From Puzzles to Proof-Writing
Supplementary Materials

Tara Davis, Lauren Grimley, Kenan İnce, Gizem Karaali, Boyan Kostadinov, Roberto Soto

Supplementary materials include solutions to selected problems and implementation notes.

Peg Solitaire

We tried the peg solitaire problems as an opening activity in an inquiry-based, introduction to proofs course with a class of 9. The purpose of including peg solitaire in the course was to help students think creatively and formally with their mathematical ideas. On the first day of the course, we briefly described the original game, its generalization to path graphs, and asked that the students try to justify which starting state on a path graph in two colors are solvable. Students quickly decided that even and odd path graphs behaved differently for comparable starting conditions. Students also quickly decided that the starting state 0111...1 was not solvable (forgetting the case on P_3) and that 1011...1 was a more likely solution. Students struggled for a couple of classes to formalize the argument for this starting state. After a quick introduction, students were sent home to complete the listed problems, to be presented in subsequent classes.

Problem 1 generated a conversation about whether or not non-playable starting states (eg. 101 on P_3) should be counted. Students generally agreed that games which could be created by a rotation were equivalent, for example 012 was equivalent to 120 on C_3. However, students struggled with the idea that boards in which colors were swapped (eg. 012222 vs 021111) should be considered equivalent. Students also struggled with the idea that a starting state with only two colors (eg. 011111) could be considered peg solitaire on three colors.

We have also engaged in undergraduate research projects about peg solitaire on graphs, and we often start students out working on problems similar to those given in this text. We find that students struggle with the distinction between a *winning game* and a *solvable graph*. In particular, they struggle with the fact that to prove that a graph is solvable, one must only provide one game together with a sequence of moves that wins that game on the graph. They often confuse this when trying to show that a game on a graph loses, by providing one game and one sequence of moves that loses; however, this is not sufficient to prove that the game actually loses.

Here are some hints and partial solutions to the problems.

1. The following example is the technique for listing the games on a path graph P_3 in 3 colors. It can be generalized to other graphs and other numbers of colors. We begin with the game having a hole in vertex v_1 and color 1 everywhere. We then use a process akin to binary addition to incrementally list all games, moving the hole to the right as needed: $011, 012, 021, 022, 101, 102, 201, 202, 110, 120, 210, 220$. We note that some games will be mirror images of others due to symmetry present in the graph. Also, one only needs to list the first half of the games with the hole in a given position, because when switching the roles of the 1 and the 2 one obtains equivalent games.

2. The proof of part *b.* is in the paper [1]. For part *c.*, we will illustrate an example that could be used as a student hint to show that the answer is "no." Consider the game 0111 in P_4 in 4 colors. There is only one move at each point, and the game goes through the intermediate states 1201 and 0311 until reaching the losing terminal state 1001. The problem becomes more complicated on path graphs with more vertices, but such graphs in more than three colors are ripe for student discovery.

3. The path is solvable in all colors.

4. It is proved in [1] that P_n is not freely solvable for $n \geq 3$.

5. List all the games and show that we can win every game. Note that many games are equivalent and we only have to play two games, 011 and 012.

6. C_4 is freely solvable whereas C_5 is not.

7. Note that in part *a.*, there are only two games that need to be played: one with the hole in the center vertex, and one with a hole at one of the outer vertices. For part *b.* there are a total of $2^5 = 32$ games but after all of the equivalent games are eliminated, there are 4 games with a hole in the center and 6 games with a hole at one of the outer vertices that need to be played.

8. Have fun!

9. This is discussed in [1].

Lights Out!

The third and fifth authors have supervised undergraduate research projects exploring the linear algebra of Lights Out! using Python and SageMath. Students with at most a linear algebra background notably were able to make progress using Python and SageMath to compute the null spaces and column spaces of the adjacency (toggle) matrices corresponding to Lights Out! on various size grids, including with boundary identifications. Additionally, there are several open questions involving playing Lights Out! on various surfaces (see Section 12.3.7) that may be suitable for undergraduate research projects.

Below, we describe strategies and solutions to selected problems from Section 12.3.7.

1. **Listing games on grid graphs.**

   (a) Assuming we are not restricted to solvable initial configurations, and ignoring any symmetry, the number of possible initial configurations for Lights Out! modulo k on a graph $G = (V, E)$ is $k^{|V|}$. However, the symmetry of $G_{3,3}$, for example, renders certain of these initial configurations equivalent.

   (b) An initial configuration $\boldsymbol{b}$ on a graph G is solvable (resp., open solvable) modulo k if and only if there exists a solution $\boldsymbol{x}$ to the matrix-vector equation $(A_G + I)\boldsymbol{x} = -\boldsymbol{b} \mod k$ (resp., $A_G\boldsymbol{x} = -\boldsymbol{b} \mod k$), where A_G is the adjacency matrix of the graph. In other words, $\boldsymbol{b}$ is solvable if and only if $\boldsymbol{b}$ is in the column space of the appropriate matrix.

   (c) Given any grid graph $G = G_{m,n}$, we can form the corresponding *domino graph* $D(G)$ by adding an edge between vertices v, w if and only if v and w are either in the same row or the same column. Then an initial configuration $\boldsymbol{b}$ is domino solvable modulo k on G if and only if $\boldsymbol{b}$ is in the column space of the adjacency matrix of $D(G)$.

2. **Domino presses on $BLO(3, 2)$.**

   (a) Denote by $O(n)$ the $n \times n$ all-ones matrix. Then the *adjacency matrix* for domino Lights Out! on a 3×3 grid is given by:

$$DBL_3 = \begin{bmatrix} O(3) & I_3 & I_3 \\ I_3 & O(3) & I_3 \\ I_3 & I_3 & O(3) \end{bmatrix}$$

(b) In general, for domino Lights Out! on an $n \times n$ grid, a given initial configuration $\boldsymbol{b}$ is solvable if and only if $\boldsymbol{b}$ lies in the column space of the block matrix:

$$DBL_n = \begin{bmatrix} O(n) & I_n & \dots & I_n & I_n \\ I_n & O(n) & \ddots & \ddots & I_n \\ I_n & \ddots & \ddots & \ddots & I_n \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ I_n & I_n & \cdots & I_n & O(n) \end{bmatrix}$$

3. **Playing all games on C_4 and C_5.** On both C_4 and C_5, all game states can be reached from any other game state via some number of button presses. In particular, the all-zero state can be reached from any initial configuration, hence Lights Out! is completely solvable modulo 2 on C_4 and C_5. **Bonus questions**: characterize the solvability of C_n modulo 2 for as many n as possible. Characterize the solvability of C_4 and C_5 modulo k for as many k as possible. **Hint**: the adjacency matrix of C_n, where each vertex is considered adjacent to itself, is circulant with all row and column sums equal to 3. As such, its determinant is divisible by 3. Since $\det(A_{C_4}) = -3$ and $\det(A_{C_5}) = 3$, both of which are invertible modulo 2, the column spaces of both adjacency matrices are full rank modulo 2, meaning any two light states are equivalent modulo 2 under the operation of button pressing.

4. **Characterizing solvability of $BLO(n,k)$, $ULO(n,k)$, and $CLO(n,k)$.** Let n and k be positive integers with $n \geq 3$. Then, by [2], there exists a k for which $BLO(n,k)$ is not completely solvable if and only if $n \equiv 4 \mod 5$ or $n \equiv 5 \mod 6$. Also by [2], if n is a multiple of 5 or 6 and k is arbitrary, then $ULO(n,k)$ is not completely solvable. For each n that is neither a multiple of 5 nor a multiple of 6, there is a k_1 such that $ULO(n,k_1)$ is universally solvable and a k_2 such that $ULO(n,k_2)$ is not completely solvable.

   Finally, by unpublished work of the second and third authors, the adjacency matrix CL_n of the game $CLO(n,k)$ of cylindrical Lights Out! has zero determinant if and only if $n \equiv 2$ or $3 \mod 6$, and hence $CLO(n,k)$ is not completely solvable for any k when $n \equiv 2, 3 \mod 6$.

10 **Computing with SageMath**: The 16×16 toggle (adjacency) matrix A_4 is given by:

$$A_4 = \begin{pmatrix} C_4 & I_4 & O_4 & O_4 \\ I_4 & C_4 & I_4 & O_4 \\ O_4 & I_4 & C_4 & I_4 \\ O_4 & O_4 & I_4 & C_4 \end{pmatrix}, \text{where } C_4 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}. \tag{1}$$

Here, I_4 is the 4×4 identity matrix and O_4 is the 4×4 zero matrix in (1). We can represent the lights configuration in Figure **??** by the matrix $B_4 \in M_4(\mathbb{Z}_2)$:

$$B_4 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \tag{2}$$

We flatten the matrix B_4 into the lights configuration column vector $\boldsymbol{b_4}$, by stacking the rows.

$$\boldsymbol{b}_4 = (0,1,0,0,0,0,1,0,1,1,1,1,0,0,1,1)^T \in (\mathbb{Z}_2)^{16}. \tag{3}$$

For the given lights configuration $\boldsymbol{b}_4$, to find a winning strategy $\boldsymbol{x}$ that turns out all the lights in $\boldsymbol{b}_4$, we need to solve the linear system modulo two:

$$A_4\boldsymbol{x} = \boldsymbol{b}_4 \mod 2. \tag{4}$$

To solve this linear system modulo two, we first need to construct the toggle matrix A_4 over $\mathbb{Z}_2$, which in SageMath is implemented as $\mathbb{GF}(2)$, the finite Galois field of size 2. The three lines of SageMath code bellow construct the matrix A4, which implements the toggle matrix A_4 over $\mathbb{GF}(2)$:

```
C4 = matrix(GF(2),[[1,1,0,0],[1,1,1,0],[0,1,1,1],[0,0,1,1]])
I4 = matrix.identity(GF(2),4)
A4 = block_matrix([[C4,I4,0,0],[I4,C4,I4,0],[0,I4,C4,I4],[0,0,I4,C4]])
```

In the code above, `C4` implements the matrix C_4 and `I4` implements the matrix I_4. Once we have A_4 constructed in SageMath, we can easily compute its determinant using the command `A4.det()`, which returns zero. Thus, the toggle matrix A_4 has a null space. This implies that not every lights configuration is solvable (winnable), and if it is solvable then it is not unique. More precisely, a configuration $\boldsymbol{b}$ is solvable (winnable) if and only if the vector $\boldsymbol{b}$ belongs to the column space (the span of its column vectors) of the toggle matrix A_4, which we denote by $\mathrm{Col}(A_4)$. The dimension of the column space, also called the rank, can be easily computed in SageMath using the code `A4.rank()`, which returns 12. The rank-nullity theorem states that the rank plus the nullity (the dimension of the null space) is the number of columns, 16 in this case. Therefore, $\dim(\mathrm{Null}(A_4)) = 4$, which you can confirm in SageMath using the command `A4.nullity()`.

We can check directly if the given lights configuration $\boldsymbol{b}_4$ is winnable, that is, whether it belongs to the column space of the toggle matrix A_4. First, we construct in SageMath the column space of A_4, using the command `Col=A4.column_space()`. Then, we define in SageMath the lights configuration vector `b`, which represents $\boldsymbol{b}_4$, as a vector over `GF(2)` and construct the subspace `V` spanned by `b` over `GF(2)`, using the command `V=span([b],GF(2))`. Finally, we can check whether `b` belongs to the column space by using the command `V.is_subspace(Col)`. Below is the SageMath code, which returns `True`, so `b` (i.e. $\boldsymbol{b}_4$) is indeed in $\mathrm{Col}(A_4)$.

```
Col=A4.column_space()
b=vector(GF(2),[0,1,0,0,0,0,1,0,1,1,1,1,0,0,1,1])
V=span([b],GF(2))
V.is_subspace(Col)
```

Even though the toggle matrix A_4 is not invertible, we can still solve the linear system $A_4\boldsymbol{x} = \boldsymbol{b}_4$, using the SageMath command `A4.solve_right(b)` and get a particular solution vector $\boldsymbol{x}$, since $\boldsymbol{b}_4$ is solvable. Alternatively, we can compute the Gauss-Jordan echelon form of A_4, using SageMath, but it is more involved.

Since the initial lights configuration is solvable (winnable), we can find one particular solution `x`, using the SageMath command `x = A4.solve_right(b)`, which returns the vector:

$$\boldsymbol{x} = (1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0). \tag{5}$$

It is easier to see which buttons to push on the 4×4 board if we reshape the solution strategy vector $\boldsymbol{x}$ into a 4×4 matrix, row-wise. We can implement this in SageMath using the command `matrix(4,4,x)`:

```
In: matrix(4,4,x)
Out: [1 0 0 1]
     [1 0 1 1]
     [0 0 1 1]
     [0 0 0 0]
```

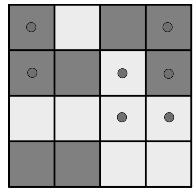

Figure 1: A solution strategy, represented by the gray dots, with 7 button pushes to turn all lights off.

In Figure 1, we show the initial lights configuration $\boldsymbol{b}_4$, together with the buttons to be pushed (the gray dots) so that all lights get turned off, according to the solution strategy $\boldsymbol{x}$.

We leave it to the reader to complete the investigation by exploring the questions below:

- Are there other winning strategies with less than 7 button pushes?
- Is there an optimal strategy that turns all the lights out with the minimum number of button pushes?

References

[1] T.C. Davis, A. De Lamere, G. Sopena, R.C. Soto, S. Vyas, and M. Wong, *peg solitaire in 3 colors on graphs*, under review.

[2] Martin Kreh, *"Lights Out" and Variants*, The American Mathematical Monthly **124** (2017), no. 10, 937-950.