CS 540: Introduction to Artificial Intelligence

Midterm Exam: 7:15-9:15 pm, March 12, 2013
Room 1240 CS Building

CLOSED BOOK
(one sheet of notes and a calculator allowed)

Write your answers on these pages and show your work. If you feel that a question is not fully specified, state any assumptions you need to make in order to solve the problem. You may use the backs of these sheets for scratch work.

Write your name on this page and initial all other pages of this exam (in case a page comes loose during grading). Make sure your exam contains ***six*** problems on ***ten*** pages.

Name ______________________________

Student ID ______________________________

Problem	Score	Max Score
1	______	15
2	______	20
3	______	14
4	______	14
5	______	12
6	______	25
TOTAL	______	100

Problem 1 – Decision Trees (15 points)

Assume that you are given the set of labeled *training examples* below, where each feature has three possible values *a, b,* or *c*. You choose to learn a decision tree and select "-" as the default output if there are ties.

F1	*F2*	*Output*
c	*a*	+
b	*c*	+
a	*b*	+
a	*a*	-
c	*b*	-
b	*b*	-

a) What score would the *information gain* calculation assign to each of the features?
Be sure to show all your work (use the back of this or the previous sheet if needed).

b) Which feature would be chosen as the root of the decision tree being built? ____________
(Break ties in favor of *F1* over *F2*.)

c) In a different data set, you have these *ages* and categories for the examples (as well as some other features not shown).

Age	Category
11	T
27	T
33	F
41	T
66	F
75	T
87	F

Which of these two 'derived' features below would be a better choice to add to a decision tree, according to the information-gain calculation? Be sure to show your work.

F1: Discretize *Age* into three (3) values: (i) *Age* < 30, (ii) *Age* ≥ 30 but < 65, (iii) *Age* ≥ 65

F2: Discretize *Age* into two values: (i) *Age* < 40, (ii) *Age* ≥ 40

Answer: ______________________

Problem 2 – Search (20 points)

a) Consider the search space below, where *S* is the start node and *G1*, *G2*, and *G3* satisfy the goal test. Arcs are labeled with the cost of traversing them and the estimated cost to a goal is reported inside nodes.

For each of the following search strategies, indicate which goal state is reached (if any) and list, *in order*, all the states *popped off of the OPEN list*. When all else is equal, nodes should be removed from OPEN in alphabetical order.

You can show your work (for cases of partial credit), using the notation presented in class, on the back of the previous page.

Beam Search (using f=h and beam width = 2)

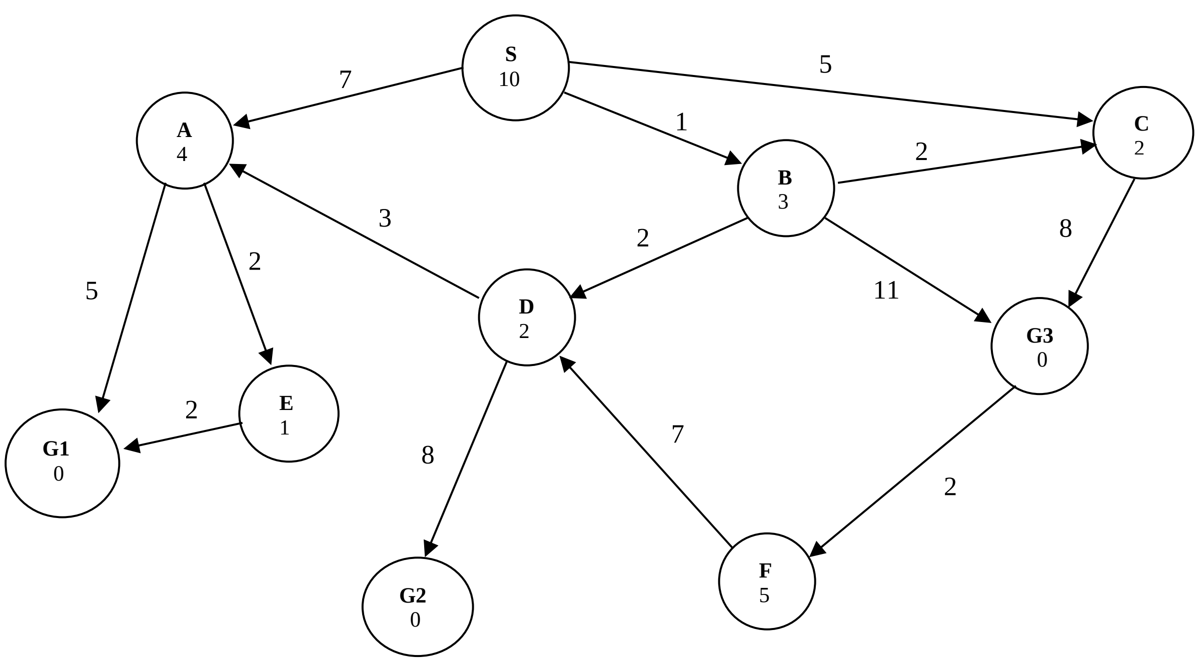
Goal state reached: ________ States popped off OPEN: ____________________________________

Iterative Deepening

Goal state reached: ________ States popped off OPEN: ____________________________________

A* (f = g + h)

Goal state reached: ________ States popped off OPEN: ____________________________________



b) (Note: This question talks about search spaces in general and is *not* referring to the specific search space used in Part a. Also, note there is much more white space on this page than needed to satisfactorily answer the question.)

Assume someone gives you *h1(node)* and *h2(node)*, both of which are admissible heuristics. Circle the expression you feel is a better choice for a heuristic function that *combines* the two. (Hint: what makes one admissible *h* function better than another?)

i) *h3(node)* = max(*h1(node)* , *h2(node)*)

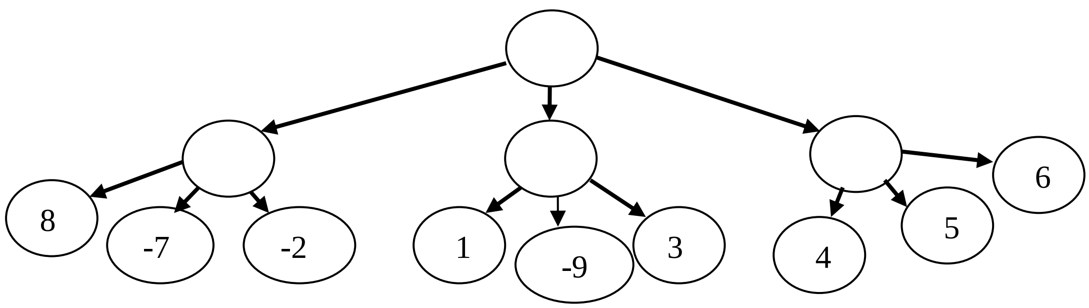
ii) *h3(node)* = min(*h1(node)* , *h2(node)*)

Briefly explain your answer below. A good explanation is needed for full credit.

Problem 3 – Game Playing (14 points)

a) Apply the *mini-max* algorithm to the partial game tree below, where it is the **maximizer's** turn to play and the game does not involve randomness. The values estimated by the static-board evaluator (SBE) are indicated in the leaf nodes (higher scores are better for the maximizer).

Write the estimated values of the intermediate nodes inside their circles and indicate the proper move of the maximizer by circling one of the root's outgoing arcs.



b) List one leaf node (if any) in the above game tree whose SBE score need not be computed. Explain why.

c) Briefly explain the *horizon effect* and why it matters for AI game playing.

Problem 4 – Multiple-Choice Questions (14 points)

Circle your chosen answer. Explaining your answers is optional but might lead to partial credit for wrong answers.

a) The *primary* purpose of the CLOSED list in search is to produce the path from the goal node found back to the initial state.

TRUE *FALSE* Why? ____

b) The probability of getting at least one inch of rain on June 1st is 0.25. How many bits of information do you expect to get when Weather Woman Wendy tells you on June 2nd whether or not it rained over an inch the day before?

Less Than 0.5 *Exactly 0.5* *More Than 0.5*

Supporting calculation: ____

c) You wish to compare your new machine learning algorithm to random (i.e., decision) forests. Your algorithm, called YOU, has an important parameter that users can set. Call this parameter θ and assume it only has three possible settings (1, 2, and 3).

You have three standard datasets used in machine-learning research, call then A, B, and C.

You run 1000 train-test partitions of these datasets and get the following average TEST SET accuracies. Assume the 'error bars' across the 1000 runs on each testset accuracy are 0.01% (and hence can be ignored).

Dataset	Random Forests	YOU with θ=1	YOU with θ=2	YOU with θ=3
A	79%	75%	81%	91%
B	62%	67%	56%	61%
C	89%	84%	93%	87%

Which interpretation of these results is best?

YOU is better because for EACH dataset SOME setting of θ produces results better than Random Forests.

Random Forests is better because for SOME setting of θ it beats YOU on EACH dataset.

Tuning sets should have been used to independently choose a good setting for θ for EVERY train-test partition (i.e., different runs can choose different θ values).

Θ should not be a settable parameter in YOU and instead should be always set to 3 because on average it produces the best accuracy.

Why?

Initials: ____________________

Problem 5 – Key AI Concepts (12 points)

Briefly describe each of the following AI concepts and explain each's significance.
(Write your answers *below* the phrases and clearly separate your description and significance.)

Pruning Decision Trees

description:

significance:

Simulated Annealing

description:

significance:

Creating Copies of Datasets by 'Randomly Drawing with Replacement'

description:

significance:

Problem 6 – Miscellaneous Questions (25 points)

a) Three bit-vectors in a genetic algorithm survive the phase where low-fitness bit-vectors 'die off'; these bit-vectors are *0101, 1110,* and *0011*. They have *fitness(0101)* = 57, *fitness(1110)* = 29, and *fitness(0011)* = 42. In *fitness-proportional reproduction*, what is the

probability(bit-vector 1110 is chosen as the first parent)

=

b) Formulate as state-based search the problem of changing some three-letter word (e.g., CAT) into a different three-letter word (e.g., DOG) by changing <u>one</u> (1) letter at a time, but only via three-letter words that appear in a given English dictionary (which you have in electronic form).

i. What would be a *state*?

ii. What would be an *arc*?

iii. What would be a good *heuristic function*? It is fine to assume your *h* function knows the words that are the initial state (e.g., CAT) and the goal state (e.g., DOG)

iv. Would your answer to Part (iii) be an *admissible* heuristic? Explain.

c) Draw a search space where *hill climbing* and *beam search with beam-width = 1* produce different results. Assume we wish to find the node with the highest *h(node)* value. Do not use more than four (4) nodes in your space. Briefly explain your answer.

Explanation:

d) Some two-player board game, which does not involve randomness, always has 10 legal moves at each turn. The *mini-max* algorithm can project (i.e., simulate) forward 4 moves per player in 1 minute and in that same amount of time *alpha-beta* pruning can project forward 7 moves per player.

For simplicity, assume games never end early; hence, at the leaves of all projected paths a call to the game's static-board evaluator (SBE) is made. The SBE always takes the same amount of time to regardless of the board configuration given to it. In other words, *mini-max* and *alpha-beta* always make the same number of SBE calls.

It is interesting to know how many SBE calls *alpha-beta* avoided that *mini-max* would have performed if *mini-max* had projected 7 moves per player. Compute the following ratio:

SBE calls *avoided* due to alpha-beta pruning when projecting 7 moves per player
____
SBE calls made by mini-max when projecting forward 4 moves per player

=