Midterm Examination

CS 540: Introduction to Artificial Intelligence

July 19, 2006

LAST NAME: ___________SOLUTION_____________________________

FIRST NAME: ____

Problem	**Score**	**Max Score**
1	________	10
2	________	22
3	________	12
4	________	16
5	________	20
6	________	14
Free		6
Total	________	100

1. [10] **Search**

Answer each of the following questions. If you are not sure of your answer, you may include a brief explanation.

(a) [2] True or False: Greedy Best-First search with an admissible heuristic is guaranteed to find an optimal (i.e., least cost) solution.

```
False.  f=h is not admissible.
```

(b) [2] True or False: Hill-Climbing search is a complete algorithm for solving constraint satisfaction problems (CSPs).

```
False.  Can get stuck at a local minimum and fail to find a
solution.
```

(c) [2] True or False: Simulated Annealing with a fixed, positive temperature, T>0, gives the same result as Hill-Climbing.

```
False.  T>0 means there is a positive probability of moving to a
worse state.
```

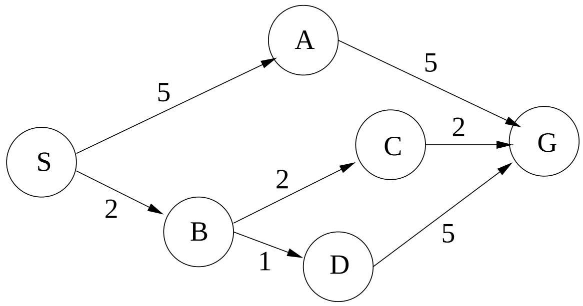
(d) [2] True or False: An optimal solution path for a search problem with positive arc costs will never have repeated states.

```
True.  For any solution path with repeated states, removing the
cycle  will also be a solution with lower cost.
```

(e) [2] True or False: Beam search with a beam width of 3 and an admissible heuristic function is guaranteed to find an optimal solution.

```
False.  It may throw away a node on the optimal path.
```

2. [22] **Heuristic Search**
Consider the following search space in which the goal is to find a path from the start state S to the goal state G.



Node	h_0	h_1	h_2
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

(a) [6] Which of the above heuristic functions, h_0, h_1, and h_2, are admissible?

h_0 and h_1

(b) [12] Give the solution path found by the A or A* algorithm using *each* of the three heuristic functions. Break ties alphabetically.

h_0: S -> B -> C -> G

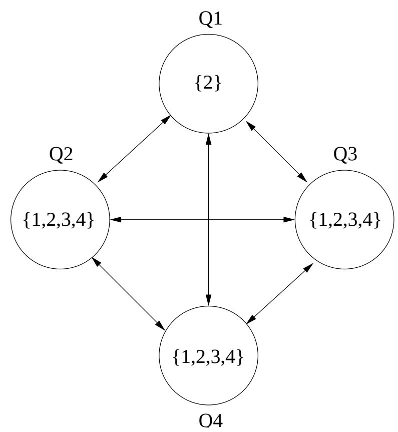
h_1: S -> B -> C -> G

h_2: S -> B -> D -> G

(c) [4] What solution path is found by Greedy Best-First search using h_1? Break ties alphabetically.

S -> A -> G

3. [12] **Constraint Satisfaction**
Consider solving the 4-queens problem as a CSP. That is, place 4 queens on a 4×4 board such that no queen is in the same row, column or diagonal as any other other queen. One way to formulate this problem is have a variable for each queen and binary constraints between each pair of queens indicating that they cannot be in the same row, column or diagonal. Assuming that the *i*th queen is put somewhere in the *i*th column, then the possible values for each variable are the row numbers in which it could be placed. Say we initially also assign queen Q1 the unique value 2, meaning Q1 is placed in column 1 and row 2. This results in an initial constraint graph given by:



(a) [6] Apply *forward-checking* and give the resulting values for the variables Q2, Q3, and Q4.

`Q2: {4}`

`Q3: {1,3}`

`Q4: {1,3,4}`

(b) [6] Starting from the same initial configuration shown in the figure, apply *arc consistency* checking and show the resulting values for the variables Q2, Q3, and Q4.

`Q2: {4}`

`Q3: {1}`

`Q4: {3}`

4. [16] **Decision Trees**

Consider the following set of 8 training examples, each indicating whether or not a student "aced" their exam, given by the Boolean classification variable *A*, given two Boolean attributes: *S*, specifying if the student studied or not, and *C*, specifying if the student used a "cheat sheet" or not. For your information, $\log 0.1 = -3.3$, $\log 0.2 = -2.3$, $\log 0.25 = -2.0$, $\log 0.3 = -1.7$, $\log 0.4 = -1.3$, $\log 0.45 = -1.15$, $\log 0.5 = -1.0$, $\log 0.55 = -0.85$, $\log 0.6 = -0.7$, $\log 0.7 = -0.5$, $\log 0.75 = -0.4$, and $\log 0.8 = -0.3$, $\log 0.9 = -0.15$, and $\log 1 = 0$, where all logs are to base 2.

S	**C**	**A**
T	F	T
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F

(a) [2] What is the **Information** content (aka entropy) of the classification variable, A, for this set of examples?

I(2/8, 6/8) = (-0.25)log 0.25 + (-0.75)log 0.75 = (-0.25)(-2.0) + (-0.75)(-0.4) = 0.8

(b) [4] Compute the **Remainder** and **Information Gain** associated with choosing the attribute S for the root of the decision tree. Show your work. Round values in order to use the log table given above.

$$Remainder(S) = 5/8\ I(5/5, 0/5) + 3/8\ I(1/3, 2/3)$$

$$= (5/8)(0) + (3/8)((-.33)log(.33) + (-.67)log(.67))$$

$$= .375((-.33)(-1.7) + (-.67)(-.5)) = 0.336$$

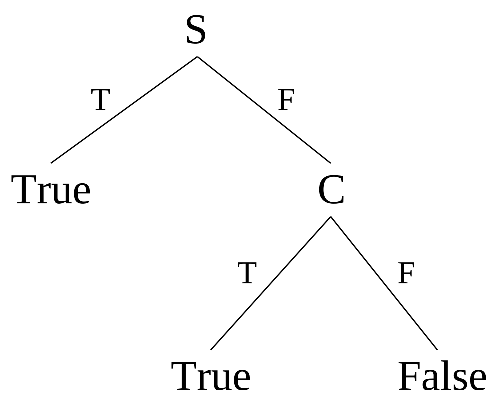
$$Gain(S) = I(2/8, 6/8) - Remainder(S) = 0.8 - 0.336 = 0.464$$

(c) [4] Compute the **Remainder** and **Information Gain** associated with choosing the attribute C for the root of the decision tree. Show your work.

$$Remainder(C) = 4/8\ I(3/4, 1/4) + 4/8\ I(3/4, 1/4) = (.5)(.8) + (.5)(.8) = 0.8$$

$$Gain(C) = I(2/8, 6/8) - Remainder(C) = 0.8 - 0.8 = 0.0$$

(d) [3] Draw the complete decision tree learned from this set of examples.




```
Note: The leaf below the "T" branch of attribute C could
alternatively have classification False depending on how the
Majority-Value function breaks ties.
```

(e) [3] After constructing a large, complex decision tree from a training set that contains many attributes you find that the training set accuracy is very high but the test set accuracy is very low. Explain why this situation might have occurred.

```
Overfitting of the training data.
```

5. [20] **Logic**

(a) [3] Is the Propositional Logic (PL) sentence $(A \Leftrightarrow B) \wedge (\neg A \vee B)$ valid, unsatisfiable, or satisfiable? Briefly explain your answer.

A	B	$A \Leftrightarrow B$	$\neg A \vee B$	$(A \Leftrightarrow B) \wedge (\neg A \vee B)$
T	T	T	T	T
T	F	F	F	F
F	T	F	T	F
F	F	T	T	T

`Since the last column contains both T and F, the sentence is satisfiable.`

(b) [2] True or False: $(A \wedge B) \; |= \; (A \Leftrightarrow B)$

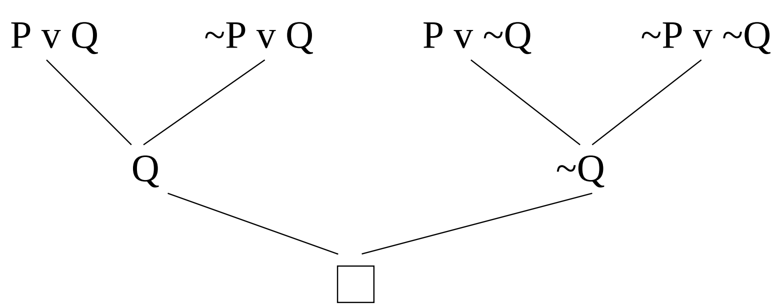
`True`

(c) [3] Prove whether or not the rule of inference $\dfrac{P \Rightarrow Q,\ \neg Q}{\neg P}$ is sound.

`This is Modus Tolens, which is proved sound by noting in the truth table that whenever the next to last column is T, the last column is also T:`

P	Q	$P \Rightarrow Q$	$\neg Q$	$(P \Rightarrow Q) \wedge \neg Q$	$\neg P$
T	T	T	F	F	F
T	F	F	T	F	F
F	T	T	F	F	T
F	F	T	T	T	T

(d) [4] Prove that the following set of 4 clauses is unsatisfiable by constructing a refutation tree.
$P \vee Q$, $P \vee \neg Q$, $\neg P \vee Q$, $\neg P \vee \neg Q$



(e) [2] Does the following pair of literals unify and, if so, give their most general unifier. P is a predicate, F is a function, A is a constant, and x and y are variable symbols.

$P(F(x, x), A)$ and $P(F(y, F(y, A)), A)$

```
Unification would produce {x/y, y/F(y,A)} which contains an
"occurs check," which means the two literals do not unify.
```

(f) [6] The English sentence "Everyone likes someone who has red hair" is ambiguous in that it might mean (i) there is someone (i.e., one particular person) with red hair who everyone likes, or (ii) each person likes a possibly different red-haired person. Using the predicates *Red-haired(x)* and *Likes(x,y)*, meaning x likes y, translate each of these two meanings into first-order logic.

(i) [3] There is someone with red hair who everyone likes.

$$\exists x\ \forall y\ (Red-hair(x) \wedge Likes(y, x))$$

(ii) [3] Each person likes somebody with red hair.

$$\forall x\ \exists y\ (Red-hair(y) \wedge Likes(x, y))$$

6. [14] **Deductive Inference in Propositional Logic**
Consider the following sentence in Propositional Logic:

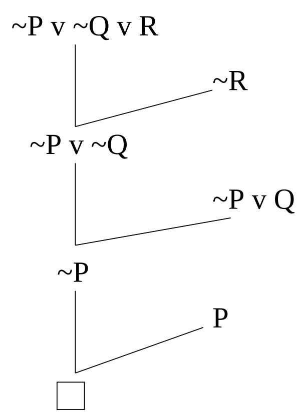
$$(P \Rightarrow (Q \Rightarrow R)) \Rightarrow ((P \Rightarrow Q) \Rightarrow (P \Rightarrow R))$$

(a) [10] Prove that the given sentence is valid using the Resolution Refutation algorithm. (Note: There are no premises in this problem, just a theorem to be proved.)

(i) [4] Give the set of clauses that will be available for your refutation proof.

$\neg P \vee \neg Q \vee R$, $\neg P \vee Q$, P, and $\neg R$

(ii) [6] Show the refutation proof tree.



(b) [4] Are the clauses you gave in (a)(i) Horn clauses? Can they be used to construct a backward-chaining goal-reduction proof? Explain briefly.

This was a poor question. The answer to the first question is yes if you use the definition in the book, which defines a Horn clause as a disjunction of literals containing at most one positive literal. This definition means that $\neg R$ is a Horn clause of the type called an integrity constraint. In class I said that Horn clauses must have a single positive literal, which is actually the definition of a definite Horn clause; using this definition, the answer is no because $\neg R$ is not a definite Horn clause. Furthermore, the second question isn't clear in terms of what the query is, given the set of clauses. So I threw out this part altogether.