Playing With Mazes

David B. Suits
Department of Philosophy
Rochester Institute of Technology
Rochester NY 14623

david.suits@rit.edu

2. Maze Construction

A maze is said to be **connected** if there is a path from any cell to any other cell. We want all mazes (which, henceforth, shall be only closed, RFP mazes) to be connected so that any two cells may be designated the start and goal cells with the guarantee that the maze is solvable. How can this guarantee be obtained? The algorithm to *solve* a maze — to find a path from the start cell to the goal cell — would be inefficient to construct a maze. For suppose we have some random maze, i.e., each side of each cell in the maze has randomly either a door or a wall, with the provision that neighboring cells have their adjacent sides the same type — both doors or both walls. (We will also assume in all of our discussions — unless otherwise indicated — that the exterior sides of the border cells of the maze are always to be walls. This is only another way of saying that the maze is closed.) Such a maze is of course easily constructed. Now, a maze solving algorithm will work only if there really is a solution path. Consequently, if the algorithm reports failure, we know that the maze is unconnected. In that case, we may remove a wall at random and run the algorithm again. And we continue removing walls until the algorithm reports success, which it will eventually do. But even given a solution path in the maze, we still have no guarantee that it is a connected maze, which requires a path from *any* start cell to *any* goal cell. We seek, therefore, a more efficient algorithm. The following are only some of the possible methods.

2.1 The Spanning Tree Algorithm

1. Choose the size (width and height) of the maze, and make it fully unconnected, i.e., no doors in the maze. Mark all cells as *unvisited*.
2. Choose a cell at random and call it the *present* cell, and mark it as visited (i.e., part of the tree).
3. For each neighbor (up to four of them) of the present cell, if it is marked as unvisited, mark it as *frontier*.
4. Choose any frontier cell in the maze at random and make it the present cell; connect it to any (random) neighbor marked visited. (Step 3 guarantees that every frontier cell will have at least one visited neighbor; and it is possible that repeated applications of step 3 might have created frontier cells with more than one visited neighbor.) If there are no frontier cells in the maze, go to step 6.
5. Mark the present cell as visited, and go to step 3.
6. The maze is now connected.

The spanning tree algorithm will generate a singly connected maze, that is, a maze with one and only one path from any cell to any other cell. (See figure 8.) Sometimes, however, it may be required to construct a maze with multiple paths between cells. In that case, take the connected maze and randomly remove a few walls (except for exterior walls). The resulting maze will be multiply connected.

2.2 Anderson's Algorithm

Peter Anderson (Dept. of Computer Science, Rochester Institute of Technology) suggested to me the following method of maze construction. It is basically the spanning tree algorithm, but its function is not to connect unconnected cells, but rather to "grow" a set of walls which, when complete, will result in a connected maze.

1. Begin with a maximally connected maze, i.e., a maze with no walls (except, of course, along the exterior of the maze).
2. Randomly pick any cell corner to which no wall is attached. If there is no such corner, stop; the maze is finished.
3. If possible, construct a wall from that corner to a randomly chosen neighboring corner, provided the neighboring corner is part of a wall. (I.e., connect a new wall to an already existing wall.)
4. Go to step 2.

Figure 9 shows a five by five cell maze in the process of construction according to Anderson's algorithm.

As in the spanning tree algorithm, Anderson's algorithm generates a singly connected maze. Multiple connections may be formed, as before, by randomly removing some walls. (An option is to add step 1.5: Randomly choose any door and make it a wall. It is possible — but not likely — that if that initial wall is not connected to the maze border, then the resulting maze might be multiply connected, namely, by having a complete path around the interior of the outside border of the maze.)

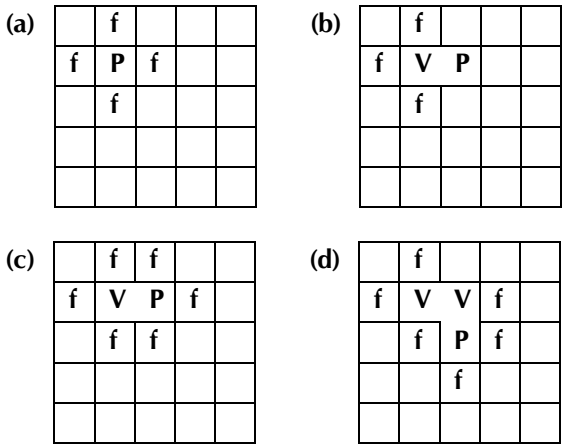


Figure 8. A maze being constructed by the Spanning Tree Algorithm. "P" represents the present cell, "f" indicates the frontier cells, and "V" represents visited cells, i.e., cells in the solution path. (a) After step 2. (b) After the first time through step 4. (c) After the second time through step 3. (d) After the third time through step 3.

2.3 Manipulating a Connection Matrix

A connection matrix is a two dimensional array indicating which cells are connected to which other cells. Figure 10b, for example, is the connection matrix for the maze in figure 10a. An entry of "1" indicates a door between the cells whose numbers appear at the left column and the top row. Each cell is trivially connected to itself. Is there some way to manipulate a connection matrix so that it will eventually represent a

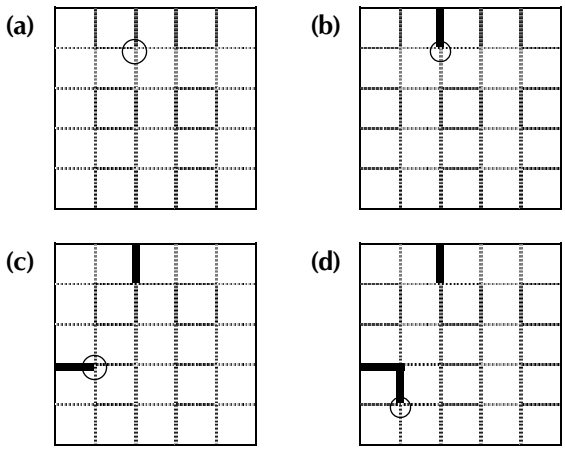
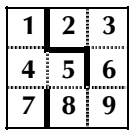


Figure 9. A maze being constructed by Anderson's algorithm. (a) After the first time through step 2. (b) After the first time through step 3. (c) After the second time through step 3. (d) After the third time through step 3.



(b)

	1	2	3	4	5	6	7	8	9
1	1	0	0	1	0	0	0	0	0
2	0	1	1	0	0	0	0	0	0
3	0	0	1	0	0	1	0	0	0
4	1	0	0	1	1	0	1	0	0
5	0	0	0	1	1	0	0	1	0
6	0	0	1	0	0	1	0	0	1
7	0	0	0	1	0	0	1	0	0
8	0	0	0	0	1	0	0	1	1
9	0	0	0	0	0	1	0	1	1

Figure 10. (a) A 3×3 maze. (b) The connection matrix for the maze in (a).

connected maze? That is, are there properties of connection matrices which are necessary and sufficient for the representation of connected mazes? There certainly are a few necessary conditions which we may notice at once: (1) The matrix must be symmetrical about the main diagonal. (This is not necessary for mazes with one-way doors.) (2) Every row must have at least two "1" entries (i.e., each cell must be connected to at least one cell other than itself.) Hence, because of the symmetry, each column must have at least two "1" entries. (3) Some connections are impossible (for RFP mazes), as shown by the shaded areas in figure 11.

Unfortunately, those three conditions are not by themselves sufficient to guarantee a connected maze, and it seems unlikely to be able easily to deduce the missing conditions. That is, a random placement of "1" entries in the matrix according to the three conditions will not guarantee a connected maze.

One reason for that is that each entry in the matrix represents where a maze traveler could go *in one move*, and the three conditions above tell us only that a traveler could go *somewhere* in one move. What we need, then, is matrix telling us where a traveler could go in n moves, where n is the longest of all the longest paths from each cell to every other cell, which is to say that n is equal to one less than the number of cells in the maze.

More information is possible if the matrix is multiplied by itself. Let M^1 be the name of the connection matrix, and let $M^2 = M^1 \bullet M^1$. Then each entry in M^2 will indicate the number of paths of *two* moves from a given cell to all others. Here's why: Figure 12 shows the multiplication of two matrices, A and B, yielding the new matrix C. Notice that the subscripts on each element of each maze indicate two cells which are being examined, and the value of the element is the number of ways to get from the first cell (a_{ij}) to the second (b_{ij}) in one step (that is, both A and B represent M^1). Let's take a look at one of the entries in C, say c_{12}. The element a_{11} is the number of ways to get from cell 1 to cell 1 in one step (and so its value is 1 — i.e., the maze traveler might stay put, as mentioned before). The element b_{12} is the number of ways to get from cell 1 to cell 2 in one step (and the value is 1, because cell 1 is connected to cell 2). We must add to that the product of a_{12} (the number of ways to get from cell 1 to cell 2 in one step) with b_{22} (the number of ways to get from cell 2 to cell 2 in one step) and the product of a_{13} (the number of ways to get from cell 1 to cell 3 in one step) with b_{32} (the number of ways to get from cell 3 to cell 2 in one step). The result is therefore the number of ways to get from cell 1 to cell 2 in two steps. An inspection of the M^2 matrix for the maze in figure 13 will serve to illustrate the result.

Now if we multiply M^2 by M^1, we shall get M^3, representing the number of ways to get from each cell to all others in three steps. Continuing in this fashion, we will eventually obtain M^8, which

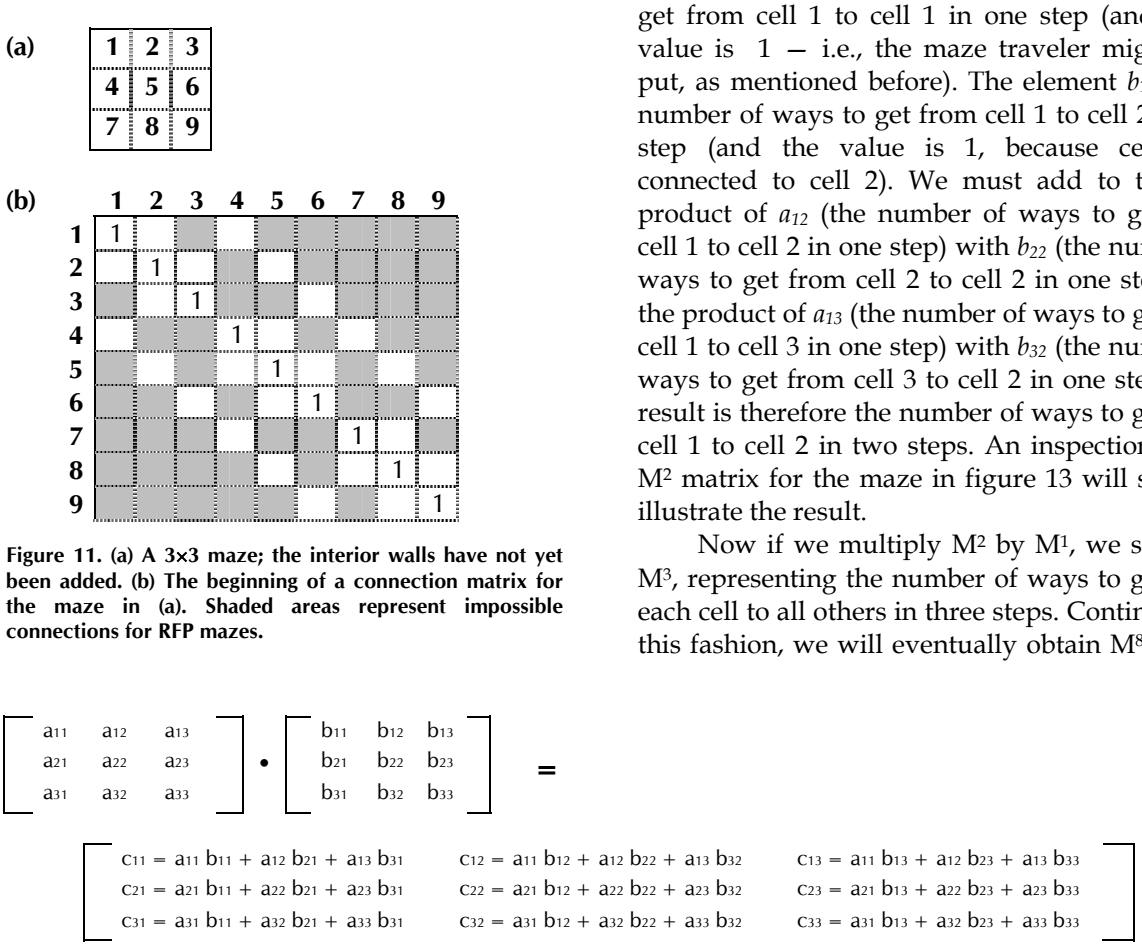


Figure 11. (a) A 3×3 maze; the interior walls have not yet been added. (b) The beginning of a connection matrix for the maze in (a). Shaded areas represent impossible connections for RFP mazes.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \bullet \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} =$$

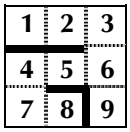
$$\begin{bmatrix} c_{11} = a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & c_{12} = a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} & c_{13} = a_{11}b_{13} + a_{12}b_{23} + a_{13}b_{33} \\ c_{21} = a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & c_{22} = a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} & c_{23} = a_{21}b_{13} + a_{22}b_{23} + a_{23}b_{33} \\ c_{31} = a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31} & c_{32} = a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32} & c_{33} = a_{31}b_{13} + a_{32}b_{23} + a_{33}b_{33} \end{bmatrix}$$

Figure 12. The general form of matrix multiplication.

will represent the number of ways to get from each cell of every other cell in 8 steps. For the maze in figure 13, there are 9 cells in the maze, and so 8 steps ought to be sufficient to get from any cell to any other cell, provided the maze is connected. Consequently, if every entry in M^8 is non-zero, the maze is connected, as is the case for the maze in figure 13.

If, however, M^8 tells us that the maze is not connected (see the example in figure 14), then we can do something reasonable to make it connected. An unconnected maze is one wherein there are two or more regions (of one or more cells each) which are inaccessible from each other. Choose any two *neighboring* cells in the maze whose entry in M^8 is zero. These will be neighbors on either side of the border between two such mutually inaccessible regions. Change the maze so that the two cells (and hence the two previously unconnected regions) are now

(a)



(b)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	1	1	0	0	0	0	0	0	0
2	1	1	1	0	0	0	0	0	0
3	0	1	1	0	0	1	0	0	0
4	0	0	0	1	1	0	1	0	0
5	0	0	0	1	1	1	0	0	0
6	0	0	1	0	1	1	0	0	1
7	0	0	0	1	0	0	1	1	0
8	0	0	0	0	0	0	1	1	0
9	0	0	0	0	0	1	0	0	1

(c)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	2	2	1	0	0	0	0	0	0
2	2	3	2	0	0	1	0	0	0
3	1	2	3	0	1	2	0	0	1
4	0	0	0	3	2	1	2	1	0
5	0	0	1	2	3	2	1	0	1
6	0	1	2	1	2	4	0	0	2
7	0	0	0	2	1	0	3	2	0
8	0	0	0	1	0	0	2	2	0
9	0	0	1	0	1	2	0	0	2

(d)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	127	197	196	28	77	161	7	1	70
2	197	323	358	84	189	343	29	7	161
3	196	358	470	190	350	547	84	28	273
4	28	84	190	393	385	350	330	189	162
5	77	189	350	385	477	548	266	134	273
6	161	343	547	350	548	750	189	77	386
7	7	29	84	330	266	189	316	196	77
8	1	7	28	189	134	77	196	127	27
9	70	161	273	162	273	386	77	27	204

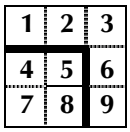
(e)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	324	520	554	112	266	504	36	8	231
2	520	878	1024	302	616	1051	120	36	504
3	554	1024	1375	624	1087	1640	320	112	820
4	112	302	624	1108	1128	1087	912	519	512
5	266	616	1087	1128	1410	1648	785	400	821
6	5041	1051	1640	1087	1648	2231	616	266	1136
7	36	120	302	912	785	616	842	512	266
8	8	36	112	519	400	266	512	323	104
9	231	504	820	512	821	1136	266	104	590

Figure 13. (a) A 3×3 maze. (b) M^1 for the maze. (c) M^2 for the maze. (d) M^7 for the maze. (e) M^8 for the maze.

connected. Compute M^8 again. Continue the process until every entry in M^8 is non-zero. Actually, it might not be necessary to compute M^8 if all entries in some M^i, $i < 8$, are non-zero. (See M^7 in figure 13d.) To generalize, if n is the number of cells in a maze, then the maze is connected if there is some $i \leq n-1$ such that all of the entries in M^i are non-zero. Such an M^i we will call the **complete connection matrix** and denote by M^c. Matrices derived from M^c will be used later in Section 4.

(a)



(b)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	1	1	0	0	0	0	0	0	0
2	1	1	1	0	0	0	0	0	0
3	0	1	1	0	0	1	0	0	0
4	0	0	0	1	1	0	1	0	0
5	0	0	0	1	1	0	0	0	0
6	0	0	1	0	0	1	0	0	1
7	0	0	0	1	0	0	1	1	0
8	0	0	0	0	0	0	1	1	0
9	0	0	0	0	0	1	0	0	1

(c)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	2	2	1	0	0	0	0	0	0
2	2	3	2	0	0	1	0	0	0
3	1	2	3	0	0	2	0	0	1
4	0	0	0	3	2	0	2	1	0
5	0	0	0	2	0	0	1	0	0
6	0	1	2	0	0	3	0	0	2
7	0	0	0	2	1	0	3	2	0
8	0	0	0	1	0	0	2	2	0
9	0	0	1	0	0	2	0	0	2

(d)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	323	512	517	0	0	384	0	0	195
2	512	840	896	0	0	712	0	0	384
3	517	896	1035	0	0	6	0	0	3
4	0	0	0	7	5	0	6	3	0
5	0	0	0	5	4	0	3	1	0
6	1	3	6	0	0	7	0	0	5
7	0	0	0	6	3	0	7	5	0
8	0	0	0	3	1	0	5	4	0
9	0	1	3	0	0	5	0	0	4

Figure 14. (a) The same maze as in figure 13a, but with a wall between cells 5 and 6, making the maze unconnected. (b) M^1 for the maze. (c) M^2 for the maze. (d) M^8 for the maze.