1]
- iv) Who *plays* with trains? [1]
- c) What is the main advantage of each of the following representations over the others: frames, semantic networks and logic [10]
- d) What does expert knowledge include? [4]

## Section B

## QUESTION THREE

- a) In the algorithm for the uniform cost search, the cost of the arc from node  $i$  to node  $j$  is denoted by  $c(i,j)$ . The cost of a path from the start node to any node  $i$  is denoted  $g(i)$ .

The algorithm is as follows:

- 1 Put the start node,  $s$ , on a list, called OPEN, of unexpanded nodes. If the start node is a goal node, a solution has been found. Otherwise, set  $g(s) = 0$ .
- 2 If OPEN is empty, no solution exists.

- 3 Select from OPEN a node  $i$  such that  $g(i)$  is minimum. If several nodes qualify, choose among them arbitrarily. Move node  $i$  from OPEN to a list, CLOSED, of expanded nodes.
- 4 If node  $i$  is a goal node, a solution has been found.
- 5 Expand node  $i$ . If it has no successors, go to (2).
- 6 For each successor node  $j$  of node  $i$ , compute  $g(j) = g(i) + c(i,j)$  and place all the successor nodes  $j$  in OPEN.
- 7 Go to (2).

Demonstrate this algorithm on the traveling salesman problem. Trace through all the steps until a solution is found. [7]

|   |   |   |   |    |
|---|---|---|---|----|
|   | A | B | C | D  |
| A |   | 4 | 6 | 10 |
| B |   |   | 7 | 10 |
| C |   |   |   | 5  |

Mileage chart

If all the arcs have equal cost this algorithm reduces to which search? Demonstrate this using the traveling salesman problem. [3]

- b) Translate the following sentences into a Prolog knowledge base:

John eats all kinds of food. Apples are food. Oysters are food.  
 Anything anyone eats is food. Tom eats snakes. Sue eats  
 everything that Tom eats. [4]

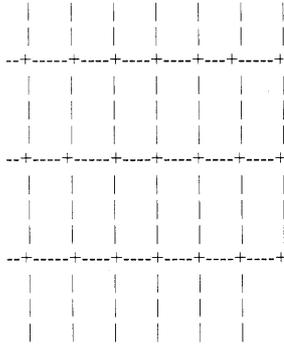
- c) Based on the above knowledge base formulate queries to find out:

- i) What John eats [1]
- ii) What Sue eats. [1]
- iii) If there is anything which both John and Sue eat. [1]
- iv) Who eats snakes [1]

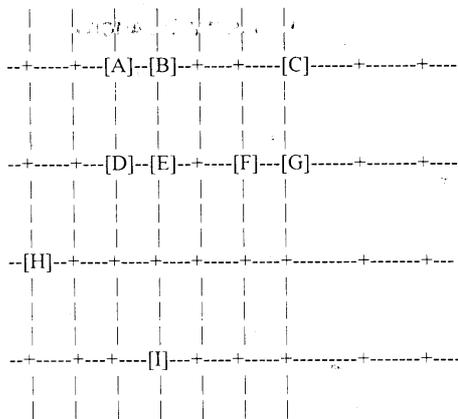
- d) Describe the 4 different versions of equality in Prolog [2]

#### QUESTION FOUR

Imagine a grid consisting of (evenly spaced) horizontal and vertical lines: assume that it is possible to place an object at the intersection of any two lines. Suppose also that the lines are potentially infinite in length.



- (a) Suggest an appropriate format for a Prolog knowledge base that will represent the position of objects without using absolute co-ordinates (remember - it's infinitely large in theory), but rather the position of the objects *relative* to each other - after all, Prolog is a *relational* language. [3]
- (b) Now, using this format write some rules which will check if
- i) an object is immediately to the right of another [1]
  - ii) an object is immediately to the left of another [1]
  - iii) an object is immediately above another [1]
  - iv) an object is immediately below another [1]
  - v) an object is exactly between two others, either horizontally or vertically [2]
  - vi) an object is directly beside another in a diagonal direction [2]
- (c) Using these rules write facts to describe the following configuration of objects on the grid [9]



**QUESTION FIVE**

- a) Without using Prolog's built-in multiplication or exponential operations, define predicates for the following:
- i)  $\text{mult}(x,y) = 0$ , when  $y=0$ .  
 $\text{mult}(x,y) = x+\text{mult}(x,y-1)$ , when  $y>0$  [3]
  - ii)  $\text{exp}(x,y) = 1$ , when  $y=0$ .  
 $\text{exp}(x,y) = \text{mult}(x,\text{power}(x,y-1))$ , when  $y>0$  [3]
- b) Explain fully the following concepts in AI
- i) Knowledge acquisition [2]
  - ii) Knowledge elicitation [2]
- (c) Come up with a state space representation for the following problem. There is a table and 3 toy blocks. The initial state of the problem is that blocks 2 and 3 are on the table, and block 1 is on top of block 2. we wish to reach a goal state in which the 3 blocks are stacked with block 1 on top, block 2 in the middle, and block 3 at the bottom. The only operator is MOVE X to Y, which moves object X onto another object Y. as preconditions to applying the operator, it is required that (a) X, the object to be moved, be a block with nothing on top of it and (b) if Y is a block, there must be nothing on Y. finally, the operator is not to be used to generate the same state more than once. [5]

- (d) Consider the following search algorithm and demonstrate it on the above problem by tracing through the steps to generate a search tree [3]
- 1 put the start node on a list, called OPEN, of unexpanded nodes. If the start node is a goal node, a solution has been found.
  - 2 If OPEN is empty, no solution exists.
  - 3 Remove the first node, n, from OPEN and place it in a list, called CLOSED, of unexpanded nodes.
  - 4 Expand node n, if it has no successors, go to 2
  - 5 Place all successors of node n at the end of the OPEN list.
  - 6 If any of the successors of node n is a goal node, a solution has been found. Otherwise go to 2
- (e) What is the name of this algorithm? state its advantages and disadvantages as a search algorithm [2]

### QUESTION SIX

- a) Explain from the provided definitions what the following predicates do:
- i) copper([X|L],X). [2]
  - ii) nickel([X],X).  
nickel([X|L],X2):-nickel(L,X2). [4]
  - iii) coal([],0).  
coal([X|L],N):-coal(L,N2),N is N2 + 1. [4]
  - iv) neon([X],X).  
neon([X|L],X):-neon(L,M),X > M.  
neon([X|L],M):-neon(L,M),X <= M. [5]
- b) From what you know about AI what are its shortcomings and what proposals would you submit for research to overcome them. [5]

**END OF QUESTION PAPER**