

Name: \_\_\_\_\_

## **CSCI E-220: Artificial Intelligence Midterm Exam, Fall Term 2001**

This is a 75-minute, open-book exam. Take a minute to look over the exam, and begin by answering the questions you feel most comfortable with. If you start a problem and have trouble with it, go on to another one. If you've spent more than 15 minutes on a question, go on to another one, and come back to the earlier questions if you have time at the end. Write all answers on these pages. You may use the backs of the pages if you need more space for any problem, but clearly number any such continuations.

### **1. Lisp**

[15 pts.] Write a **recursive** function **EVENS** in CommonLISP that returns every other element of a list, beginning with the second. That is, the function returns all the elements in even-numbered positions in the list. For example, the function call

(EVENS ' (A B C D E F G))

should return the value (B D F). The function call (EVENS ' (A)) should return NIL. You don't have to error-check the input (that is, you can assume you're getting a list). Your function must not have any side effects.

## 2. State-space Problem Representation

Consider the following problem (adapted from an old book of puzzles):

A robot has an empty 9-gallon jug and an empty 4-gallon jug. A water tap and a drain are available. The robot needs to get exactly 6 gallons of water into the 9 gallon jug. The robot can perform the following primitive functions only:

1. Completely fill a jug from the water tap (regardless of how much water is in it already);
2. Completely empty a jug down the drain;
3. Pour from one jug into the other until either
  - (a) the first jug is empty, or
  - (b) the second jug is full.

For instance, if the big jug had 7 gallons in it and the small jug had 1, the robot could pour 3 gallons from the big jug into the small jug (filling the small jug), or could pour 1 gallon from the small jug into the big jug (emptying the small jug), but could not pour any other amount from one into the other. The robot could also fill either jug to the brim, or empty either jug.

### **Set this problem up as a state-space problem, as follows:**

(a) [5 pts.] Begin by choosing a representation for a state. (What are the important bits of information you need to represent?). Describe and/or demonstrate your state representation.

(b) [7 pts.] Using the representation and notation you have chosen:

What is the start state?

What is/are the goal state(s)?

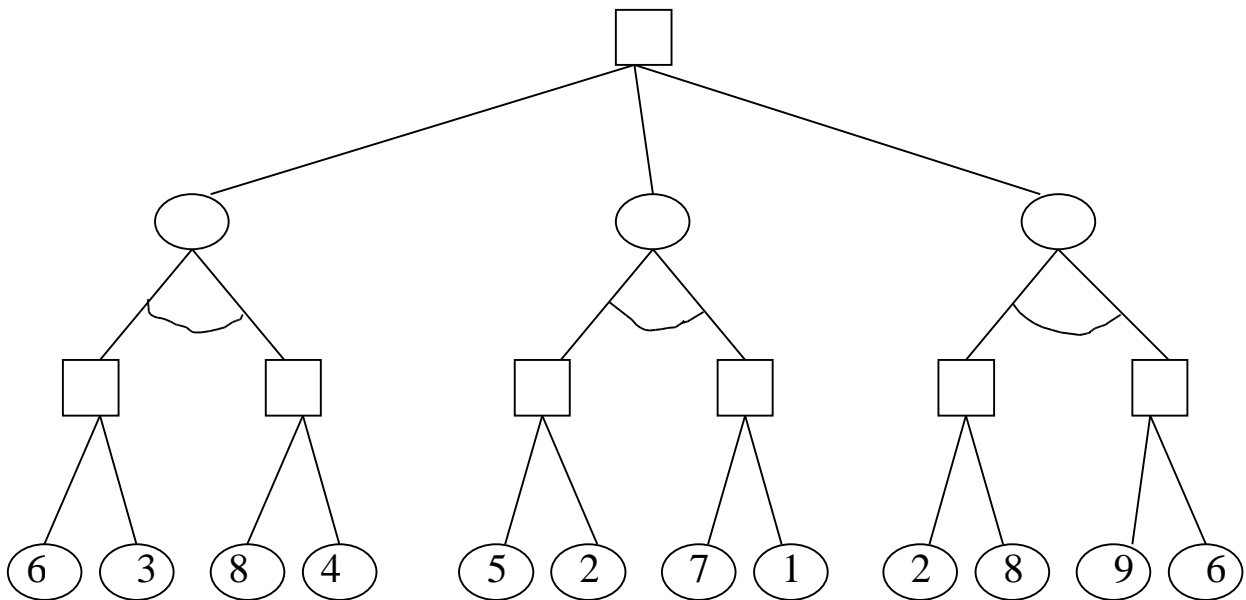
How many different possible states are there, in total?

(continued on next page)

(c) [10 pts.] Define all the operators that can be used to change one legal state into another. **Your operators must be defined in terms of your particular state representation** (using variables and mathematical notation to avoid writing out every single state transformation).

### 3. Minimax Search with Alpha-Beta Pruning

[15 pts.] In the tree below, clearly show the final backed-up values that would be calculated at each node if you were performing a left-to-right Minimax search of this tree, using Alpha-Beta pruning. As with the examples in class, show with a double strike-through line any nodes which would be pruned, and do not continue to calculate below the pruning marks. Put an X through nodes at which values are not calculated. Do this for leaf nodes as well (the values listed are those that would be calculated if necessary, but your algorithm might never actually get to them!) Finally, show with an arrow which move would actually be made from the root OR node.



#### 4. Knowledge Representation

Here is a set of facts that one might want to represent in some type of knowledge base:

- (1) Encyclopedias and dictionaries are (kinds of) books.
- (2) Webster's Third is a dictionary.
- (3) Britannica is an encyclopedia.
- (4) Every book has a color property.
- (5) Red and green are colors.
- (6) All Dictionaries are red.
- (7) Encyclopedias are never red.
- (8) The Britannica encyclopedia is green.
- (9) Every book is either an encyclopedia or a dictionary but not both.

Read all four parts below (a, b, c, and d) before beginning:

(a) [12 pts.] Represent the set of facts above using an inheritance net (with IS-A and property links, as described in class). Be sure to explain the meaning of every kind of link or node symbol you use. If there are any facts that cannot easily be represented, leave them out. Your representation should be flexible enough to enable you to answer part (c) below.

(b) [5 pts.] In the previous part, you had to leave out one or more facts. Explain briefly why it/they could not be represented using inheritance networks.

(c) [5 pts.] Using your knowledge representation formalism, briefly explain the algorithm you would use to answer the question "What color is Webster's Third?" What is the answer?

(d) [5 pts.] If you had set this problem up using predicate calculus instead of an IS-A hierarchy, could you have represented the fact(s) that were not representable in part (a)? If so, show a reasonable way to represent those specific facts. Be sure to provide **explicit semantic interpretations** for any relations or constants you use.

