Dylan Hutchison
CS 541 Artificial Intelligence
HW 1 due 19 September 2012

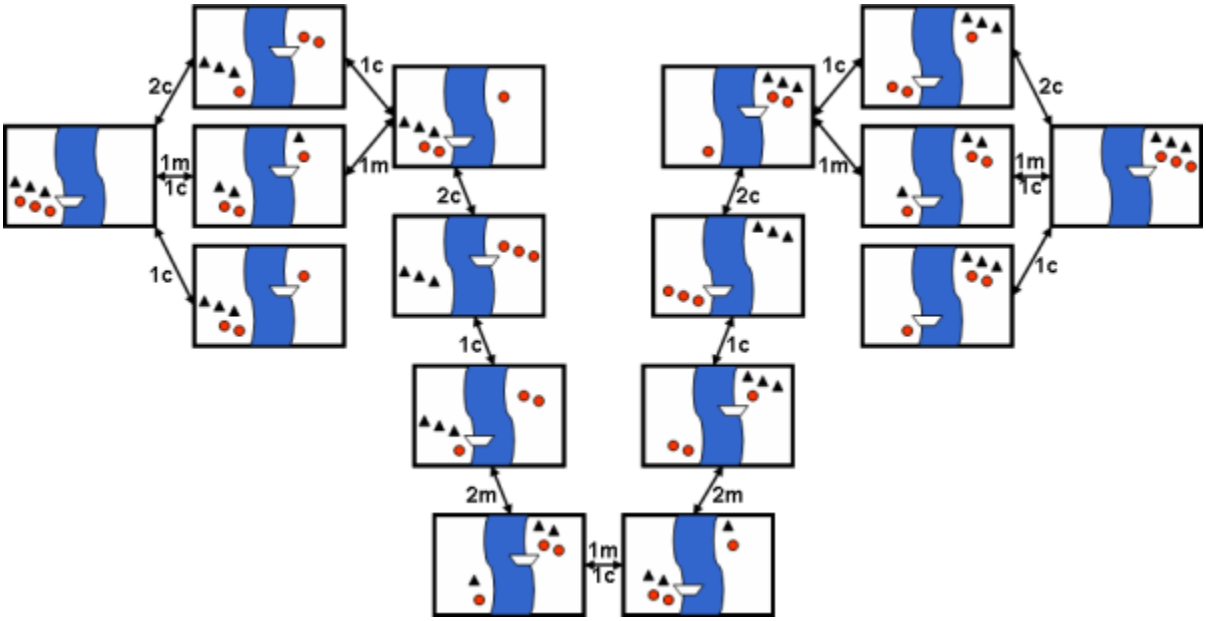
Problem 1: Russel & Norvig Book Problem 3.9

a. The missionaries and cannibals (M&C) problem is a single state problem because the initial state is a fully known single state. The environment is deterministic (how actions change the state is known), fully observable (the entire state is known at all times), discrete, episodic (each state can be viewed independently), and there is a single agent. I define the state with 3 variables:

1. LC, the number of cannibals on the left side of the river (number on right = 3-LC)
2. LM, the number of missionaries on the left side of the river (number on right = 3-LM)
3. L or R, a 1-bit variable representing the current location of the boat (left or right side of the river)

- Initial state: `{LC:3,LM:3,L}`
- Goal state: `{LC:0,LM:0,R}`
- Actions/successors: moving up to 2 people from the side with the boat to the other side of the river. Such a move is legal only if neither side results in missionaries outnumbered by cannibals.
- Path cost: 1/move (assuming our goal is to minimize the number of moves required to reach the goal state from the initial state)
- Solution: a sequence of actions from the initial state to the goal state of minimum length

Rather than enumerate the complete state space myself, I include here a nice diagram of the complete state space posted by Dr. Gerhard Wickler, a professor at the School of Informatics at the University of Edinburgh, at this location: http://www.aiai.ed.ac.uk/~gwickler/missionaries.html



b) I implemented a program to find an optimal solution to the M&C problem in the D programming language. Please see MissCanAI\src\MissCanAI.d inside the MissCanAI.zip project archive for implementation details. Here is an overview of the program copied from the source for your convenience:

> Finds an optimal solution to the Missionaries and Cannibals problem with 3 missionaries and cannibals on one side of the river.
>
> Approach: Graph-based Uninformed Bidirectional search. Alternates expanding state nodes from both the initial state and the goal state until the two searches meet at a common state in the middle. The solution sequence of actions is constructed from the two search sequences. Remembers which states have been explored in the past to prevent loops by storing them in an associative array.
>
> Search Strategy: Breadth First Search - guaranteed to find an optimal solution as the first solution found will have minimum cost.

To run the program, just execute the already compiled binary in bin/MissCanAI.exe (compiled on 64-bit Windows 7 machine). To compile the program from source, you will need the DMD compiler which comes with the Phobos standard library. The compiler and the standard library are available here: http://dlang.org/download.html It may be possible to use another compiler, but I have not tested it.

Here is the output of the program:

```
SOLUTION: [{LC:3,LM:3,L}, {LC:1,LM:3,R}, {LC:2,LM:3,L}, {LC:0,LM:3,R}, {LC:1,LM:3,L},
{LC:1,LM:1,R}, {LC:2,LM:2,L}, {LC:2,LM:0,R}, {LC:3,LM:0,L}, {LC:1,LM:0,R},
{LC:2,LM:0,L}, {LC:0,LM:0,R}]
```

c) People often try to apply heuristics to problems like these that “seem right,” but really result in a line of thinking that loops between states by taking actions that undo each other. A purely logical AI, on the other hand, can detect these loops and always try every possible action, whereas a human might overlook an action.