Solving Color-Switching Puzzle on an Infinite Grid

Braedon Besteman

Abstract

In this paper, we embark in the study of the puzzle Parity Lights, studying its properties. The puzzle is a simple one. It is played on a grid with variable dimensions. Traditionally, each cell is either on or off. The puzzle starts with each cell in a random state. The player is allowed one move, called a ”tap” at a target cell: by swapping the state in each cell defined by the *von Neumann neighbourhood* at $r = 1$ around the target cell. The goal of the puzzle is to put the state of each cell in the grid to off. A modified version of this puzzle will be applied to this such that an on cell is one and an off cell is zero. We try to generalize the puzzle such that many modulos can be applied allowing for n-state Parity Lights puzzles. The transition function mentioned is a representation of ”tapping” each lit cell on an empty grid.

Keywords: Convolution, grids, functional composition, linearity, puzzles, parity, parity lights, color-switching graphs, puzzle, abstract algorithms, cellular automata

1 Motivation

The theorem should be one which, even if stated originally [...] in a quite special form, is a capable of considerable extension and is typical of a whole class of theorems of its kind.
∼ G.H. Hardy, A Mathematician's Apology [5]

This paper was originally created in response to a problem that existed when programming a specific game. It is difficult to find how this concept had arisen to mind, as this writing is five years after the event. The problem that existed was the need to be able to determine if the puzzle that was generated was actually solvable for any starting state of the puzzle. After performing more study (by observing the patterns made by the game), fascination arose from the patterns that became of this. Awareness increased as time passed of the certain generality that these rules must have, and it became important for this question of tractability to be determined. We hope to speak generally within this paper so that the ideas may be significant to apply to many other concepts. The understanding and communication of the concepts that are relevant is still a significant part of this endeavor, so we will not withhold from specific example. We would like to display some important and beautiful fractals that we have been able to explore in the process of solving this problem that we hope will convince you to continue to read if we have failed at that task so far.

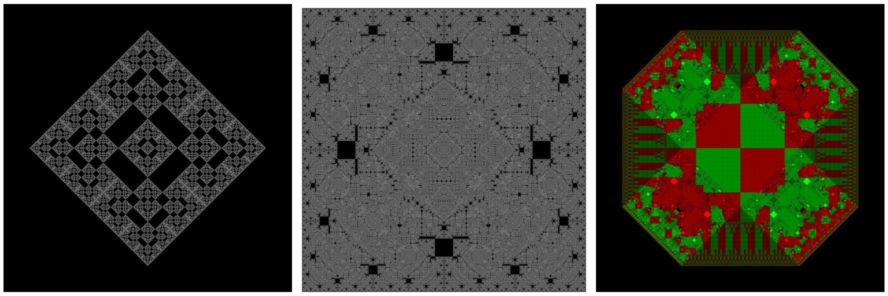

Figure 1: These are three fractals at a power of two from a variety of neighborhoods and starting configurations.

2 Introduction

Throughout this paper, it will be considered our principal purpose to determine an algorithm that will allow us to solve a grid of any size and state or determine that it is not solvable. Our secondary purpose, although directly irrelevant, will be to understand and appreciate the entirety of the amazing patterns that form as a result of our approaches to this problem. *Math is art.* Therefore, we ask the reader to approach this paper as an artist, as a mathematician, and as a pursuer of an understanding of *Nature*. Alexander Giffen, of the University of Dayton, and Darren B. Parker, of Grand Valley State University, propose a generalized version Lights Out, a handheld game by Tiger Electronics©, which would include *Parity Lights* as a subset of this new description [4]. In order to understand how these two systems interact with each other, it becomes imperative that one translates the concept of varying terms between each of these systems of description. Throughout their paper, they refer to the graph of interactions of their game as G. We will adopt this notation with a caveat, that G is a set of graphs, and that $G = \{G_{m,n} \mid m, n\}$. They use a notation of a set of colors for the states that each vertex can be in, $\mathcal{C}$, and I will say that $\mathcal{C} = \mathbb{Z}_2$. The permutation function will be: $T(a) = a+1$. If the grid is solvable, then it is true that there exists a parity dominating set on the grid [4]. I hope to explain my reasoning and mathematical processes within this paper in their entirety with completeness in order for the reader to grasp an intuitive understanding regarding the problem.

3 Preliminaries

3.1 Basics

It becomes vitally important to understand the construction of this game in the realm of mathematics, in order to derive a form of algorithm to solve this game for any grid with dimensions of m, n. For our convenience, we shall define a set of all grids with these dimensions to be $G_{m,n}$. More formally, this can be expressed as: we let $G_{m,n} = \mathbb{Z}_2^{m\times n}$ denote the set of $m \times n$ grids over $\mathbb{Z}_2 = \{0, 1\}$.

Definition 1. *Let $G_{m,n}$ be the set of $m \times n$ grids such that $G_{m,n} = \mathbb{Z}_2^{m\times n}$ and $\mathbb{Z}_2 = \{0, 1\}$*

For our convenience, we shall also define a neighborhood of direct influence for every point. These are the cells that will invert parity upon the cell at (i, j) being tapped. Colloquially, we shall state that this is the von Neumann neighbourhood such that $n = 1$, but we shall also define it more formally to be:

Definition 2. *For every $(i, j) \in \mathbb{Z}_m \times \mathbb{Z}_n$, define the von Neumann neighbourhood (neighbourhood for short) $N_{i,j}$ as*

$$N_{i,j} = \{(i, j), ((i+1) \bmod m, j), ((i-1) \bmod m, j), (i, (j+1) \bmod n), (i, (j-1) \bmod n)\}.$$

For the sake of clarity, we shall also include an image of a cell here, with its neighborhood clearly displayed.

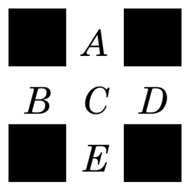

Figure 2: *Cell C* and its neighborhood $\{A, B, C, D, E\}$

In Figure 2, we see on the left side a cell that is being referenced, and on the right side, we see the cells that are in its neighborhood. With these two constructs, we have all that we need to state the puzzle, however, we shall need to define more later in order to fully solve this and complete it.

3.2 Supergrid Function

We shall define a very useful function for our purposes, the Supergrid function. This function will output a grid, where each cell is the parity of the sum of the cells in its neighborhood. We shall refer to it as $f\colon G_{m,n} \to G_{m,n}$, and will formally define it as the following:

Definition 3. *The transition function $f\colon G_{m,n} \to G_{m,n}$ by*

$$f(A)_{i,j} = \sum_{(k,\ell) \in N_{i,j}} A_{k,\ell},$$

where $A \in G_{m,n}$.

The usefulness of this statement may not be inherently obvious, but we will hopefully have convinced you of its profound purpose soon. We ask the reader to consider the act of solving a grid.

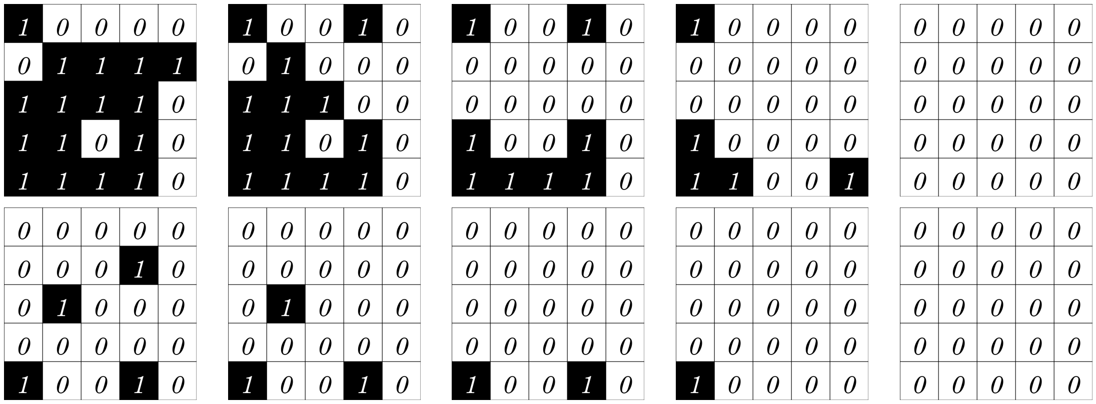

Figure 3: The top grids are the grids shown, which are the supergrids of the other grids

It is important to notice an interesting effect. As cells switch their parity, it operates equivalently to tapping its associated supergrid at that point. As this property exists, one can reasonably assert that by knowing the subgrid of any grid will provide all of the information necessary to solve the grid. This is because one could simply flip all of the cells on the subgrid to *0*, and the supergrid of *0* is 0[i]. Thererfore, to find the subgrid of a grid is equivalent to being able to solve it! Through this realization it becomes crucial to realize the patterns that such a process exhibits. For the security of information within the readers mind, the following, Figure 4, is a redundant, but is nonetheless important, as it serves to provide a complete understanding of the supergrid process.

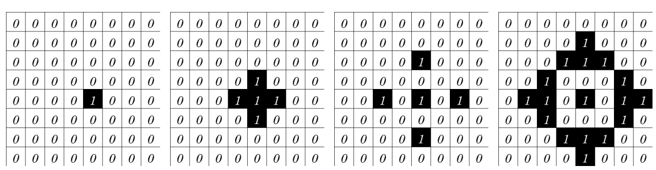

Figure 4: $A_n \in G_{8,8}$ such that $A_{n+1} = f(A_n)$.

3.3 Types of Sets of Grids

Since the object of our desire has now become to discern some grid B from a given grid A such that $f(B) = A$, it has become desirable to travel backwards along f. As one may quickly notice $\forall m, n \in \mathbb{N} : G_{m,n}$ is finite. Therefore, it is impossible to have $A_0 \in G_{m,n} : A_{q+1} = f(A_q)$ and also have the sequence $(A_0, A_1, ...)$ not repeat[ii] Look closely at Figure 4, the supergrid of the final iteration is the beginning iteration. It repeats. This immediately becomes useful information, because for any grid in that sequence, one can simply look at the grid before it as the subgrid. It is important to notice the interesting utility that knowing

[i]It is important to note that 0 may refer to a scalar or, because of convenience, it may also refer to any grid that is composed entirely of 0. That is to say: $\forall A \in G_{m,n} : (\nexists p, q : A_{p,q} \neq 0 \iff A = 0)$

[ii]Suppose that one had a sequence of $A = (A_0, A_1, ..., A_{mn})$. Since $|G_{m,n}| = mn$, it is only possible to select mn unique grids from $G_{m,n}$. As the number of grids in A exceeds this count, it must contain a duplication grid.

what grid, this will be known as the *dot grid*[iii], produces a single dot, which will be known as the *standard basis matrix* or $1^{8,8}_{4,4}$[iv], as it allows us to determine any other subgrid for any grid in $G_{8,8}$. It is true that the supergrid process is linear[v].

3.3.1 Solvable Sets

A solvable set of grids will be a set of grids for which all of the grids within it are solvable. We shall proceed to prove that $G_{8,8}$ is a solvable grid, in order to further understand the process of solving these types of grid. Suppose that $A \in G_{8,8}$, it is true that there exists $Q = \left\{1^{8,8}_{p,q} \mid A_{p,q} = 1\right\}$. It is also true that $\sum Q = A$[vi]. Now, we consider the inverse of f^{-1} to be applied to both sides of this formulation.

$$f^{-1}(\sum Q) = f^{-1}(A)$$

Since, in the case of $G_{8,8}$, we can assert that $\exists h \in \mathbb{N}, f^h = f^{-1}$. Since, $\forall h \in \mathbb{Z}$ we can assert that f^h is linear, f^{-1} is linear. Therefore, the previous equation can be rewritten as:

$$\sum_{A_{p,q}=1} f^{-1}(1^{8,8}_{p,q}) = f^{-1}(A)$$

Since we know that $f^{-1}(1^{8,8}_{p,q})$ to be $D^{8,8}_{p,q}$,

$$\sum_{A_{p,q}=1} D^{8,8}_{p,q} = f^{-1}(A)$$

Since, we know that $\exists D^{8,8}$, we can know that $\exists f^{-1}(A) \forall A$. Therefore, $G_{8,8}$ is a solvable set. We will note that we have directly tied the existence of $D^{m,n}$ to the solvability, denoted as the predicate $\mathcal{S}(G)$, of $G_{m,n}$. In particular, $\mathcal{S}(G_{m,n}) \iff \exists D^{m,n}$.

3.3.2 Null Grids and Unsolvable Sets

There are $G_{m,n}$ such that $\nexists D^{m,n}$, which makes them harder to solve, as one can not simply convolute a grid by another to find its solution. However, it is not impossible, yet it is still important to be able to identify these more difficult sets, so this section will be dedicated to the identification of these sets. $\mathcal{S}(G_{m,n}) \iff \exists D^{m,n} \iff (f \texttt{ is a bijection from } G_{m,n} \texttt{ to } G_{m,n})$. We would like to define another type of grid, that will be denoted as a null grid. A null grid is some grid $N \in G_{m,n}$ such that $N \neq 0$ and $f(N) = 0$. We hesitate to define a notation for null grids as to not overwhelm the reader with notation [vii].

[iii] Some side comment on notation, $D^{m,n}_{p,q}$ is equivalent to some grid $A : A \in G_{m,n}$ and $f(D^{m,n}_{p,q}) = 1^{m,n}_{p,q}$

[iv] This notation is again for our convenience. Generally, $1^{m,n}_{p,q} = A : A \in G_{m,n} \wedge (\forall \mu, \nu \in \mathbb{Z} : (A_{\mu,\nu} = 1 \iff (\mu, \nu) = (p, q)))$

[v] Proof in the Appendix A.1

[vi] The addition of two grids Q, V will be defined as $(Q + V)_{i,j} = Q_{i,j} + V_{i,j}$

[vii] We will maintain to suggest notation for a future paper to adopt. Let $N^{m,n} \subset G_{m,n}$ such that $\forall N \in N^{m,n}$, $N \neq 0$ and $f(N) = 0$. It is also the case that $\nexists Q \in G_{m,n}$, such that the following are all true: $Q \neq 0$, $f(Q) = 0$, and $Q \notin N^{m,n}$.

Theorem 1 (Null Unsolvability)**.** *The following two statements are equivalent: there exists a null grid in $G_{m,n}$ and f on $G_{m,n}$ to $G_{m,n}$ is not a bijection.* [viii]

$$\mathcal{S}(G_{m,n}) \iff \nexists N$$

This becomes a powerful tool of operation, as it allows for the rapid disqualification of sets of grids with certain properties to be solvable. An example of a null grid is the following:

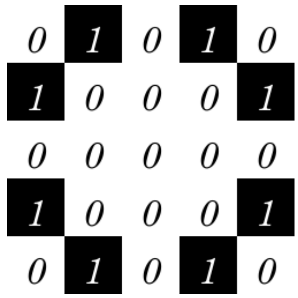

Figure 5: This is a null grid $\in G_{5,5}$.

One of the most truly wonderful aspects of null grids is their ability to be tiled, and still be valid null grids.

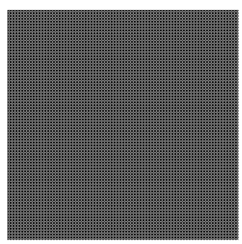

Figure 6: This is a tiled null grid from $G_{5,5}$. It is presumed that this pattern is continued *ad infinitum.*

A spectacular effect of this ability to be tiled[ix] is that it generates null grids in all sets of grids, which have a dimensions that are integer multiples of a null grid with integer multiples. This creates two new important theorems:

Theorem 2. *If there exists a null grid in $G_{m,n}$, then there must also exist a null grid in, for all p, q in the set of all natural numbers, $G_{mq,np}$.* [x]

$$\neg\mathcal{S}(G_{m,n}) \to (\forall p, q \in \mathbb{N} : \neg\mathcal{S}(G_{pm,qn}))$$

It quickly follows that its contrapositive is true:

Theorem 3. *If there exists a dot grid in $G_{m,n}$, then there must exist a dot grid in $G_{p,q}$ such that p is a factor of m and q is a factor of n.*

$$\exists D^{m,n} \to (\forall p, q \in \mathbb{N} : (m \in \mathbb{N}p \land n \in \mathbb{N}q) \iff \exists D^{\frac{m}{p},\frac{n}{q}})$$

[viii] Proof in the Appendix A.2 under Theorem 4
[ix] See appendix for other tiled null grids in Appendix A.2.1
[x] Proof in the Appendix A.2.1 and Proof A.2.1

3.4 Grid Operations

3.4.1 Grid Relations

Grid Equivalence Two grids A and B are said to be equivalent if for all cells in each grid, each cell is equivalent. To be more formal:

Definition 4. *If two grids are equivalent, it would mean that they are the same size. Besides that,* $\forall A, B \in G_{m,n},\ A = B \iff (\forall i, j : A_{i,j} = B_{i,j})$

Grid Congruence Two grids A and B are said to be congruent iff for there exists a translation such that they are equivalent.

Definition 5. *Two grids,* $A, B \in G_{m,n}$*, are congruent if and only if there exists a rigid transformation* $g : G_{m,n} \to G_{m,n}$*, such that* $A = g(B)$*.*

3.4.2 Unary Operations

Rigid Transformation The set of all rigid transformation function contains reflections, rotation, and translations. First, we shall define elementary rigid transformations, T. The function of $T_{i,j}$ will be defined $\forall A \in G$ as $\forall x, y : (T_{i,j}(A))_{x,y} = A_{i+x,j+y}$. $\forall x, y :$ the function $R_{\frac{\pi}{2}}(A)_{x,y} = R_{90^\circ}(A)_{x,y} = A_{-y,x}$, the function $R_{\pi}(A)_{x,y} = R_{180^\circ}(A)_{x,y} = A_{-x,-y}$, the function $R_{\frac{3\pi}{2}}(A)_{x,y} = R_{270^\circ}(A)_{x,y} = A_{y,-x}$. $R_x(A)_{x,y} = A_{x,-y}$ and $R_y(A)_{x,y} = A_{-x,y}$.

$$T = \{i, j : T_{i,j}\} \cup \{R_{90^\circ}, R_{180^\circ}, R_{270^\circ}\} \cup \{R_x, R_y\}$$

The set of all rigid transformations is T^*.
This may seem like too much writing for an intuitive topic, but we feel as though a formal definition would be effective.

Access Operation The value of the cell (i, j) in the grid $A \in G_{m,n}$, can be described as $A_{i,j}$. $\forall i, j : A_{\bmod (i,m),\ \bmod (j,n)} = A_{i,j}$.

3.4.3 Binary Operations

Addition and Multiplication These are three trivial operations, as they operate element-wise.

Definition 6 (Addition)**.** *The sum of two grids,* $A, B \in G_{m,n}$*, can be described as:* $(A + B)_{i,j} = A_{i,j} + B_{i,j}$

Definition 7 (Multiplication)**.** *The product of a grid,* $A \in G_{m,n}$*, and a scalar,* $a \in \mathbb{R}$*, can be described as:* $(aA)_{i,j} = (A)_{i,j}a$

Definition 8 (Hadamard Product)**.** *The product of two grids,* $A, B \in G_{m,n}$*, can be described as:* $(A \odot B)_{i,j} = A_{i,j}B_{i,j}$

Definition 9 (Dot Product)**.** *The dot product of two grids,* $A, B \in G_{m,n}$*, can be described as:* $A \cdot B = \sum_{i,j} A_{i,j}B_{i,j}$

Definition 10 (Convolution)**.** *The convolution of* A *by* B *can be described as:* $(A * B)_{i,j} = (A \cdot T_{i,j}(B))$

4 Algorithms

4.1 Primary Algorithms

It is important to remember that knowing the dot grid of a set tells you instantly how to solve all elements of that set.

Algorithm 1.

There is a valid assertion that we can make, which is fundamental to the operation of the following algorithm. If and only if $\exists k \in \mathbb{N} : f^k(1^{m,n}) \cong 1^{m,n}$*, then there exists* $D^{m,n}$*, which is equivalent to* $f^{k-1}(1^{m,n})$*.*

$A \leftarrow 1^{m,n}$
$N \leftarrow 0$
while *true* ***do***
    $A \leftarrow f(A)$
    $N \leftarrow N + 1$
    if $A \cong 1^{m,n}$ ***then return*** *Solvable* ▷ *This would mean that the previous state of A is the dot grid, which implies the solvability of the set.*
    end if
    if $N \geq 2^{mn}$ ***then return*** *Unsolvable* ▷ *This would imply that there must be a null grid, as it is impossible to return to the original grid.*
    end if
end while

If one were to denote the previous grid from termination, must be the dot grid.

This is one of the most important algorithm throughout this paper, and we shall be using this algorithm as a basis to speed up. One of the most important innovations for speeding up an algorithm, is the discovery of convolutions. It is possible to quickly calculate f^q, using the factors of n. This all derives from the assertion:

$$f^q(1^{m,n}_{0,0}) * A = f^q(A)$$

This assertion is so powerful as it allows for the rapid creation of $f^q(1^{m,n}_{0,0})$. This, is because $f^p(1^{m,n}_{0,0}) * f^q(1^{m,n}_{0,0}) = f^p(f^q(1^{m,n}_{0,0})) = f^{p+q}(1^{m,n}_{0,0})$. If $p = q$, this allows us to calculate f^{2p} directly from f^p. It is quite clear that this would allow us to rapidly calculate f^{2^h} for any h. This formula can also be used to calculate f^{pq}, as this convolution process can be repeated multiple times. This is important, as we are trying to find $\exists k : f^k(1^{m,n}_{0,0}) = 1^{m,n}_{0,0}$. We shall assert that there exists c, which is the minimum possible value of k. This must mean that k is an integer multiple of c. If we were to create an algorithm, that could provide $f^{w!}(1^{m,n}_{0,0})$, then for an arbitrarily large w, one can be fairly certain that all of the factors of c are in $w!$. This would mean that $f^{w!}(1^{m,n}_{0,0}) = 1^{m,n}_{0,0}$ is likely true if $f^k(1^{m,n}_{0,0}) = 1^{m,n}_{0,0}$. Therefore, we can test $f^{w!}(1^{m,n}_{0,0})$ to evaluate this proposition.

Algorithm 2.

```
A ← 1_{0,0}^{m,n}
N ← 0
while true do
    D ← A
    for k = 0, k < N, k++ do
        A ← A * D
    end for
    if A ≅ 1_{0,0}^{m,n} then return Solvable
    end if
    if N ≥ 2^{mn} − 1 then return Unsolvable
    end if
end while
```

While this algorithm is effective at determining the solvability of a set, it does not effectively determine the actual value of $D^{m,n}$. Although a similar algorithm can be used to determine that from this standpoint.

Algorithm 3.

```
A ← 1_{0,0}^{m,n}
N ← 0
while true do
    A_N ← A
    D ← A
    for k = 0, k < N, k++ do
        A ← A * D
        if A ≅ 1_{0,0}^{m,n} then
            K ← k + 1
            for j = N - 2, j > 0, j-- do
                E ← A_j
                E_D ← E
                for i = 0, i < K, i++ do
                    E ← E * E_D
                end for
                H ← E
                H_D ← H
                M ← 1
                for i = 0, i ≤ j, i++ do
                    if (H * H_D) ≅ 1_{0,0}^{m,n} then
                        M ← M + 1
                        Break
                    end if
                    H ← H * H_D
                    M ← M + 1
                end for
```

$K \leftarrow KM$
if $j = 1$ ***then return*** H
end if
end for
end if
end for
$N \leftarrow N + 1$
$A_N \leftarrow A$
if $N \geq 2^{mn} - 1$ ***then return*** *Unsolvable*
end if
end while

This algorithm operates through propagating backwards to find the relevant factors that are the bare minimum to generate c. The result created by this algorithm also rapidly finds the dot grid for each set of grids.
The following algorithm will require many seemingly arbitrary assertions, and will need to develop new functions. One of the most important functions that we will need to develop is $g : G_{m,n} \to G_{m,n}$, which is equivalent to tapping all of the cells on the grid.

$$g(A) = f(A) + A$$

$$f(A) = g(A) + A$$

This algorithm also depends on the existence of $C_0 \in G_{m,n}$ and $C_1 \in G_{m,n}$. These grids must have the following properties: $(C_0)_{i,j} + (C_1)_{i,j} = 1$, $\forall B \in G_{m,n} \to \exists D \in G_{m,n} : g(B \odot C_0) = C_1 \odot D$, and $\forall B \in G_{m,n} \to \exists D \in G_{m,n} : g(B \odot C_1) = C_0 \odot D$. We can also define some functions to tap the cells of C_0 and C_1:

$$g_0(A) = f(C_0 \odot A) + A$$

$$g_1(A) = f(C_1 \odot A) + A$$

We can now discuss some function $(g_0 \circ g_1)$, which operates on the grid. It ends up being true that:

$$(g_0 \circ g_1)(A) = C_1 \odot (g^2(A) + g(A)) = C_1 \odot (f^2(A) + f(A))$$

An interesting thing happens with repeated compositions of the previous functions, in turn we can define $(g_0 \circ g_1)^2$, which operates on the grid.

$$(g_0 \circ g_1)^2(A) = C_1 \odot (g^4(A) + g^2(A))$$

In fact,

$$(g_0 \circ g_1)^d(A) = C_1 \odot (g^{2d}(A) + g^d(A))$$

Therefore if we can find an d (number of repetitions) such that $g^{2d}(A) = g^d(A)$, that would solve the grid. This is true because $(g_0 \circ g_1)^d(A) = C_1 \odot (g^{2d}(A) + g^d(A)) = C_1 \odot (g^d(A) + g^d(A)) = C_1 \odot (0) = 0$. A fascinating fact is that there must always exist an n for which

$g^{2d}(A) = g^{d}(A)$ is true. [xi] For any grid that has both even sides $G_{2m,2n}$, there are some very clear values for C_0 and C_1 that complete these properties. $(C_0)_{i,j} = i + j \mod 2$ and $(C_1)_{i,j} = i + j + 1 \mod 2$. [xii]

Algorithm 4.

$S \leftarrow 0$
$U \leftarrow 0$
while $A \neq 0$ ***do***
  $U \leftarrow U + C_{S \bmod 2} \odot A$
  $A \leftarrow g(C_{S \bmod 2} \odot A) + (C_{S+1 \bmod 2} \odot A)$ ▷ *This is equivalent to tapping all of the cells in $C_{S \bmod 2} \odot A$, which is what one would do when they play the puzzle*
  $S \leftarrow S + 1$
end while
return U ▷ *Returns the calculated subgrid*

There are multiple other approaches that we have tried, and appear productive to the solution of a variety of grids.

4.2 Supplemental Algorithms

The most trivial and unfortunate of these algorithms is the idea that we could traverse every possible subgrid, and test it for validity as the subgrid of the relevant grid. We normally refer to this function as $I_{m,n} : ([0, 2^{mn}) \cap \mathbb{Z}) \to G_{m,n}$. This function is naturally bijective.

Algorithm 5. *To solve a grid A, this function will return the subgrid,*

for *k = 0, k < 2^{mn}, k++* ***do***
  if $f(I_{m,n}(k)) + A = 0$ ***then***
    return $I_{m,n}(k)$
  end if
end for

[xi]This stems from the fact that g is a function that maps a finite set to another finite set. It is inevitable that after more iterations than exist in the domain of the function that some of those iterations must repeat. This would mean that all of the following iterations after the repeating iteration must be equivalent as the function does not depend on any previous. Formally, suppose that $g^a(A) = B$ and that there exists b such that $g^b(A) = B$. We can rewrite this as: $g^a(A) = g^b(A)$. which is the same as $g^a(A) = g^{(b-a)+a}(A)$. We could say that $(b - a) = \Delta a$. Therefore, the following is true $g^a(A) = g^{\Delta a + a}(A)$, which taking $g^{\Delta a}$ of each side provides us, $g^{\Delta a + a}(A) = g^{2\Delta a + a}(A)$. In fact, we achieve $g^a(A) = g^{\Delta a + a}(A) = g^{2\Delta a + a}(A) = g^{3\Delta a + a}(A) = ...$ or $\forall q \in \mathbb{N} : g^a(A) = g^{q\Delta + a}(A)$. If we define $k > a$ and define $\Delta k = k - a$, then we can say that $g^k(A) = g^{\Delta k + a}(A) = g^{\bmod (\Delta k, \Delta a) + a}(A)$. We will now try to calculate the value of k for which $g^k(A) = g^{2k}(A)$. That can be rewritten as $g^{\bmod (k-a, \Delta a)+a}(A) = g^{\bmod (2k-a, \Delta a)+a}$. Therefore, $k - a \equiv 2k - a \pmod{\Delta a} \to k \equiv 2k \pmod{\Delta a} \to k \equiv 0 \pmod{\Delta a} \to k = \Delta a$, which completes our initial assertion.

[xii]For all i, j, $(C_0)_{i,j} + (C_1)_{i,j} = 1$ as $i + j + 1 + i + j \equiv 1 \pmod 2$. $2(i + j) + 1 \equiv 1 \pmod 2 \iff 1 \equiv 1 \pmod 2$.

Unfortunately, this algorithm will likely take a long time to complete, but we considered that it was a worth mentioning. Another approach is to use a logical process similar to Wave Function Collapse Algorithm.

Algorithm 6. *To solve a grid $A \in G_{m,n}$, one must only take the following steps.*

$C \leftarrow$ *an* ***empty grid*** *in* $G_{m,n}$ ▷ *This will become the subgrid of A.*
$B \leftarrow 0$
repeat
    $B \leftarrow 0$
    for $U_{k,\ell} \in C$ ***do***
        $L \leftarrow 0$
        for $(a, b) \in N_{k,\ell}$ ***do***
            $L \leftarrow L + U_{a,b}$
        end for
        if $L = A_{k,\ell} \pmod 2$ ***then***
            $C_{k,\ell} \leftarrow 1 - C_{k,\ell}$
            $B \leftarrow 1$
        end if
    end for
until $B = 0$
return C

5 Future Research

These algorithms provided do not solve all cases, and we recommend finding more algorithms to solve all of the cases. This can also be extended to all for more states, have the grids be in $\mathbb{Z}_q$. One could also experiment with other neighborhoods, and try to generalize this entire area of research. There is a very strange pattern in finding if a set of grids is solvable, and it would be a fascinating as well as certainly difficult problem to try to determine a technique to tell if a set of grids are solvable.

Acknowledgments

This project would not have occurred without Anatoly Zavyalov, a computer science PhD student at Boston University, helping creating the first step, and assisting with the entire understanding as well as notation in creating this paper. He encouraged and sent resources to help, and gratitude is extensive for his support. The project would have never been able to be completed without Olga Korosteleva, a professor at CSULB, who encouraged and believed in it. She instructed its construction, with patience, in the proper method of creating such a document, and for that we are incredibly grateful. We also owe thanks to David Yreuta, a professor at Davenport University, who found encouraged and nurtured mathematical growth, which was so consequential in the understanding of the topics within

this paper. It is at this point, we thank our educators, family, and friends, for all of their assistance in forming the knowledge we have today.
Email Address: braedonbesteman@gmail.com

References

[1] Marlow Anderson and Todd Feil. "Turning Lights Out with Linear Algebra". In: *Mathematics Magazine* 71 (Oct. 1998), p. 300. DOI: 10.2307/2690705.

[2] Ahmet BATAL. "Parity of an odd dominating set". In: *Communications Faculty Of Science University of Ankara Series A1Mathematics and Statistics* 71 (Dec. 2022), pp. 1023–1028. DOI: 10.31801/cfsuasmas.1051208.

[3] Stephanie Edwards et al. "Lights Out on finite graphs Lights Out on finite graphs". In: *involve* 3 (2010), p. 1.

[4] Alexander Giffen and Darren Parker. *On Generalizing the "Lights Out" Game and a Generalization of Parity Domination*. 2009.

[5] G H Hardy. *A Mathematician's Apology*. Cambridge University Press, 2017.

[6] C. David Leach. "Chasing the Lights in Lights Out". In: *Mathematics Magazine* 90 (Apr. 2017), pp. 126–133. DOI: 10.4169/math.mag.90.2.126.

[7] Stephen Wolfram. *A New Kind of Science*. Champaign, Il Wolfram Media, 2019.

Appendices

A The Supergrid Process

A.1 Proof of Linearity

Additivity: For any $A, B \in G_{m,n}$, we have $f(A+B) = f(A) + f(B)$.

Proof. Immediate, as

$$\begin{aligned} f(A+B)_{i,j} &= \sum_{(k,\ell)\in N_{i,j}} (A+B)_{k,\ell} \\ &= \sum_{(k,\ell)\in N_{i,j}} (A_{k,\ell} + B_{k,\ell}) \\ &= \left(\sum_{(k,\ell)\in N_{i,j}} A_{k,\ell} \right) + \left(\sum_{(k,\ell)\in N_{i,j}} B_{k,\ell} \right) \\ &= f(A)_{i,j} + f(B)_{i,j}. \end{aligned}$$

□

Homogenity of degree 1: For any $A \in G_{m,n}$ and c, we have $cf(A) = f(Ac)$,

Proof. Immediate, as

$$\begin{aligned} cf(A)_{i,j} &= c \sum_{(k,\ell)\in N_{i,j}} (A)_{k,\ell} \\ &= \sum_{(k,\ell)\in N_{i,j}} (Ac)_{k,\ell} \\ &= f(Ac)_{i,j} \end{aligned}$$

□

Therefore the function is linear as it is additive and homogeneous of degree 1.

A.2 Null Grids and Solvability

It becomes important to prove that f is bijective. I will provide a theorem:

Theorem 4. *Suppose that there exists $A, B \in G_{m,n}$ with $A \neq B$ and $f(A) = f(B)$. Then $G_{m,n}$ contains a null grid.*

Proof. By the Linear Property of Supergrid Process, we have

$$f(A - B) = f(A) + f(-B) = f(A) - f(B) = 0,$$

so $A - B \neq 0$ is a null grid. □

Since, the implication of Theorem 4, we can state that the property of not being a bijection implies that there exists a null grid. Therefore, by the proof of Theorem 4, it implies the verity of Theorem 1.

A.2.1 Tiled Null Grids

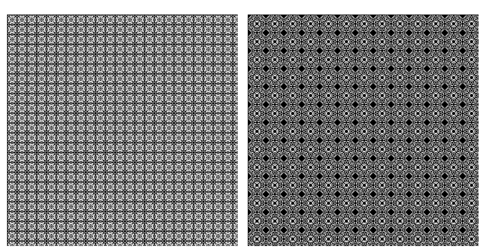

Figure 7: This is a tiled null grid $\in G_{17,17}$, $\in G_{31,31}$.

Proof of Theorem 2

Proof. Immediate, Suppose that there existed a null grid of some size, one could simply duplicate it, and due to the property of accessing beyond a border causing a loop, it must hold true that the next grid is also a null grid. □