THE UNIVERSITY OF HONG KONG

FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS

CSIS0270 Artificial Intelligence

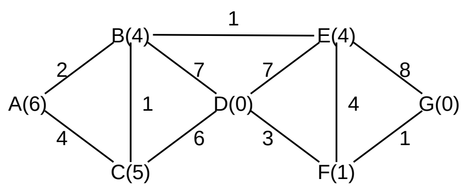
Date: June 3, 2002 Time: 9:30 am–12:30 pm

Candidates may use any calculator which fulfils the following criteria: (a) it should be self-contained, silent, battery-operated and pocket-sized; (b) it should have numeral-display facilities only and should be used only for the purpose of calculation; (c) it should not have any printing device, alphanumeric keyboard, or graphic display; and (d) it should not contain any recorded data or program. It is the candidate's responsibility to ensure that the calculator operates satisfactorily and the candidate must record the name and type of the calculator on the front page of the examination scripts. Lists of permitted/prohibited calculators will not be made available to candidates for reference, and the onus will be on the candidate to ensure that the calculator used will not be in violation of the criteria listed above.

Answer all questions in the answer book provided. Each of the 7 questions is worth 10%, and the marks of the 3 questions you score the most will be doubled.

1. *(Searching)*

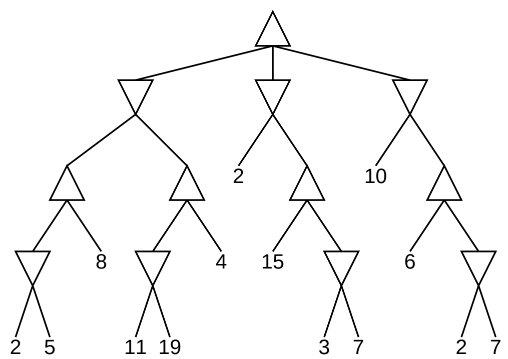
   a. *(6%)* In the graph below, we want to search from A to find a goal state, which is either D or G. The path cost is the sum of edge costs, which are shown in the diagram. A heuristic function is available (which values are indicated below within parentheses). Explain whether the heuristic is admissible. Perform A* search in the graph, and draw the resulting search tree and the f-value, g-value and h-value of each node. You can choose to use any reasonable strategy to deal with duplicates.



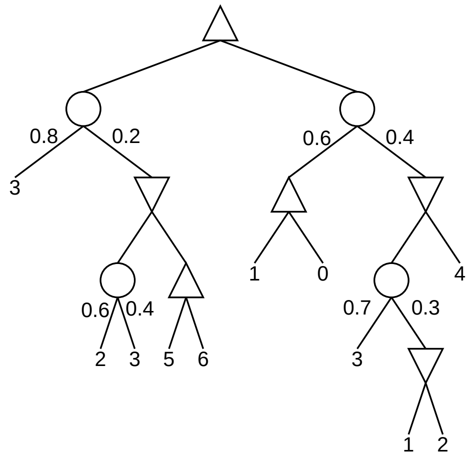
   b. *(4%)* Under what kind of search space will breath-first search be much better than iterative-deepening depth-first search?

2. *(Game playing)*

   a. *(5%)* Perform Alpha-Beta on the following game tree, evaluating branches from left to right. Show all your workings, and indicate all the places at which pruning is done.



b. *(5%)* Perform expectiminimax on the following game tree to find the utility of all the nodes. Which parts of the evaluation can be avoided if alpha-beta is applied on the tree, given that the utility of all nodes is in the range [0, 10]? Explain your answer.



3. *(Unification and Chaining)*

a. *(4%)* Convert the following into Horn form. Is there any knowledge that cannot be used for backward chaining? Explain why. Use backward chaining to derive that Miss Chan is happy. Show the complete and-or tree searched.

> Anyone passing his AI exam and winning a lottery is happy. Anyone who study or who is lucky passes his AI exam. Anyone who is lucky wins a lottery. Miss Chan does not study, but she is lucky.

b. *(6%)* Suppose you want to add a substitution $x/Term$ to an already existing unifier θ. There are four different cases, where x already has a binding in θ, *Term* already has a binding in θ, x occurs in *Term*, and other cases. Explain what the unification algorithm should do in each case, and give example illustrating your answer. Explain why we must bind variables when we perform the occurrence check (if we don't perform it elsewhere).

4. *(CNF and Resolution)*

a. *(4%)* Convert the following sentence into CNF by normalization. Show all your steps. Comment about how to simplify the knowledge further by subsumption.

$$\forall x\ (P(x) \vee Q(x)) \Rightarrow \neg\exists y\ (P(x) \wedge P(y) \wedge \forall x\ R(x, y))$$

b. *(6%)* Suppose we have the following knowledge:

1. $P(x, y) \vee R(y, x)$
2. $\neg Q(B, x) \vee \neg P(B, x)$
3. $\neg P(x, y) \vee Q(x, A) \vee R(y, x)$

Give a linear resolution proof for the sentence $R(A, B)$.

5. *(Prolog)* Suppose you have a graph defined by a Prolog relation `next`. E.g., if there is an edge from node `a` to node `b`, the following relation holds:

```
next(a, b)
```

- *(6%)* Write a predicate `path(X, Y, Z)`, which is true if `Y` is a list with the first element `X`, the last element `Z`, and there is an edge between each consecutive pairs of elements in the list `Y`. E.g., if `next(a,b)` and `next(b,e)`, then `path(a, [a,b,e], e)` is true. The predicate `path(a, [a], a)` should be true for all `a`.

- *(4%)* Using the result of part (a), write a predicate `shortestPath(X, Y, Z)`, such that when `X` and `Z` are bounded and `Y` is unbounded, `Y` is unified to a shortest path between `X` and `Z`.

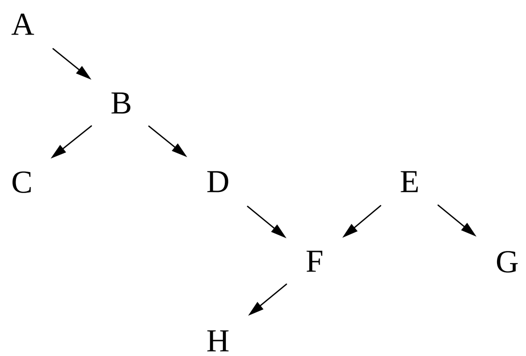
6. *(Planning)* Consider the following planning problem, in STRIP notation:

- Objects: C, D, E
- Predicates: $P1(x), P2(x, y)$ (x and y can be any of the 3 objects.)
- Initial state: $P1(C)$
- Goal: $P2(D, E) \wedge P2(E, D)$
- Action 1: $A1(y)$, precondition $P1(C)$, effects $P1(y)$
- Action 2: $A2(x, y)$, precondition $P2(x, y)$, effects $P1(x) \wedge \neg P2(x, y)$
- Action 3: $A3(x, y)$, precondition $P1(x) \wedge P1(y)$, effects $P2(x, y) \wedge \neg P1(x)$

a. *(6%)* Give a smallest (i.e., fewest state and ordering constraints) partial-order plan that solves the problem. Show all the causal links (you may use double arrows to represent causal links).

b. *(4%)* For each ordering constraint that is not supported by a causal link, explain what conflict the ordering constraint resolves, and explain whether you are performing promotion or demotion.

7. *(Uncertainty)* Consider the following belief network, with each symbol being a boolean event:



Suppose we observed the events B, G and $\neg H$.

a. *(5%)* What are the three situations when d-separation can be established? For each of these situations, lists all pairs of un-observed events which can thus be deduced to be independent.

b. *(5%)* Assume the following tables of probability. Illustrate the linear-time BELIEF-NET-ASK algorithm by computing the posterior probability $P(F | B \wedge G \wedge \neg H)$.

Show all your workings. (Partial credit is awarded if you compute the probability using any other method.)

P(A) = 0.6

A	P(B)
true	0.5
false	0.8

B	P(C)
true	0
false	0.7

B	P(D)
true	1
false	0.5

P(E) = 0.6

D	E	P(F)
false	false	0.2
false	true	0.9
true	false	0.7
true	true	0.1

E	P(G)
true	0.3
false	0.5

F	P(H)
true	0.4
false	0.7

END OF PAPER