A Brief Mathematical Analysis of *To the Moon*'s Memento Puzzles

CJ Dowd

September 21, 2023

1 Intro

To the Moon is short story-based adventure game about a pair of scientists enlisted to fulfill the dying wishes of old man, Johnny Wyles. To do so, they travel through Johnny's memories, with the goal of altering them to restructure Johnny's memory of his own life in a manner consistent with his wishes. They leap from memory to memory via mementos, objects of great personal importance to Johnny. Upon finding each memento, the player must solve a puzzle to proceed.

I've always found the existence of this puzzle minigame rather odd, since (a) the player's performance on the puzzle has no consequences whatsoever and (b) it seems stylistically out of place with the rest of the game. However, I did find solving the puzzle interesting enough to make me think about how to solve it algorithmically, so I began to analyze it mathematically. It turns out that the game is fairly easy to model with linear algebra over $\mathbb{Z}/2\mathbb{Z}$.

The goal of this short essay is to outline an algorithm that solves any memento puzzle automatically and efficiently. The maximum grid size reached in *To the Moon* is 5×5, but we generalize to puzzle grid of arbitrary size. We also demonstrate how to determine whether a given puzzle is unsolvable, though every puzzle presented in-game is solvable.

2 Rules

The puzzle takes place on a rectangular grid of tiles containing an image. The size of this grid varies between 3×3 to 5×5 over the course of the game, and is not necessarily square. Each tile has two sides, with one side blank and the other side containing the correct piece of the image; we'll say the a tile is "facing up" if the image is displayed and "facing down" if the blank side is displayed. On each move, the player can do one of three things:

1. Flip every tile in a given row.
2. Flip every tile in a given column.
3. Flip every tile on the lower-left to upper-right diagonal (which from now on I will simply call the "diagonal").

Initially, some tiles are facing up and some tiles are facing down. The goal of the game is to transform the grid from its initial configuration into a complete image using a minimal number of moves. That is, the player must perform a sequence of moves as above so that every tile is facing up.

3 Analysis

A couple preliminary observations:

- Every move is reversible, since performing any move twice in a row undoes it.
- The order of the player's moves does not matter.
- A given move never needs to be made more than once, since based on the previous two observations, performing a particular move twice is the same as not performing it at all. Therefore we will assume that any solution contains no repeated moves.

We model the game as follows. Suppose the game is being run on an $n \times m$ grid. We model the current state of the game as an element of the mn-dimensional $\mathbb{Z}/2\mathbb{Z}$-vector space $V = \mathrm{Mat}_{n\times m}(\mathbb{Z}/2\mathbb{Z})$, where $\mathbb{Z}/2\mathbb{Z} = \{0,1\}$ is the field with two elements and $\mathrm{Mat}_{m\times n}(\mathbb{Z}/2\mathbb{Z})$ is the set of $n \times m$ matrices with entries in $\mathbb{Z}/2\mathbb{Z}$. We assign each tile a matrix coordinate in this vector space corresponding to its position on the game board. If a tile is facing up, we associate the value 0, and if it is facing down, we associate the value 1, so that any given game state has a unique representation as an element $v \in V$.

Performing a move corresponds to adding an appropriate vector to the current game state. In contrast to standard mathematical convention, we'll order the matrix rows starting from the bottom in order to be consistent with the notation in the puzzle, e.g. the first row of a matrix is the bottom row and the diagonal goes from the lower-left to the upper-right.

- Let e_i denote the $n \times m$ matrix whose i-th row consists of all 1's and all other entries 0. Flipping the i-th row in the puzzle corresponds to adding e_i to the current game state.
- Let f_i denote the $n \times m$ matrix whose i-th column consists of all 1's an all other entries 0. Flipping the i-th column in the puzzle corresponds to adding f_i to the current game state.
- Let d denote the matrix with 1's on the diagonal and all other entries 0. Flipping the anti-diagonal corresponds to adding d to the current game state.

Letting v_0 be the initial game state, the goal of the game is to find the smallest subset $I \subseteq \{e_1, \ldots, e_n\} \cup \{f_1, \ldots, f_m\} \cup \{d\}$ such that

$$v_0 + \sum_{v \in I} v = 0,$$

or equivalently

$$v_0 = \sum_{v \in I} v,$$

recalling that we are working modulo 2. We may represent the latter equation in matrix form. Take an ordered basis for V to treat $V \cong (\mathbb{Z}/2\mathbb{Z})^{nm}$, so that the e_i, f_i, and d take the form of nm-tuples of 0's and 1's. Let M denote the column matrix

$$M = \begin{pmatrix} e_1 & e_2 & \ldots & e_n & f_1 & f_2 & \ldots & f_m & d \end{pmatrix},$$

which has dimensions $nm \times (n+m+1)$. Solutions to the puzzle (not necessarily with the minimum number of moves) correspond to solutions $a \in (\mathbb{Z}/2\mathbb{Z})^{n+m+1}$ to

$$Ma = v_0.$$

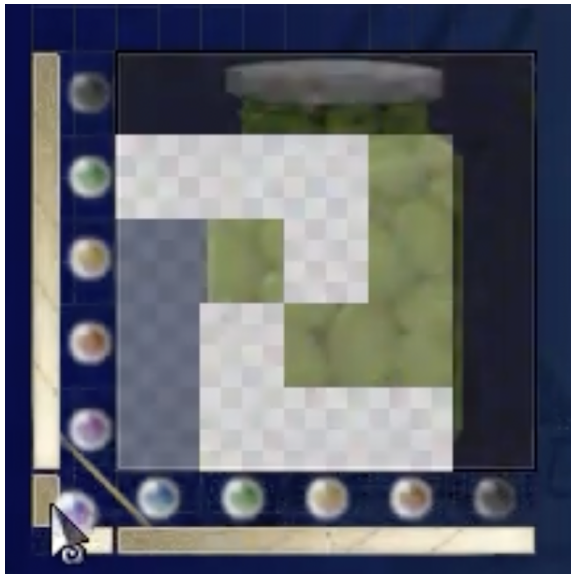
A solution vector a corresponds to the following collection of moves. For every coordinate of a that is equal to 1, if the index i of the coordinate is:

- $1 \le i \le n$: Flip the i-th row.
- $n+1 \le i \le m+n$: Flip the i-th column.

- $i = n + m + 1$: Flip the anti-diagonal.

Therefore finding an arbitrary solution can be done efficiently using standard techniques in linear algebra, e.g. Gauss-Jordan elimination.

As an example, consider the 5th memento puzzle in the game:



The grid is 4×4, and the matrix M is the 16×9 matrix

$$M = \begin{pmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1
\end{pmatrix}$$

The puzzle can be completed in 3 moves by flipping the first row, the fourth row, and the diagonal. The corresponding initial state vector v_0 and the solution vector a are

$$v_0 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} \qquad a = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

and indeed we have $Ma = v_0 \bmod 2$.

However, we wish to find the solution with the minimal number of moves, which is equivalent to finding a solution vector a that is as sparse as possible. Any two solutions a, a' differ by an element of $\ker M$, so we turn to analysis of this kernel.

4 Optimizing the solution

4.1 General case

We'll start with the case of a sufficiently large puzzle, when $n, m \geq 3$. We claim that $\ker M$ is 1-dimensional, spanned by the vector $(1, 1, \ldots, 1, 0)$ consisting of all 1's except for the last entry. Any element of the kernel corresponds uniquely to a sequence of moves, with no repetitions, that ultimately leaves the puzzle unchanged. This means that every tile must be flipped an even number of times.

We'll first assume that the puzzle is square, with $n = m$. Suppose that the sequence of moves is nonempty; since flipping the diagonal alone is clearly not a solution, the solution must contain at least one row or column flip. By symmetry, without loss of generality we may assume that this move is e_1, flipping the first row. For $i \neq 1$, there are only two ways to flip the $(1, i)$ tile: the moves e_1 or f_i. Hence, all moves $f_2, \ldots, f_n$ must also be contained in the move set. Since we are assuming $n \geq 2$, this means in particular that f_2 and f_3 are in the moveset, and from this we similarly deduce that $e_2, \ldots, e_{n-1}$ are also contained in the in the moveset. Finally, the existence of e_2 in the moveset similarly implies that f_1 is also contained in the moveset. This means that any nontrivial solution must perform every row and column move, which flips every tile exactly twice, giving one nontrivial element of the kernel. We've shown that every nontrivial element of the kernel must have its first $2n$ entries all equal to 1, so since the all 1's vector $(1, 1, \ldots, 1, 1)$ is evidently not an element of the kernel, we conclude that $\ker M = \langle (1, 1, \ldots, 1, 0) \rangle$.

If the puzzle is not square, say without loss of generality $n \geq m$, then by considering the $m \times m$ subgrid we conclude that the moves $e_1, \ldots, e_m$ and $f_1, \ldots, f_m$ must all be part of the solution, from which we easily conclude that the solution must also contain $e_{m+1}, \ldots, e_n$. Thus the same conclusion follows.

Hence, assuming the puzzle is solvable, it has two solutions, differing by $(1, 1, \ldots, 1, 0)$. We can find the minimal solution by comparing which of these contains fewer 1's, which is easy to do by inspection. Note that an arbitrary initial configuration need not be solvable. For $n, m \geq 3$, the rank of M is $m+n+1-1 = m+n$, so the column space of M does not span all of $(\mathbb{Z}/2\mathbb{Z})^{mn}$. Of course, all puzzles given in the game are solvable. It might have been an interesting addition by the developers to include an unsolvable puzzle for purposes related to the story. Such a puzzle could be proven unsolvable by demonstrating that the corresponding state vector v_0 does not lie in the image of M, which is doable by standard linear algebraic techniques.

4.2 Case of small grids

Suppose either $n = 1$ or $m = 1$; without loss of generality $m = 1$. We can describe an optimal solution directly: if strictly more than half of the tiles are facing down in the initial configuration, perform the unique column move. Then, perform row moves to flip over the remaining face-down tiles one at a time. We can ignore the diagonal move since it is the same move as flipping the first row.

Now suppose $n, m = 2$. We have

$$M = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

We can verify directly that this matrix has 2-dimensional kernel $\ker M = \langle (1, 1, 1, 1, 0), (1, 0, 1, 0, 1) \rangle$. Hence, if any solutions exist, there are always exactly four solutions, and again finding the optimal one is a matter of inspection, which is particularly easy in this case since the number of possibilities is small.

Now suppose, without loss of generality, that $m = 2$ but that $n \geq 3$. By considering the 2×2 subgrid, a nontrivial element of the kernel must include one of the move sequences $\{e_1, e_2, f_1, f_2\}$, $\{e_1, f_1, d\}$, or $\{e_2, f_2, d\}$. But if not all of the columns are in the solution, there is no way to flip the tiles on rows 3 and higher more than once, so we conclude that both column moves f_1, f_2 must be present, and therefore all row moves must be present. This implies that the solution does not involve d, so we obtain the same result as the general case with $n, m \geq 3$.