CS/ENGRI 172, Fall 2002
9/13/02: Homework One

How many AI people does it take to change a lightbulb?

Answer one: 5. One to define the goal state; one to define the operators; one to describe the universal problem solver; one to hack the production system; one to indicate about [sic] how it is a model of human lightbulb-changing behavior.

Answer two: 5. One to design a two-player game tree with the robot as one player and the lightbulb as another; one to write a minimax search algorithm that assumes optimal play on the part of the lightbulb; one to build special-purpose hardware to enable 24-ply search; one to enter the robot in a human lightbulb-changing tournament; one to state categorically that lightbulb changing is "no longer considered AI".

—excerpted from a piece attributed to Jeff Shrager

Due at the *beginning* of class on Monday, September 23. Hand in the parts *separately*, with your name on each part. Answers will be graded on both correctness and clarity. If a question asks for explanation, *no credit* will be given for answers without explanation.

This problem set requires a number of drawings. Please make your drawings and labels large enough and clear enough to be easily readable.

Part A

1. This problem examines the tradeoffs between depth-first and breadth-first search and the dependence of such tradeoffs on ordering issues when inducing a path tree from a problem space.

To save you some work, we've put the path trees referred to here on a separate page of this assignment, so you can detach that page and turn it in rather than spending time drawing trees.

First, consider the path tree in Figure 1, where we haven't been told which problem-space states (indicated in parentheses, as in class) are goal states.

(a) Assume that the problem space has no goal states. Redraw the tree (or use the attached copy) and indicate the order in which the pictured nodes would be visited/touched by depth-first search and breadth-first search. Use the "v/t" notation from class (writing in "d" or "r" optional).

(b) Is there a way to choose one of A, B, ..., E to be a goal state so that depth-first search is "faster", that is, visits fewer nodes than bfs touches? Explain your answer; you may find your answer to the above subproblem useful.

(c) Is there a way to choose one of A, B, ..., E to be a goal state so that breadth-first search is "faster", that is, touches fewer nodes than dfs visits? Explain your answer; you may find your answers to the first subproblem useful.

(d) Now, note that for the same problem space, we could have drawn a different path tree by putting each node's children in a different order. In particular, the path tree in Figure 2 represents exactly the same paths — all we have done is taken the "mirror image" of Figure 1's path tree.

For this "flipped" path tree, is there a way to choose one of A, B, ..., E to be a goal state so that depth-first search is "faster", that is, visits fewer nodes than bfs touches? Explain your answer; as part of your explanation, redraw (or use the attached copy of) this path tree and show the order in which dfs and bfs visit/touch the nodes using the v/t notation from class (writing in "d" or "r" is optional).

Part B

2. Depicted in Figure 3 (on a separate page for your convenience) is the top portion of a game tree for a two-player zero-sum game, where evaluation function values are indicated at the leaf nodes. Player 1 wishes to maximize their score.

(a) Which operator, α_1 or α_2, should Player 1 choose (or are they equally good)? Explain your answer by computing (some) minimax values via alpha-beta pruning. Show your work (on the attached copy, if you like) using the notation from class. As an aid to the graders, circle all the evaluation function values that are actually consulted (i.e. that you made use of) during the process.

Furthermore, for *each* node where you decided not to look at some of its children's values (i.e., nodes where *pruning* took place), give its Gorn number and explain why looking at all its children's values was unnecessary.

(b) It's natural to wonder how sensitive one's chosen strategy is to the quality of the evaluation function. Is there a way to change *just one leaf's* evaluation function value in the game tree of Figure 3 so that Player 1 chooses a different operator than the one you picked in the previous subproblem? (Or that, if the operators were equally good, that this one change makes one operator preferable?)

If it is possible to change the operator preferability, redraw the game tree (or use the second version on the attached copy) with exactly one leaf value changed (and very clearly marked). Then, explain why the operator preferabilities change; as part of your explanation, show your computation of (some) minimax values on your new tree.

If it is not possible to change operator preferability, explain clearly why this is the case.

Part C

In this part, we revisit the class scheduling problem we looked at earlier in lecture. Assume the following roster for all fall and spring semesters, present and future.

	Fall	Spring
9:00	MATH 471 PHYS 116	CHEM 211
10:00	ENGRI 172 MATH 191	MATH 191
11:00	ENGRI 123 MATH 191 MATH 454	FWS 270 MATH 293

The baseline goal is to end up with a two-semester, fall and spring schedule that contains at least two math (MATH) courses, one Introduction to Engineering (ENGRI) course, two science (PHYS or CHEM) courses, and a Freshman Writing Seminar (FWS). Furthermore, schedules in which two different courses are taken at the same time are disallowed. Some of the problems below may indicate further constraints or conditions on the goal states.

Note that schedules that seem intuitively unreasonable but which do not violate any stated constraints are to be considered. For instance, if not otherwise ruled out, schedules involving taking MATH 191 twice in the same semester are permitted.

3. Suppose the states are restricted to the following form:

$$[\text{math1: } m_1;\ \text{math2: } m_2;\ \text{engri: } e_1;\ \text{sci1: } s_1;\ \text{sci2: } s_2;\ \text{fws: } f_1]$$

where the italicized variables can take either the value $\perp$ (not yet taken) or the name of a course taken that matches the type specified (e.g., m_1 can be either $\perp$ or any MATH course). Furthermore, any variable with subscript 2 must be $\perp$ if the corresponding variable with subscript 1 is $\perp$ (convince yourself that this cuts down on the number of possible states). An example state is

$$[\text{math1: MATH 191; math2: MATH 293; engri: } \perp\text{; sci1: } \perp\text{; sci2: } \perp\text{; fws: FWS 270}].$$

The operators are restricted to be of the form "Add class x", where x is the name of a class (e.g. MATH 191).

Furthermore, we also impose the constraint of a maximum of two math courses per semester.

Is this partial problem-space definition (it's partial because we haven't yet given the initial and final states) valid? *If it is valid*, give the initial state and goal states. Then, draw all the reachable states and put in directed edges between states to show the effect of all the applicable operators (basically, you should create a diagram akin to the one we drew for the river-crossing problem of lecture three). Label each edge with the operator it corresponds to. *If the definition is not valid*, describe what problem or problems occur with this definition, and give a concrete example of each such problem.

4. In this problem, we remove the constraint from the previous problem regarding the math courseload cap, and instead add the following constraints: *exactly* three courses must be taken in each of the fall and spring semester, with no additional semesters allowed.

However, we now loosen the restrictions on representation. First, the operators must still correspond to the action of scheduling only one course at a time, but you may add extra information beyond what was allowed in the previous problem's operator representation. Second, you may choose a state representation.

Your task in this subproblem is to design state and operator representation schemes that *minimizes* the number of states reachable from the initial state (that you define).[1]

(a) Describe your choice of state and operator representation. Your representation must be consistent and make explicit all the information needed to summarize the corresponding situation completely, so that from the state and operator labels alone it is possible to determine which operators are applicable for a given state and indeed whether the state is even legal. Use the statement of the previous problem as a guide to the level of specificity we are looking for.

Note that you may not have conditions that essentially say, "no states that are unnecessary are included", since this is not determinable from the labels alone. However, you are allowed to make conditions such as "states with both slots of type x filled with the same value are disallowed", since such a condition is based just on the labels.

(b) Draw all the states in your problem space that are reachable from your initial state. Be sure to indicate initial and goal states. Put in directed edges between states to show the effect of all the applicable operators; label the edges with the operator name. How many states did you need?

(c) How was your choice of representation affected by the need to minimize the number of problem states? Explain the choices you made that would not be robust to changes we might make to the problem setting to make it more realistic. (We are expecting your answer to be something on the order of two to three paragraphs.)

[1] Postmortem note: some things we should have made clearer: all possible operator sequences leading to a goal state needed to be preserved; the initial state should have corresponded to a "blank slate" schedule (no courses filled in yet)".

Part D

Fill out and return the attached student information and waiver form.

Postmortem note: there are blank pages in what follows to allow the students to detach and turn in sheets separately.

CS/ENGRI 172 - PATH TREES FOR HW1, PART A

If you are turning in this sheet, put your name here:

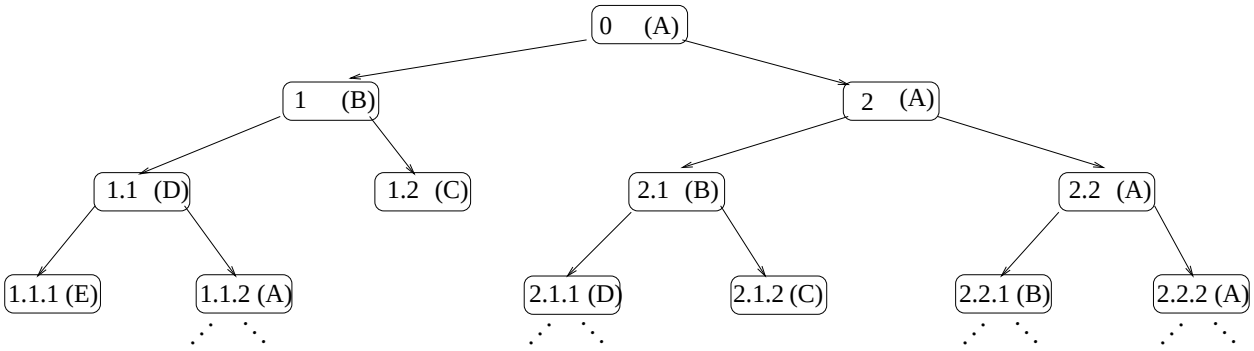


Figure 1: Path tree for the dfs/bfs problem.

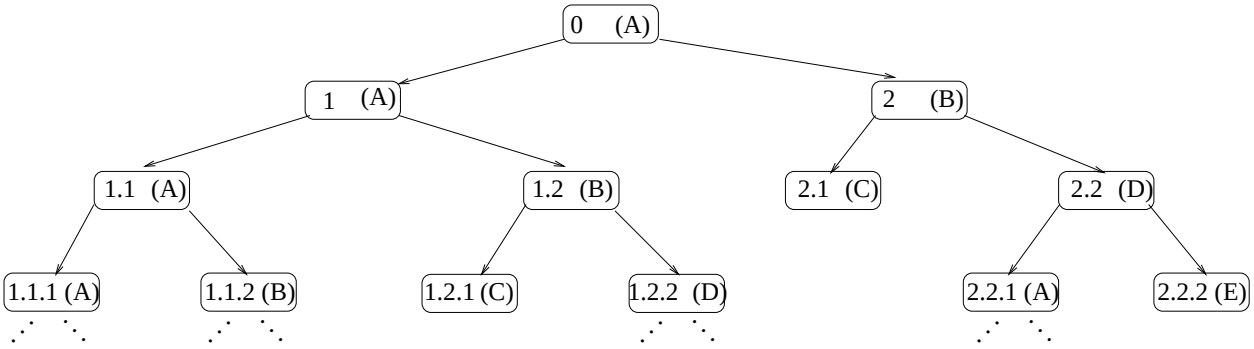


Figure 2: The second path tree — a "mirror image" of the above — for the dfs/bfs problem.

.

CS/ENGRI 172 - GAME TREES FOR HW1, PART B

If you are turning in this sheet, write your name here:

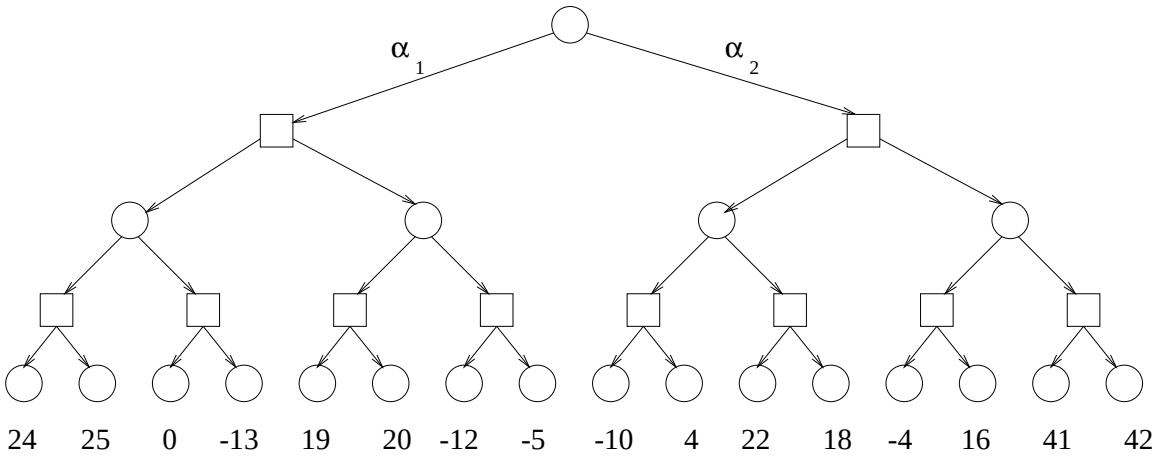
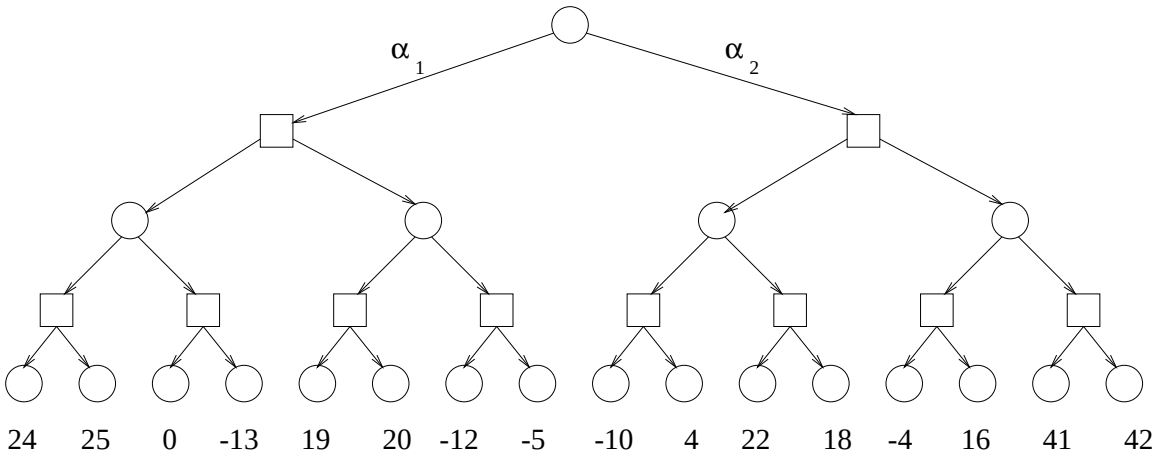


Figure 3: Game tree for the alpha-beta pruning question, with most operator labels omitted for clarity. ◯ and □ indicate next move for Player 1 and 2, respectively.

A copy of the above figure:



.

CS/ENGRI 172, Fall 2002: Computation, Information, and Intelligence
Student information and waiver form

Name:

ID number:

College (e.g. Arts and Sciences, Engineering) and year (e.g. first-year, sophomore):

Major or intended major (please indicate which):

Campus address and phone number:

Email address:

Check the appropriate box below:

- ☐ I consent to having my CS172 assignments returned in the batch return bins in Upson 303, thereby waiving my rights in this regard under the Family Educational Rights and Privacy Act of 1974. I understand that such consent is not required by the University, and I affirm that it is in all respects freely and voluntarily given.
- ☐ I do not consent to batch return. I will write "Private Return" directly below my name on my assignments.

Check the appropriate box below:

- ☐ I consent to the use of my student ID number for the purpose of posting grades for CS172 for the Fall 2002 semester, thereby waiving my rights in this regard under the Family Educational Rights and Privacy Act of 1974. I understand that such consent is not required by the University, and I affirm that it is in all respects freely and voluntarily given.
- ☐ I wish my grades to be posted by the following codename instead of my ID number (fill in codename):

Signature:
