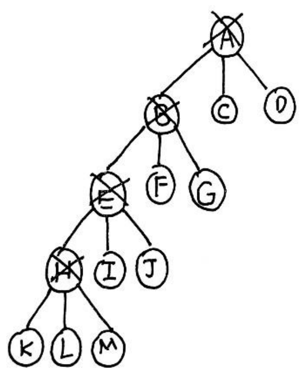
Solver: Jesslyn Chew

1) a) Search algorithm optimality means that the search algorithm used will always find the best or least-cost solution. For example, breadth-first search will find the optimal solution when all step costs are equal. Whereas the search algorithm Greedy does not guarantee optimality as it may not always provide the solution will the lowest cost.

b) Total states = 4^8 = **65536**

c) Based on Dominance. The better heuristic to use would be the heuristic that results in a larger value for all n, since it will expand more nodes. <This is proven in lecture slide on informed search methods, page on dominance.> However, the heuristics still needs to fulfill the condition that the heuristics value must be less or equal to the actual cost. This results in (iii) not being a viable heuristic. Hence, option (ii) max (h1; h2) would yield a better heuristic.

d) Frontier nodes are candidate nodes for expansion.



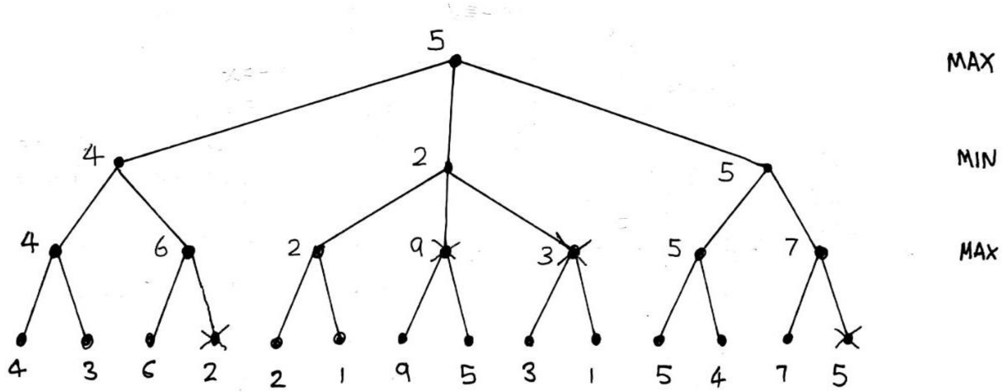
Let b = 3 and d = 5. The crossed-out nodes are in explored nodes. The diagram above displays the maximum frontier nodes. The frontier nodes of a depth-first search are the unexplored children nodes of the branch being explored (E.g. node K, L, M) and the internal nodes that have yet to be selected to be branched (E.g. node C, D, F, I, etc). From the diagram, we can derived that the maximum frontier size is 2*3 + 3. From this we can infer that the maximum size of a DFS frontier is (b-1)(d-2) + b.

e) I would prefer BFS for this problem as DFS will not provide a complete solution for finite-depth space with loops, unless there is repeated-state checking. Whereas BFS will provide a complete solution.

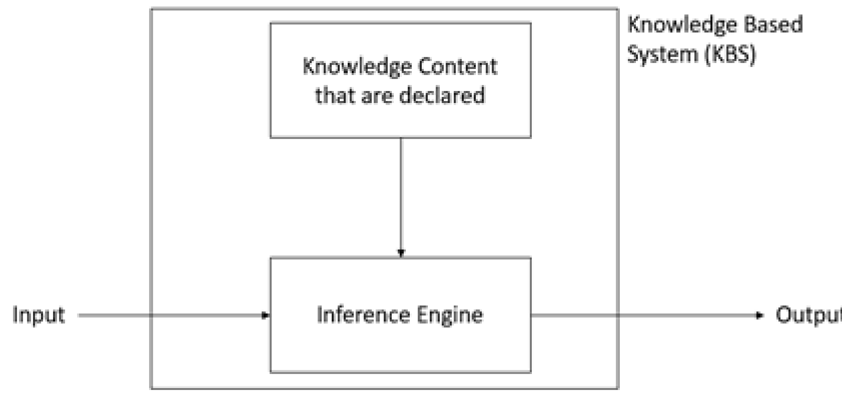
2) a) Arc consistent implies that for each of its admissible values is mapped to some admissible value of the second variable. In this case, it means that for every value in the domain of X, there exists r(X; Y) that is within the domain of the second variable.

b) <No solution available>

c)



3) a) i)



The knowledge content and inference engine are stored separately, such that any query of knowledge first goes to the inference engine. The inference engine would then retrieve knowledge from the knowledge content and determines the output. This is similar to the network of neurons in the human body, where the neurons are the inference engine. It gets input from the dendrites, does processing as well and sends output to the axon.

ii) <No solution available>

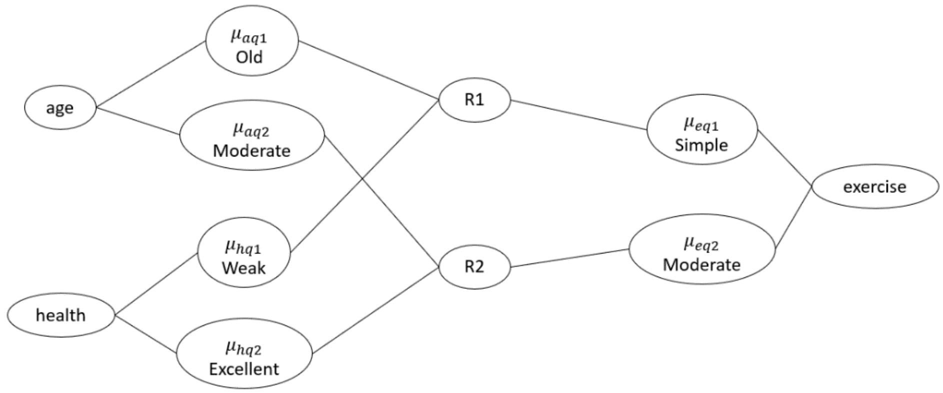
iii) <Question 3(a)iii] and 3(b) are similar to SEM 2 EXAM 2017-2018 Q4(c)>

b) <Question 3(a)iii] and 3(b) are similar to SEM 2 EXAM 2017-2018 Q4(c)>

c) i) Let there be a rule $A \wedge B \Rightarrow C$

For each if-then rule in the fuzzy operation, each antecedent (premise) is considered. For each antecedent, we will obtain its intersection with the fact. The conclusion is then updated using the fuzzy implication relation (AND) on those intersections. The fuzzy output is the aggregate of the conclusion of each fuzzy rule. The final output is produced when the fuzzy output is defuzzified using either maximum membership or centroid.

ii)



iii) Fuzzy logic enables to select a degree of truth rather than restricting the truth value to either true or false. In order for fuzzy inference to the rule conflict issues in traditional first order logic, the degree of truth will be perceived as either 1 or 0 instead of a range between 0 to 1.

4) a) Predicates:

$predecessor(x, y)$ - "y is a predecessor of x"
$successor(x, y)$ - "y is an immediate successor of x"
$pass(x, y)$ - "x passes holdings to y"
$eldestsuccessor(x, y)$ - "y is eldest successor of x"

Knowledge base:
"… all agricultural or .. if first-born not female"
$\forall x, \forall y, successor(x, y) \land eldestsuccessor(x, y) \rightarrow pass(x, y)$ ①
"This line of ownership .. along order of birth"
$\forall w, \forall x, \forall y, \forall z, \big(\neg successor(x, y) \land predecessor(x, z) \land successor(z, w) \land \neg(w = x) \rightarrow pass(x, w)\big)$ ②

b) Translating ①: $\neg successor(x, y) \lor \neg eldestsuccessor(x, y) \lor pass(x, y)$
②: $\neg(\neg successor(x, y) \land predecessor(x, z) \land successor(z, w) \land \neg(w = x) \lor pass(x, w)$
$successor(x, y) \lor \neg predecessor(x, z) \lor \neg successor(z, w) \lor w = x \lor pass(x, w)$

c) <No solution available>

--End of Answers--