

# Reachability, Complexity, and the Limits of Conway's Game of Life

Sebastian Pucher & Adam Gohain

April 27, 2025

## Contents

<b>1</b>	<b>Introduction: Can Life be Simulated?</b>	<b>2</b>
1.1	The Game . . . . .	2
1.2	So... What's the Big Deal? . . . . .	3
<b>2</b>	<b>Definitions, Theorems, and other Important Terminology</b>	<b>3</b>
2.1	What about Complexity? . . . . .	3
<b>3</b>	<b>Reaches-Configuration</b>	<b>3</b>
3.1	Still life Analysis . . . . .	3
3.2	Oscillator Analysis . . . . .	3
3.3	Spaceship Analysis . . . . .	4
<b>4</b>	<b>What can be computed/decided in Conway's Game of Life?</b>	<b>4</b>
4.1	Gliders . . . . .	4
4.2	A Turing Machine in Life . . . . .	6
4.3	Constructing a Universal Constructor . . . . .	6
4.4	Universal Computers . . . . .	6
4.5	Constructing a Universal Turing Machine . . . . .	6
4.6	Some PRIME Examples . . . . .	6
4.6.1	Prime Calculator . . . . .	6
4.6.2	Twin Prime Calculator . . . . .	6
4.6.3	Fermat Prime Calculator . . . . .	6
4.6.4	Mersenne Prime Calculator . . . . .	6
<b>5</b>	<b>REACHES-CONFIGURATION is undecidable</b>	<b>6</b>

# 1 Introduction: Can Life be Simulated?

*That is the question, isn't it.* For centuries, mathematicians, philosophers, artists, and computer scientists, have spent their lives trying to uncover what it means to truly be alive. Many come together, often crossing their respective disciplines to construct answers to these larger, often existential questions. Even today, humanity has continued to develop complicated technology in hopes of understanding more about life, how it can be studied, or even how it could possibly be synthesized.

*What is life?* Back in 1940, John Von Neumann and Stanislaw Ulam set out to prove an answer to this very question. Von Neumann was an American Mathematician whose research focused on self-replicating systems and cellular automata. Alongside his colleague Ulam, who worked together with Von Neumann on the Manhattan Project, they proposed a simple discrete game that replicated life. Their mathematical model consisted of a two-dimensional grid of square cells, where the state of the next generation of cells would depend on the interaction between living cells and their neighbors [Beginning'Life'2006]. They called it *The Universal Constructor* which produced fascinating properties of time and space usage [Freitas'2004].

Not long after their proposal, a British Mathematician known as John Conway extended upon Ulam and Von Neumann's research to fabricate an instance of their Universal Constructor that better replicated Alan Turing's "universal computer". By experimenting with different rules and states between neighboring automata, Conway was able to simplify the model into a game that was only composed of only a few basic rules [Beginning'Life'2006]. [see section (CITE SECTION)].

Shortly after, in 1970, the *Scientific American* published an article articulating how to play the game which resulted in the greatest number of letters reactions from readers at that time [Izhikevich'Conway'Seth]. In the paper, Conway proposed that no initial pattern could grow without limit, and offered fifty dollars to the first person who could disprove him by the end of the year [math-games]. This catalyzed immense popularity in the game, and set forth the many mathematical discoveries that have now been proven about the game, such as its undecidable nature (see section (CITE SECTION)), and recurring patterns(see section).

Going to add a nice image here

## 1.1 The Game

Similar to how Alan Turing proposed models of computational thinking prior to modern day computers, Conway's Game of Life started as a simple mathematical idea that was "played" on chalkboards and Go boards [Izhikevich'Conway'Seth]. Here are the rules, and how to play:

### Rules & Properties:

#### 1. Domain:

The automata in the game interact within an *infinite* two-dimensional grid of cells. Every cell has eight neighbors [Izhikevich'Conway'Seth][figure x].

#### 2. States:

Each automata is represented independently by a single cell which can be either *alive* or *dead*.

#### 3. Initial Configuration:

The beginning set of live or dead cells is determined or "seeded" by the player prior to any evolution.

#### 4. Evolution:

The following rules are applied to all cells simultaneously in fixed time intervals called *generations* [Bontes2019].

#### 5. Birth:

A new cell is born at generation  $t + 1$  when its state is currently *dead* and has exactly three live neighbors (reproduction) [Bontes2019].

#### 6. **Death:**

Any currently living cell will die at generation  $t+1$  if it has less than 2 live neighbors (underpopulation) or more than three live neighbors (overpopulation) [Bontes2019].

#### 7. **Persistence:**

Any live cell will persist at  $t+1$  if it has two or three live neighbors at generation  $t$  [Izhikevich·Conway·Seth].

It's important to note that Conway's Game of Life is not a game of how we traditionally think of games. There's no objective, or winning or losing. They're aren't even any players – it is known to be a zero player game [Beginning·Life·2006]. As technology advanced, Conway's Game of Life proved to be well-suited for implementation on computers. Today, the game has been optimized to explore the many unresolved problems and complexity the game poses.

## 1.2 So... What's the Big Deal?

At first glance, Conway's game appears to be a simple simulation of life based on just a few rules. However, beneath the surface, the Game of Life hides mathematical complexities that have perplexed many since its proposal way back in 1970. Many of its implications stem from the fact that the game is Turing Complete. Entire self-replicating machines, digital circuits, and unique patterns have been discovered and studied within the properties of the game. In fact, new behaviors are still being uncovered today.

**Proposal:** The purpose of this paper is to both synthesize concrete theoretical mathematics from Complexity Theory into a specific instance of Conway's game of life. More specifically, the paper will be divided into two parts. The first pertaining to the Pattern reachability question, and the second pertaining to Undecidability and Turing Completeness in Conway's Game of Life.

## 2 Definitions, Theorems, and other Important Terminology

### GOALS FOR SUB SECT

1. Explain the common patterns / life forms that can be found in the game
2. Explain the three patterns we will be examining (Still life, oscillator, and spaceship)
3. Add images to these
4. Explain other relevant mathematical terminology / information that's worth introducing

### 2.1 What about Complexity?

#### GOALS FOR SUB SECT

1. More of an intro sub section connecting the game to complexity analysis
2. This is where we'll define important complexity classes like NP
3. outline any other important info we want to outline before getting into more gritty / mathy stuff

## 3 Reaches-Configuration

### 3.1 Still life Analysis

#### GOALS FOR SECTION

### 3.2 Oscillator Analysis

#### GOALS FOR SECTION

### 3.3 Spaceship Analysis

#### GOALS FOR SECTION

## 4 What can be computed/decided in Conway's Game of Life?

The quest to understand the limits of computation has been a central theme in computer science since its inception. Alan Turing's groundbreaking work introduced the concept of a theoretical machine, now known as the Turing machine (TM), capable of simulating any algorithm. As we talked about in class, systems that possess this capability are deemed *Turing complete*, signifying they have the maximum theoretical computational power. Discovering that a system, especially one with seemingly simple rules like Conway's Game of Life, is Turing complete reveals a profound depth and complexity. It implies that the system can, in principle, compute anything that any other computer can. This section delves into the remarkable computational capabilities embedded within the evolving patterns of Conway's Game of Life, exploring how complex computational structures, including Turing machines themselves, can emerge from its basic local interactions.

In short, we'll provide an overview of how it's even possible to construct a Universal Turing machine (UTM) in Life, and why that validates it as a Turing complete model of computation.

### 4.1 Gliders

A *glider* is a specific pattern in Conway's Game of Life that exhibits translational movement across the grid while undergoing periodic transformations in its structure. Formally, we define a glider as follows:

**Definition 4.1** (Glider). A glider is a configuration  $G$  of live cells in Conway's Game of Life such that:

1. After exactly  $p$  generations (where  $p = 4$ ), the pattern transforms into a translated version of itself:  $G_{t+p} = T(G_t)$ , where  $T$  represents a translation operation.
2. The translation  $T$  corresponds to a diagonal movement by one cell diagonally (specifically,  $\pm 1$  cell horizontally and  $\pm 1$  cell vertically).
3. The pattern consists of exactly 5 live cells in each of its phase configurations.
4. The pattern cycles through exactly 4 distinct configurations before returning to a translated version of its initial state.

As illustrated in Figure 1, the glider completes a full cycle every 4 generations, moving one cell diagonally in the process. The minimal glider discovered by Conway consists of 5 live cells arranged in an asymmetric pattern that evolves through 4 distinct phases before returning to its original configuration, albeit in a new position.

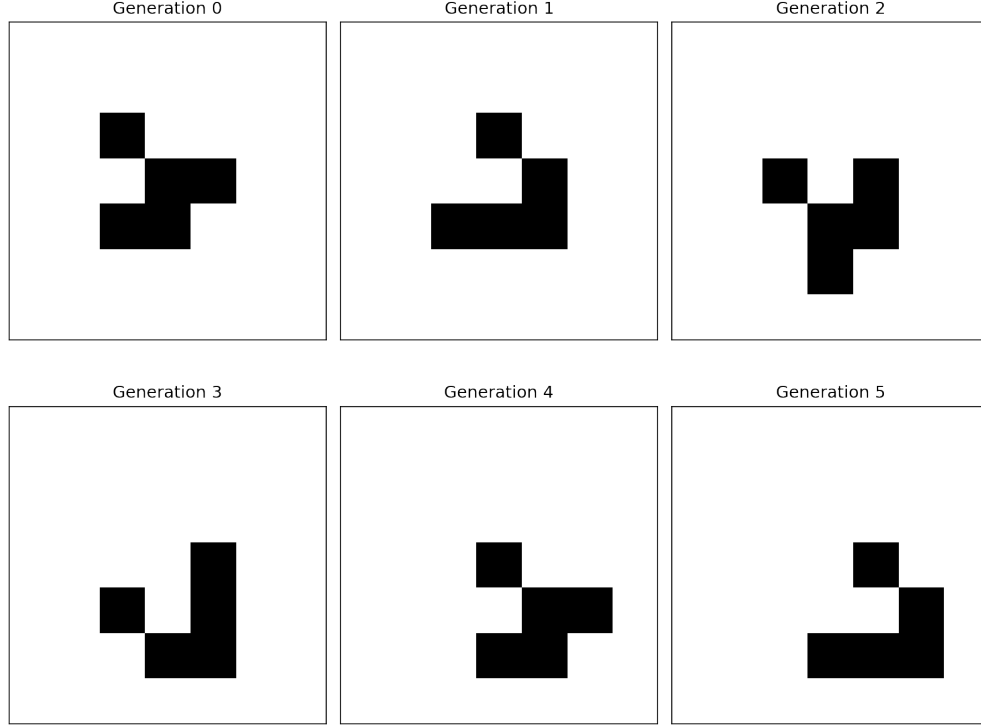


Figure 1: Evolution of a glider pattern through generations 0 to 5, demonstrating the characteristic diagonal movement and shape cycle.

Gliders are particularly significant in Conway's Game of Life because they serve as the fundamental mechanism for information transfer, as we'll see in the next few sections. Since they can travel arbitrarily far across the infinite grid, they effectively function as signals or "data packets" in computational constructions. In order to talk about operations on such "data" later on, we'll also need to define the following:

**Definition 4.2** (Glider Collision). Let  $G_1$  and  $G_2$  be glider patterns with positions  $P_1(t)$  and  $P_2(t)$  at time  $t$ , where  $P_i(t)$  denotes the set of coordinates of live cells comprising glider  $G_i$ . A glider collision occurs at time  $t_c$  if there exists a time  $t_c$  such that:

1. For all  $t < t_c$ , the evolution of  $G_1$  and  $G_2$  proceeds independently, i.e., the state of each cell at time  $t$  depends only on its own glider pattern's evolution.
2. At time  $t_c$ , there exists a neighborhood  $N$  such that both gliders influence the evolution of cells in  $N$ , formally:  $\exists N \subseteq \mathbb{Z}^2$  such that  $d(P_1(t_c), N) \leq 1$  and  $d(P_2(t_c), N) \leq 1$ , where  $d$  denotes minimum Manhattan distance.

The outcome of a collision at time  $t_c + k$  (for some finite  $k > 0$ ) can be classified as one of:

1. *Annihilation*: The resulting pattern  $R(t_c + k)$  contains no persistent structures, i.e.,  $\exists t' > t_c + k$  such that  $\forall t > t', R(t) = \emptyset$ .
2. *Reflection*: The resulting pattern contains exactly two gliders  $G'_1$  and  $G'_2$  with velocity vectors  $v'_1$  and  $v'_2$  such that  $v'_i \neq v_i$  for at least one  $i \in \{1, 2\}$ .
3. *Transformation*: The resulting pattern contains at least one glider with a velocity vector different from both  $v_1$  and  $v_2$ , or the number of resulting gliders differs from the initial number.
4. *Creation*: The resulting pattern contains both one or more gliders and at least one persistent non-glider structure (still life or oscillator).

**Definition 4.3** (Glider Gun). A pattern  $G \subseteq \mathbb{Z}^2$  is a glider gun with period  $p \in \mathbb{Z}^+$  if there exists a finite core region  $C \subseteq \mathbb{Z}^2$  and a sequence of configurations  $(G_t)_{t=0}^\infty$  representing the evolution of  $G$  under Conway's Game of Life rules such that:

1. There exists a non-empty finite set  $C \subseteq G$  such that  $\forall t \geq 0, G_t \cap C = G_{t+p} \cap C$
2. There exists a sequence of times  $(t_n)_{n=1}^\infty$  with  $t_n = np + d$  for some fixed offset  $d \in \{0, 1, \dots, p-1\}$  and a glider pattern  $H$  such that:
  - For each  $n \geq 1$ , there exists a translation vector  $\vec{v}_n = n \cdot \vec{v}$  for some fixed  $\vec{v} \in \mathbb{Z}^2 \setminus \{(0, 0)\}$  such that  $H + \vec{v}_n \subseteq G_{t_n}$
  - For any distinct  $n, m \geq 1$ , the patterns  $H + \vec{v}_n$  and  $H + \vec{v}_m$  are disjoint
3. There exists  $T \in \mathbb{Z}^+$  such that for all  $n \geq 1$  and all  $t \geq t_n + T$ , the evolution of  $H + \vec{v}_n$  proceeds independently of both the core region  $C$  and all other emitted gliders
4. The distance between the core region  $C$  and the  $n$ -th emitted glider  $H + \vec{v}_n$  increases without bound as  $n \rightarrow \infty$

The discovery of the Gosper glider gun in 1970 was pivotal, as it disproved Conway's conjecture that no pattern could grow indefinitely. More importantly, the ability to generate endless streams of gliders provides the foundation for information processing (which we keep omniously alluding to!!!).

## 4.2 A Turing Machine in Life

## 4.3 Constructing a Universal Constructor

## 4.4 Universal Computers

## 4.5 Constructing a Universal Turing Machine

## 4.6 Some PRIME Examples

### 4.6.1 Prime Calculator

### 4.6.2 Twin Prime Calculator

### 4.6.3 Fermat Prime Calculator

### 4.6.4 Mersenne Prime Calculator

## 5 REACHES-CONFIGURATION is undecidable