CSE 341 Programming Assignment: Lisp

Due: March 3

In the **Missionaries and Cannibals Problem** there are n missionaries and n cannibals standing on the left bank of a river who all want to cross over to the other side. There is also a boat on the left bank that can hold up to two people. The boat can be used to move missionaries and cannibals, either one or two at a time, across the river. The boat could hold one missionary, one cannibal, two missionaries, two cannibals, or one missionary and one cannibal on its trip.

The problem is to find a sequence of boat trips that achieve moving ALL the missionaries and cannibals to the right side. One difficulty is that if the number of cannibals is greater than the number of missionaries on either side, the cannibals will eat the missionaries, which isn't allowed.

This problem can be formalized as a state-space search. The states in the problem each have the form (M,C,S) where

- M is the number of missionaries on the left bank
- C is the number of cannibals on the left bank
- S is the side the boat is on

For example, for the n=3 problem, the *initial state* is (3,3,L) and the *goal state* is (0,0,R). There are 10 possible boat operations that can change the states: MCR, MMR, MR, CCR, CR, MCL, MML, ML, CCL, and CL. MCR means the boat has one missionary and one cannibal in it and moves from the left bank to the right bank. CCL means the boat has two cannibals in it and moves from the right bank to the left bank. Note that the boat movement is *between* states; there is no state with some people on the left, some on the right, and some in the boat; that's just a transition.

A state (M,C,S) is **dead** if there are more cannibals than missionaries on either side OR it is a state that has been encountered before in the search. For the n=3 problem, a state (M,C,S) is dead if

1. C > M

2. 3-C > 3-M

3. There is an ancestor state (M',C',S') of (M,C,S) where M=M', C=C', and S=S'.

A **solution** to the n=3 state-space problem is a sequence of operators that leads from the initial state (3,3,L) to the goal state (0,0,R). In this assignment, the solution also includes all the intermediate states. For example, one solution is:

(3,3,L) CCR (3,1,R) CL (3,2,L) CCR (3,0,R) CL (3,1,L) MMR (1,1,R) MCL (2,2,L) MMR (0,2,R) CL (0,3,L) CCR (0,1,R) CL (0,2,L) CCR (0,0,R)

Assignment:
The assignment is to implement in Common Lisp a search program that is given the number of missionaries and cannibals (n of each) and finds all solutions to the state space problem of how to get from the initial state (n,n,L) to the goal state (0,0,R) by performing a depth first search using recursion. The program should print out each solution as it finds it. A solution consists of both the states encountered and the operators that produced those states. So, for example you might print

(3,3,L) MCR (2,2,R) ML (3,2,L) etc.

You should NOT use the standard depth first search procedure found in some books, which uses a queue. You should instead program the specific problem (ie. Missionaries and Cannibals) directly from the statement of the problem. You do not need an explicit tree structure; the recursion can handle it. You will probably, however, need a separate stack or list or some mechanism for remembering what states you have been in as you come down the tree, so that you can determine dead states and so you can easily print the whole solution when you get to the goal node.

Your program should have a main routine that handles interaction with the user and initialization. That should call the search procedure, which is recursive. The search procedure will call on the procedures for the 10 operators and any other utility procedures it needs.