Last Name **……………………………………….First Name** **………………...……………**
Student # **……………………………………………………………….**

The University of British Columbia
Department of Computer Science
Midterm examination - Fall 2004
Computer Science 322 (Introduction to Artificial Intelligence)

Please try to write your answers in the space provided below each question (you can use the back of the page as well). If you need to use additional sheets of paper, remember to write your name on it.

Question	**Maximum**	**Received**
One	10	
Two	10	
Three	10	
Four	15	
Five	5	
TOTAL	45	

Question 1 [10 marks]
Consider the language with constant symbols *a, b, c*, predicate symbols *p, q* and *s* and the knowledge base KB:

s*(X) <- q(X,Y) & s(Y)*
s(Y) <- p(Y).
q(a,b).
p(a).

Suppose there are three individuals *pet, man*, *kid*

a. Give a model of KB where D={pet, man, kid}. You must clearly and completely specify ϕ and π.[7 marks]

Phi(a) = pet, phi(b) = man, phi(c) = kid

Pi(p)(pet) = true, pi(p)(man) = true, pi(p)(kid) = true
Pi(q)(pet, pet) = true, Pi(q)(pet, man) = true, Pi(q)(pet, kid) = true,
Pi(q)(man, pet) = true, Pi(q)(man, man) = true, Pi(q)(man, kid) = true,
Pi(q)(kid, pet) = true, Pi(q)(kid, man) = true, Pi(q)(kid, kid) = true,
Pi(s)(pet) = true, pi(s)(man) = true, pi(s)(kid) = true

b. Give an interpretation with the same domain that isn't a model of KB. It is sufficient that you indicate how to modify ϕ and/or π in (a) to answer this question.[3 marks]

Same as above except for: Pi(p)(pet) = false

Question 2 [10 marks]

Consider the knowledge base KB with numbered clauses:

1) rich-neighbors(X,Y) <- location(house-of(X),Z) & location(house-of(Y),Z) & upscale(Z).
2) location(P,westvan) <- street(main,P).
3) location(house-of(john),westvan).
4) street(main,house-of(peter)).
5) upscale(westvan).

Suppose we have the query ?rich-neighbors(Q,R). The following is a derivation for the query in which we always select the leftmost atom in the body. The derivation has some parts missing (labeled **(a), (b), (c) (d)** and **(e)**, that you need to fill in the space provided. You do not need to worry about renaming clauses.

yes(Q,R) <- rich-neighbors(Q,R)

choose clause 1

substitution {X/Q,Y/R}

yes(Q,R) <- **(a)______location(house-of(Q),Z) & location(house-of(R),Z) & upscale(Z)________**

choose clause 2

substitution **(b) ________{P/house-of(Q), Z/westvan}__________________________________.**

yes(Q,R) <- street(main, house-of(Q)) & location(house-of(R),westvan) & upscale(westvan).

choose clause: **(c) ______________4__________________**

substitution {Q/peter}

(d)___yes(peter,R)_____ <- location(house-of(R),westvan) & upscale(westvan).

choose clause 3

substitution **(e)___________{R/john}____**

yes(peter,john) <- upscale(westvan).

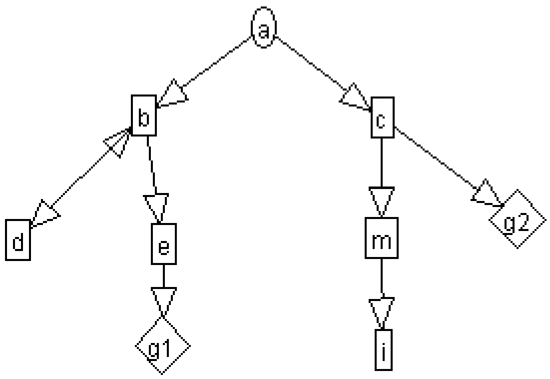
choose clause 5

substitution {}

yes(peter,john)

Question 3 [10 marks]

Consider the following graph, in which the neighbors are given from left to right in the graph, that is:



neighbours(a, [b,c])
neighbours(b, [d, e])
neighbours(d, [b])
neighbours(e, [g1])
neighbours(c, [m,g2])
neighbours(m, [i])

a. For each of the following search algorithms, list the nodes that it expands to get to a goal node, and indicate the solution path found, if any. Assume that multiple-path pruning is used, and that the search stops after the first solution is found.

– [2 marks] Depth first search (no cycle checking, no multiple path pruning)

Nodes expanded: **a, b, d, , b, d…………..**

Solution path **none**

– [2 marks] Depth first search (with cycle checking)

Nodes expanded: **a, b, d, e , g1**

Solution path **a, b, e, g1**

– [2 marks] Breadth first search (no cycle checking, no multiple path pruning)

Nodes expanded: **a, b, c, d, e, m, g2**

Solution path: **a, c, g2**

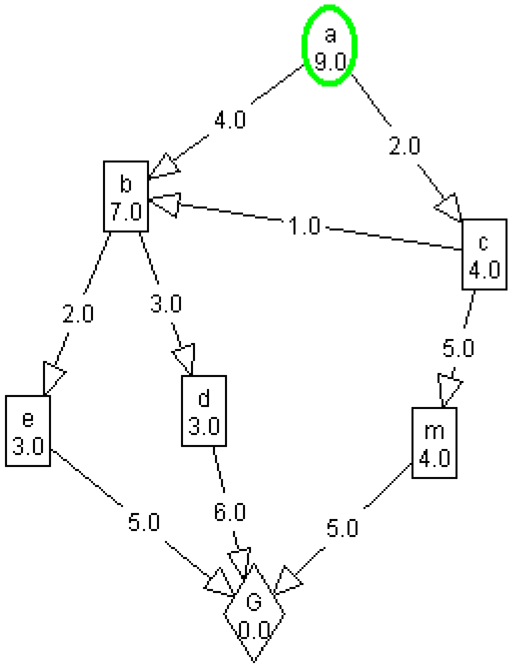
b. In each of the spaces below, list the nodes that are expanded by Iterative Deepening Search (IDS) at that depth bound, in order to find a solution (no cycle checking, no multiple path pruning). List the nodes in the order they are expanded. Show clearly what is the solution path found, and how many nodes have been expanded to find it. Assume that the search stops after the first solution is found. Add more depth bounds if you think is necessary [4 marks].

Depth bound 0	**a**
Depth bound 1	**a, b, c**
Depth bound 2	**a, b, d, e c, m g2**

– Solution path found? a, c, g2

– Number of nodes expanded? 11

Question 4 [10 marks]



Consider the search space in the figure to the left. The numbers on the arcs represent the cost of the path between the two nodes, the number inside each node represents the value of the heuristic function *h()* for that node. You have to show how A* finds a path to the goal node *G*.

For each iteration of A*, show the path removed from the frontier and the path(s) added to the frontier to get from node *a* to node G. Give the g-value, h-value and f-value for each new node added to the frontier. Assume \that the search stops after the first path is found. Please continue the trace provided and show clearly the path found to the goal node *G*.

TRACE (suggestion: check off a path when you remove it from the frontier, it will be easier too see which paths are still on the frontier)

Number Path g h f
1 a 0 10 10
Remove path number 1
2. a -> b .4..........7.............11...
3. a -> c 2...........4.............6..
Remove path number ..3..
4.....................a -> c->b 3...........7.............10..
5.....................a -> c->m 7...........4.............11..
Remove path number 4
6................a -> c->b->e 5...........3.............8
7............... .a -> c->b->d 6...........3.............9...
Remove path number 6
8................a -> c->b->e->g 10...........0.............10
Remove path number 7
9............... .a -> c->b->d->g 12...........0.............12.............
Remove path number 8
Goal found a->c->b->e->g

....

....

....

....

....

....

....

b) (b) [5 marks] Consider the search graph in part (a). Can we do multiple path pruning with A* and still get an optimal solution? Explain your answer (you'll get no marks without an explanation).

No, using multiple path pruning does not give the optimal path because h(n0) does not satisfy the monotone restriction. Because $h(a) – h(c) > d(a,c)$, A* finds the path a->b before it finds the shorter path a->c->b, which would then be pruned by multiple path pruning.

Question 5 [5 marks]

Suppose that you have a search space for which you found two different heuristic evaluation functions h_1 and h_2, such that

$h_1(n) \leq h_2(n)$ for every node n in the search space.

Also, both h_1 and h_2 underestimate the actual cost of the path from each node to the goal

Can we say that A* using h_2 will always expand equal or fewer nodes than A* using h_1 to find a solution? Explain your answer (you'll get no marks without an explanation)

Yes. If f* is the actual cost of the optimal solution, we know that f* will expand all the nodes n with $f(n)< f^*$ before expanding the goal node.
Thus, every node with $h(n) < f^*-g(n)$ will be expanded. Because h2 is as least as big as h1, all the nodes expanded by h2 will also be expanded by h1, and h1 may also cause other nodes to be expanded.