

CSCI5511
Programming assignment

Wei-Wei Wu
Wuxx1045
5052973

Due 12/5/17
Professor Nikos Papanikolopoulos

Files:

cannibals.lisp
puzzles.lisp

Missionary and cannibals problem

Program Implementation Description

For this problem we can use a breadth first search to do a level order traversal of the state tree. We hard coded all possible valid combinations of missionaries and cannibals on the boat (actions). We then use breadth first search. We expand every node in the queue and check that those states are valid.

A state is valid if the number of missionaries outnumber the number of cannibals, the number of missionaries is the same as the number of cannibals, or if the number of missionaries is zero. Each state can be expanded by all the actions, so I wrote a helper function to generate all valid successor states given an input of a state. We expand every node currently in the queue and place the new paths in a temp queue. At the end of the level, we replace queue with the temp queue. And at the end of each level we toggle the flag to tell if we are subtracting for adding actions in generating successors.

We represent the state as the number of missionaries on the left side, number of cannibals on the left side, if the left side has the canoe. For example, (3, 3, 1) would mean that there are 3 missionaries, 3 cannibals, and a canoe on the left side. Since we only keep track of the state on the left side. We alternate between adding or subtracting the possible actions. Since the boat is either leaving or coming to the left side. When we validate a state. We make sure that the counts on both sides are valid.

To keep track of the correct path, I store all current traversed paths in the queue. And create a new path with a valid successor state in the beginning to add to the queue at every node expansion.

If the most current state in the path is the end state where there is no cannibals or missionaries on the left side, then we exit the function by returning the current path. In the case of infeasible puzzles, the breadth first search will return nil.

Known bugs and deficiencies

None. It works for the 15 and 24 count problems

Terminal script

Run

clisp cannibals.lisp

Also more info on how to run in README.md

Output

15 missionaries and 15 cannibals

Starting: left, m: 15 c:15

Starting: right, m: 0 c:0

move 0 missionaries and 6 cannibals from left to right.

Result: left, m: 15 c: 9

Result: right, m: 0 c: 6

move 0 missionaries and 1 cannibals from right to left.

Result: left, m: 15 c: 10

Result: right, m: 0 c: 5

move 0 missionaries and 2 cannibals from left to right.

Result: left, m: 15 c: 8

Result: right, m: 0 c: 7

move 0 missionaries and 1 cannibals from right to left.

Result: left, m: 15 c: 9

Result: right, m: 0 c: 6

move 6 missionaries and 0 cannibals from left to right.

Result: left, m: 9 c: 9

Result: right, m: 6 c: 6

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 10 c: 10

Result: right, m: 5 c: 5

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 8 c: 8

Result: right, m: 7 c: 7

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 9 c: 9

Result: right, m: 6 c: 6

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 7 c: 7

Result: right, m: 8 c: 8

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 8 c: 8

Result: right, m: 7 c: 7

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 6 c: 6

Result: right, m: 9 c: 9

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 7 c: 7

Result: right, m: 8 c: 8

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 5 c: 5

Result: right, m: 10 c: 10

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 6 c: 6

Result: right, m: 9 c: 9

move 6 missionaries and 0 cannibals from left to right.

Result: left, m: 0 c: 6

Result: right, m: 15 c: 9

move 0 missionaries and 1 cannibals from right to left.

Result: left, m: 0 c: 7

Result: right, m: 15 c: 8

move 0 missionaries and 6 cannibals from left to right.

Result: left, m: 0 c: 1

Result: right, m: 15 c: 14

move 1 missionaries and 0 cannibals from right to left.

Result: left, m: 1 c: 1

Result: right, m: 14 c: 14

move 1 missionaries and 1 cannibals from left to right.

Result: left, m: 0 c: 0

Result: right, m: 15 c: 15

24 missionaries and 24 cannibals

Starting: left, m: 24 c: 24

Starting: right, m: 0 c: 0

move 0 missionaries and 6 cannibals from left to right.

Result: left, m: 24 c: 18

Result: right, m: 0 c: 6

move 0 missionaries and 1 cannibals from right to left.

Result: left, m: 24 c: 19

Result: right, m: 0 c: 5

move 0 missionaries and 2 cannibals from left to right.

Result: left, m: 24 c: 17

Result: right, m: 0 c: 7

move 0 missionaries and 1 cannibals from right to left.

Result: left, m: 24 c: 18

Result: right, m: 0 c: 6

move 6 missionaries and 0 cannibals from left to right.

Result: left, m: 18 c: 18

Result: right, m: 6 c: 6

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 19 c: 19

Result: right, m: 5 c: 5

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 17 c: 17

Result: right, m: 7 c: 7

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 18 c: 18

Result: right, m: 6 c: 6

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 16 c: 16

Result: right, m: 8 c: 8

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 17 c: 17

Result: right, m: 7 c: 7

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 15 c: 15

Result: right, m: 9 c: 9

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 16 c: 16

Result: right, m: 8 c: 8

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 14 c: 14

Result: right, m: 10 c: 10

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 15 c: 15

Result: right, m: 9 c: 9

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 13 c: 13

Result: right, m: 11 c: 11

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 14 c: 14

Result: right, m: 10 c: 10

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 12 c: 12

Result: right, m: 12 c: 12

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 13 c: 13

Result: right, m: 11 c: 11

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 11 c: 11

Result: right, m: 13 c: 13

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 12 c: 12

Result: right, m: 12 c: 12

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 10 c: 10

Result: right, m: 14 c: 14

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 11 c: 11

Result: right, m: 13 c: 13

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 9 c: 9

Result: right, m: 15 c: 15

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 10 c: 10

Result: right, m: 14 c: 14

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 8 c: 8

Result: right, m: 16 c: 16

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 9 c: 9

Result: right, m: 15 c: 15

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 7 c: 7

Result: right, m: 17 c: 17

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 8 c: 8

Result: right, m: 16 c: 16

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 6 c: 6

Result: right, m: 18 c: 18

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 7 c: 7

Result: right, m: 17 c: 17

move 2 missionaries and 2 cannibals from left to right.

Result: left, m: 5 c: 5

Result: right, m: 19 c: 19

move 1 missionaries and 1 cannibals from right to left.

Result: left, m: 6 c: 6

Result: right, m: 18 c: 18

move 6 missionaries and 0 cannibals from left to right.

Result: left, m: 0 c: 6

Result: right, m: 24 c: 18

move 0 missionaries and 1 cannibals from right to left.

Result: left, m: 0 c: 7

Result: right, m: 24 c: 17

move 0 missionaries and 6 cannibals from left to right.

Result: left, m: 0 c: 1

Result: right, m: 24 c: 23

move 1 missionaries and 0 cannibals from right to left.

Result: left, m: 1 c: 1

Result: right, m: 23 c: 23

move 1 missionaries and 1 cannibals from left to right.

Result: left, m: 0 c: 0

Result: right, m: 24 c: 24

9-puzzles problem

Program Implementation Description

For this problem, I implemented a A* algorithm to find the solution to the puzzle. The heuristic used is the number of numbers that are in different positions from the ending state. The puzzle is organized in a 3x3 grid with numbers from 1 to 8 and with an empty space.

I chose to represent each state as a vector of length 9 with the empty space being the number 0. Based on the numbers position in this vector we can calculate the row and column that it belongs to in the grid. And we can also calculate the position from the row and column values. I chose to use this representation because it was easier to manage in Lisp. For example, if the position is 5, the row is $\text{floor}(5/3) = 1$ and the column is $5 \bmod 3 = 2$. Vice versa, given the row and column, the position in the vector is $(3 * \text{row}) + \text{column} = 5$.

I also defined a state node structure to hold the state, G value, and H value. The body of the A* search algorithm is very similar to the breadth first search algorithm. The queue used for this algorithm holds a list of list of state nodes (path). At every iteration, we get the path in the queue that has the lowest f value for the most recent node. If this node is the end state then we return the current path.

We expand this node and add each successor node to a copy of the current path and then add the paths to queue. If the puzzle is unable to be solved, the A* algorithm will return nil.

Known bugs and deficiencies

None. It works for the given starting state and an infeasible state

Terminal script

Run

clisp puzzles.lisp

Also more info on how to run in README.md

Output

E 1 3

4 2 5

7 8 6

1 E 3

4 2 5

7 8 6

1 2 3

4 E 5

7 8 6

1 2 3

4 5 E

7 8 6

1 2 3

4 5 6

7 8 E

Number of nodes expanded: 4

infeasible puzzle

1 1 1

1 1 1

1 1 E

Puzzle cannot be solved

Number of nodes expanded: 9