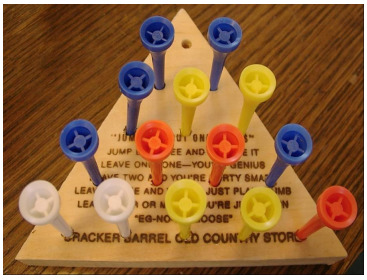
CSCE420: Introduction to Artificial Intelligence
Programming Assignment 1 : Enumerating State Spaces

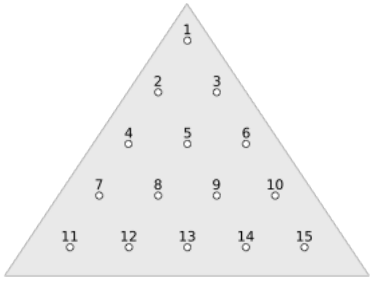
Dylan Shell

January 13th, 2014

1 Problem Domain

The Cracker Barrel restaurant chain (which has an instantiation on Briarcrest Drive) has a peg-in-hole solitaire game to bemuse, befuddle, and frustrate customers. The game involves fifteen holes and fourteen pegs. Each peg fits in a single hole. The following photo and diagram shows their standard board:





The board starts with an open spot in one of the fifteen holes (in the photo's configuration, spot #1). Moves are performed by hopping one peg over another into an open hole. Any peg that has been hopped over is removed from the board. The moves must be in a straight line parallel to one of edges of the board. In the arrangement above, only two moves are legal:

- The peg at position 4 may be hopped over peg 2 and placed in hole 1. Peg 2 would be removed from the board.
- The peg at position 6 may be hopped over peg 3 and placed in hole 1. Peg 3 would be removed from the board.

2 Assignment

Write a program to generate a tree representing all possible sequences of moves. You can use any programming language of your choice (although, do read the final section of this document before making a selection). You need to submit code to answer two questions:

Starting in the initial configuration shown in the photograph (*i.e.,* with the #1 spot empty),

1. how many distinct ways are there to end with a single peg *anywhere?*
2. how many distinct ways are there to end with a single peg *in a corner?*

3 Submission

Due date: 22 January at 11:59pm.
Submission method: Via e-mail to the professor.

Submit (in electronic form) the following:

1. Turn in a zip file which includes all the source files and a documentation like pdf file, with the items below. (Do not describe your results in the e-mail)
   2. The zip file should be named `student-last-name_hw-number.zip`
   e.g., `jones_hw1.zip`
   3. Subject of e-mail should be named as `[csce420]last-name_hw-number`
   e.g., `[csce420]jones_hw1`

The zip should include the following:

1. Output from your program. This should consist of two numbers, one for each of the questions above.
2. The code you wrote for this assignment.
3. A description of how to compile and run the submission.
4. A list of the resources used (*e.g.*, online forums, links to example code on the web, *etc.*).
5. A statement of the Aggie Code of Honor.

You may discuss this openly with your friends and classmates, but are expected to write your own code and compile your submission independently. If in doubt about whether a resource you used should be included in item 4 above, err on the side of caution and include it.

Bonus/Extra credit

An additional 10% will be added to your project score if your implementation is in one of the following functional languages Scheme, Lisp, Common Lisp, ML, Haskell. The first three of these programming languages were partly developed in order to solve these types of search problems by early AI practitioners. They have convenient list data structures and efficient recursion mechanisms.

CMU Common Lisp is installed on `sun.cs.tamu.edu` and a basic howto is available at via the CSE wiki (search for "LISP"). Also the following resource may be useful:

`http://www.apl.jhu.edu/~hall/Lisp-Notes/Introductory-Lisp-Overview.ps`