Midterm Examination

CS540: Introduction to Artificial Intelligence

October 21, 2009

LAST NAME: ______________SOLUTION______________

FIRST NAME: ____

SECTION (1=Dyer, 2=Zhu): ____________________________

Problem	Score	Max Score
1	________	10
2	________	12
3	________	15
4	________	9
5	________	9
6	________	15
7	________	15
8	________	15
Total	________	100

Question 1 [10]. Entropy

Consider a six-sided die with equal probability on each side. We define the Boolean variable LARGE=True if the die roll outcome is 4, 5, or 6, and LARGE=False otherwise. We define EVEN=True if the outcome is even, and EVEN=False otherwise.

(a) [5] What is the entropy of LARGE? What is the entropy of EVEN?

```
H(LARGE)= -0.5*log(0.5) - 0.5*log(0.5) = 1 bit
H(EVEN)= 1 bit
```

(b) [5] What is the information gain in bits for LARGE in predicting EVEN? Be sure to show the formula and the steps.

```
I(LARGE; EVEN) = H(LARGE) – H(LARGE | EVEN)
```

$$= 1 - \sum_{e=T,F} P(EVEN = T)H(LARGE \mid EVEN = T)$$

$$= 1 - 0.5H(LARGE \mid EVEN = F) - 0.5H(LARGE \mid EVEN = T)$$

$$= 1 - 0.5(\sum_{l=T,F} -P(LARGE = l \mid EVEN = F)\log P(LARGE = l \mid EVEN = F))$$

$$-0.5H(LARGE \mid EVEN = T)$$

$$= 1 - 0.5(-1/3\log(1/3) - 2/3\log(2/3)) - 0.5(-2/3\log(2/3) - 1/3\log(1/3))$$

$$= 1 + 1/3\log(1/3) + 2/3\log(2/3)$$

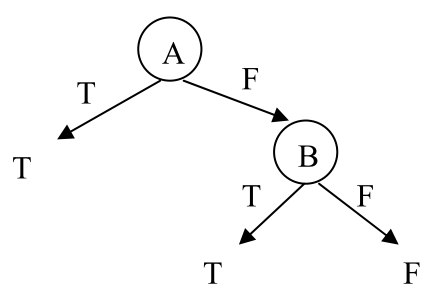
$$= 0.0817 bits$$

Question 2 [12]. Decision Tree and Logic

Consider the following set of 4 training examples, each containing two Boolean attributes, A and B, and a desired Boolean classification.

A	B	Class
T	F	T
T	F	T
F	T	T
F	F	F

(a) [4] Draw the decision tree trained from the above examples, with A as the root node.

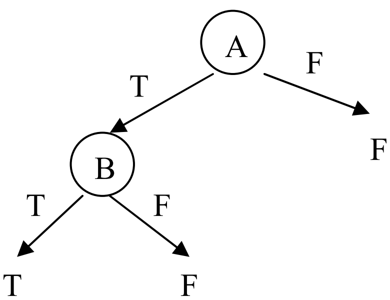


(b) [4] Write down the shortest equivalent propositional logic sentence as represented by the tree, in the sense that it should produce the same classification on these 4 training examples (hint: use two symbols A,B).

$A \vee B$

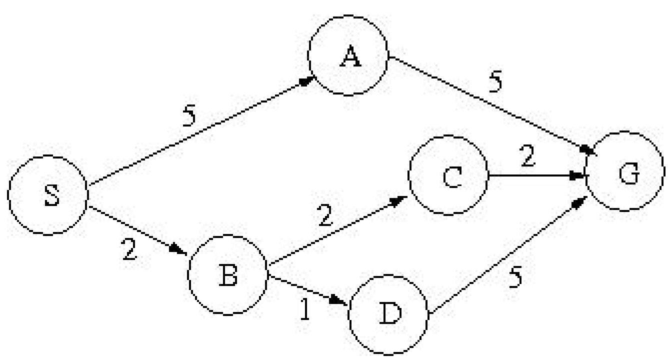
(c) [4] (This question is unrelated to the questions above.) Draw a decision tree equivalent to $\neg(A \Rightarrow \neg B)$.

Note that $\neg(A \Rightarrow \neg B)$ is equivalent to $A \wedge B$. The tree is therefore



Question 3 [15]. Search

Consider the following search space where we want to find a path from the start state S to the goal state G. The table shows three different heuristic functions *h1, h2*, and *h3*.



Node	*h1*	*h2*	*h3*
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) [5] What solution path is found by Greedy Best-first search using *h2*? Break ties alphabetically.

S, A, G

(b) [5] What solution path is found by Uniform-Cost search? Break ties alphabetically.

```
Sequence of nodes expanded:  S, B, D, C, A, G
Solution path:  S, B, C, G
```

(c) [5] Give the three solution paths found by algorithm A using each of the three heuristic functions, respectively. Break ties alphabetically.

h0: This is the same as uniform-cost search, so the answer is the same as (b). That is, the solution path is: S, B, C, G

h1: S, B, C, G expanded; solution: S, B, C, G

h2: S, B, D, G expanded; solution: S, B, D, G

Question 4 [9]. Heuristics

(a) Consider the 8-puzzle in which there is a 3 x 3 board with eight tiles numbered 1 through 8. The goal is to move the tiles from a start configuration to a goal configuration, where a move consists of a horizontal or vertical move of a tile into an adjacent position where there is no tile. Each move has cost 1.

(i) [3] Is the heuristic function defined by $h = \sum_{i=1}^{8} \alpha_i d_i$ **admissible**, where d_i is the number of vertical plus the number of horizontal moves of tile i from its current position to its goal position assuming there are no other tiles on the board, and $0 \le \alpha_i \le 1$ is a constant weight associated with tile i? Explain briefly why or why not.

Yes, it is admissible because each d_i is a lower bound on the number of moves to get each tile to its goal position and the weights decrease those values.

(ii) [3] Is the heuristic defined by $h(n) = 8 - \mathrm{cost}(n)$ **admissible**, where cost(n) is the cost from start to node n? Explain briefly why or why not.

No, it is not admissible because, for example, if a start node configuration is only one move away from the goal configuration, then h(start) = 8 – 0 = 8 but h*(start) = 1. Also, there are situations where h(n) could be negative for some nodes, which is not allowed.

(b) [3] Given two arbitrary admissible heuristics, *h1* and *h2*, which composite heuristic is better to use, max(*h1*, *h2*), (*h1* + *h2*)/2, or min(*h1*, *h2*)? Explain briefly why.

Max(h1, h2) is best because it is admissible but greater than or equal to the other two composite heuristics for all nodes.

Question 5 [9]. Hill Climbing

For each statement, decide whether it's True or False, and give a one-sentence justification.

(a) [2] There can be more than one global optimum.

```
True.  These global optima have the same value.
```

(b) [2] It is possible that every state is a local optimum. (A local optimum is defined to be a state that is NO BETTER than its neighbors)

```
True.  The whole state space is a plateau.
```

(c) [2] Hill climbing with random restarts is guaranteed to find the global optimum if it runs long enough on a finite state space.

```
True.  Eventually, random restart will hit the
global optimum, and hill climbing will stay there.
```

(d) [3] Let *f(s)* be the score (i.e., value or fitness) of state *s*. Genetic Algorithm is expected to work better than Simulated Annealing, if in the middle of the search we suddenly change *f* to –*f* (equivalently, changing the problem from finding the state with the maximum score to finding the state with the minimum score).

```
True.  Genetic algorithm maintains a population,
which has the ability to produce new states that can
adapt to the new scoring function (since it's
stochastic search).  On the other hand, simulated
annealing could get stuck in the vicinity of its
current state if the temperature is low, and cannot
move to the new global optimum.
```

Question 6 [15]. Game Playing

Consider the following game: there are three piles of sticks, with 1, 1, and 2 sticks in each pile, respectively. There are THREE players A, B, C who play in turn in that order. Each player can take one or more sticks from one and only one pile. The player who takes the last stick wins one dollar from each of the other two players (i.e., two dollars total).

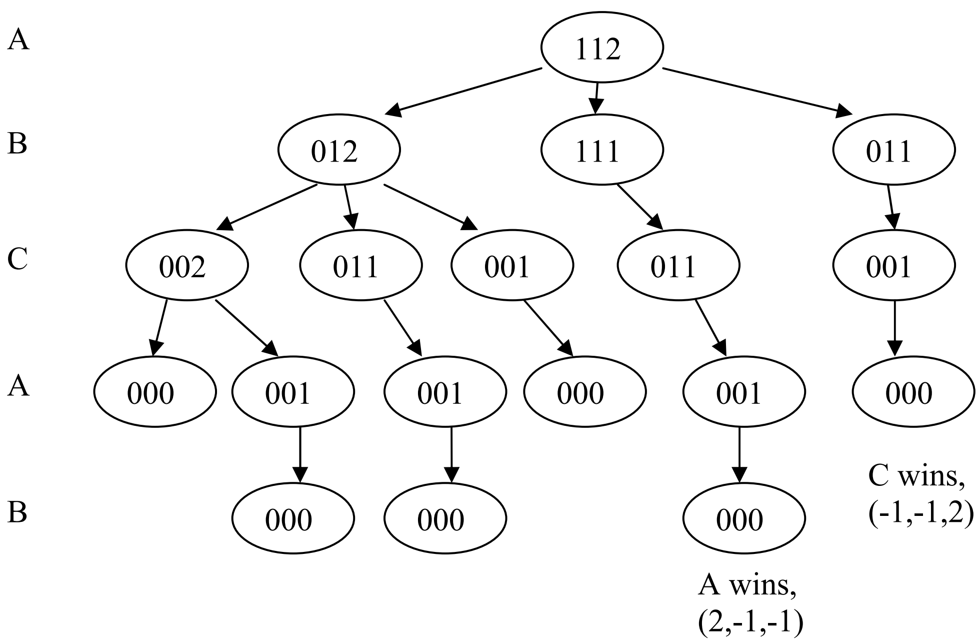
(a) [2] Circle the properties that describe this game:

two-player, zero-sum, discrete, deterministic, perfect information

(b) [4] How would you represent the game value of the leaf nodes in the game tree, now that there are three players?

```
Use a vector of three numbers.  For example, if A wins,
the game value can be represented as (2,-1,-1), which
reflects the money gained for each player.
```

(c) [6] Draw the complete game tree (hint: the piles (1,2,1) are equivalent to (1,1,2). You may want to always sort the numbers from small to large).

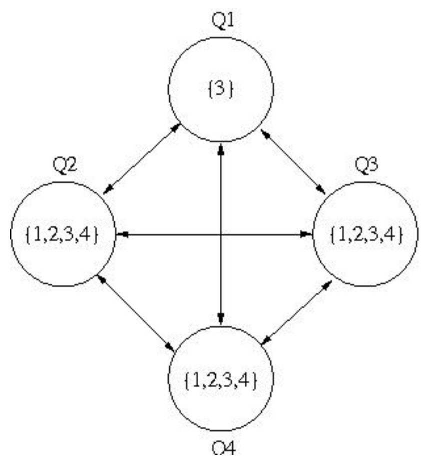


(d) [3] What should A's initial move be?

```
A should move to (111), i.e., take one stick from the
pile with two sticks.
```

Question 7 [15]. Constraint Satisfaction

Consider solving the 4-queens problem as a constraint satisfaction problem. 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 queen. One way to formulate this problem is to 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 the i^{th} queen is put somewhere in the i^{th} column, then the possible values in the domain for each variable are the row numbers in which it could be placed. Say we initially assign queen Q1 the unique value 3, meaning Q1 is placed in column 1 and row 3. This results in an initial constraint graph given by (the set of candidate values of each variable is shown inside the node):



(a) [5] Apply **forward checking** and give the remaining candidate values for the variables Q2, Q3 and Q4.

```
Q2:  {1}

Q3:  {2, 4}

Q4:  {1, 2, 4}
```

(b) [8] Fill in the table below with the candidate values of each queen after each of the following steps of applying the **arc consistency** algorithm to the figure. An arc "*x* → *y*" is consistent if for each value of *x* there is some value of *y* that is consistent with it.

	Q1	Q2	Q3	Q4
Initial domain	3	1,2,3,4	1,2,3,4	1,2,3,4
After Q2 → Q1	3	1	1,2,3,4	1,2,3,4
After Q3 → Q1	3	1	2,4	1,2,3,4
After Q2 → Q3	3	1	2,4	1,2,3,4
After Q3 → Q2	3	1	4	1,2,3,4

(c) [2] In general, when will the arc consistency algorithm halt?

The algorithm iterates until no more arc inconsistencies remain. This is checked by maintaining a queue of all arcs that need to be checked for consistency; if any value is deleted from the domain of a variable, then all arcs pointing to that variable must be added to the queue to be rechecked. The algorithm halts when the queue is empty.

Question 8 [15]. Propositional 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) [3] Given a domain using a vocabulary of 4 propositional symbols, A, B, C, and D, how many <u>models</u> are there for the following PL sentence: $(A \wedge B) \vee (B \wedge C)$

```
Of all 16 possible combinations of truth values for
the 4 symbols, there are 6 which make the above
sentence True.
```

(c) [3] Prove $(A \wedge B) \models (A \Leftrightarrow B)$ using a truth table.

```
Since for each row where the next to last column is T,
the last column is also T (this only occurs here for
the first row), entailment is proved.
```

A	B	$A \wedge B$	$A \Leftrightarrow B$
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	T

(d) [3] Given that a sentence α in PL is satisfiable but not valid, then which of the following must also be true? Your answer can be one or more of (i) – (iv).

(i) α is valid
(ii) $\neg\alpha$ is valid
(iii) $\neg\alpha$ is unsatisfiable
(iv) None of the above

```
(iv)
```

(e) [3] Prove whether or not the rule of inference $\dfrac{P \Rightarrow Q, \neg Q}{\neg P}$ is sound. Justify your answer by showing an appropriate truth table.

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

P	Q	P⇒Q	¬Q	(P ⇒ Q) ∧ ¬Q	¬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