CAL-BRIDGE

# Advancing the Study of Larger than Life Cellular Automata with Lua Scripting

Brandon Ismalej, Computer Science, CSUN
Cal-Bridge, CSUN Department of Mathematics, Dr. Kellie Evans

Cal-Bridge Research Symposium | September 14, 2024

# Outline

- Introduction
- Motivation
- Related Work
- Design and Methodology
- Experimental Results and Analysis
- Conclusion
- References

CAL-BRIDGE

# Larger than Life Cellular Automata

- Cellular Automata (CA): A class of discrete, grid-based computational models which are based on simple rules and algorithms
  - Cell states: Typically binary, being either "live" (1) or "dead" (0), and can change states based on states of neighboring cells

- Conway's Game of Life (*Life*): A CA with simple rules for cell birth, survival, and death, which can lead to dynamic patterns like "spaceships" [1]

- Larger than Life (LtL): A generalization of Life, that extends to larger neighborhoods and uses intervals for birth and survival thresholds [2]
  - Complexity: Allows for exploration of more intricate patterns, such as "bugs"

CAL-BRIDGE

# How can we explore these CA?

Golly is an open-source software for the exploration and simulation of Conway's Game of Life and other cellular automata, including Larger than Life [3]

- Written in C++
- Supports scripting in Python and Lua
- Supports bounded and unbounded universes

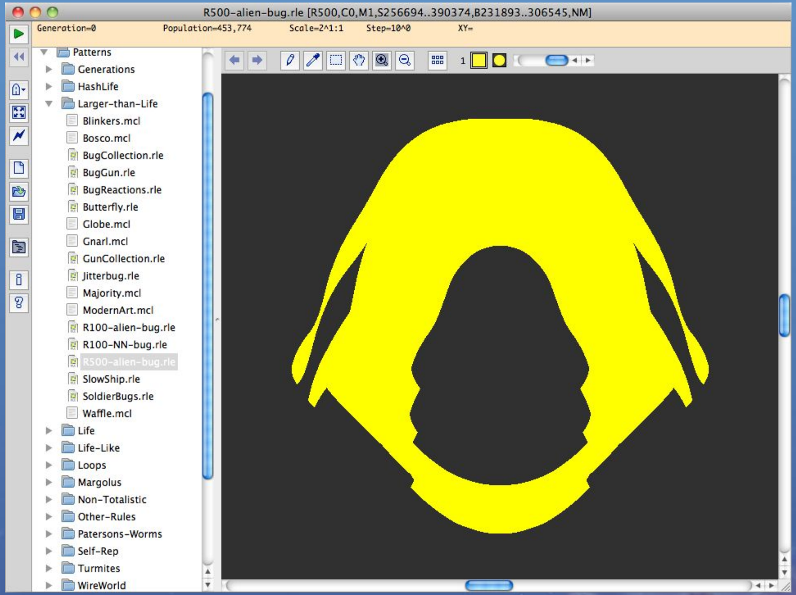



Figure 1: Snapshot of Golly GUI [3]

# Motivation

- In *Life*, the most intriguing patterns are known as "spaceships"
  - Can carry information across space as time updates
  - The most famous spaceship is known as a glider
- In LtL, generalizations of *Life's* spaceships are known as "bugs"
  - Exhibit complex dynamics and behaviors

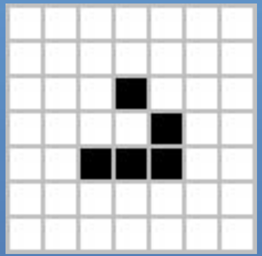


Figure 2: Life's glider [4]

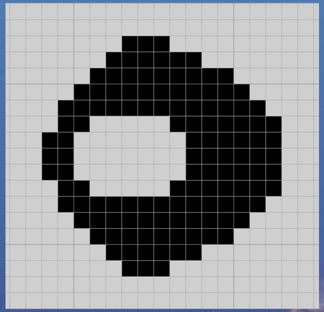


Figure 3: LtL Range 25 Bug

CAL-BRIDGE

# Big Question:

“If cellular automata follow specific rules and algorithms, how can we systematically discover and identify complex patterns like “bugs” within these systems?”

CAL-BRIDGE

# Current Methods for Bug Discovery

- Random Soup Searching
  - "apgsearch" is an automated search program written by Adam P. Goucher [5]
    - Generates large amounts of random asymmetrical soups and runs each soup with a user provided CA rule
- Finite Deterministic Configurations
  - Evans describes the use of geometric initial configurations, such as rectangles and circles, that a rule will "sculpt" into a bug [2] [6]
  - Commonly used and leverages configurations that resemble the geometry of bugs

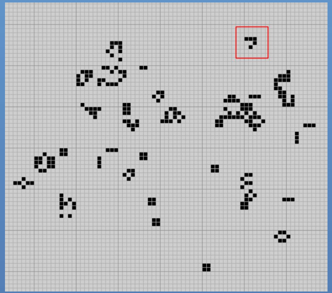


Figure 4: Glider Emerging from Random Soup

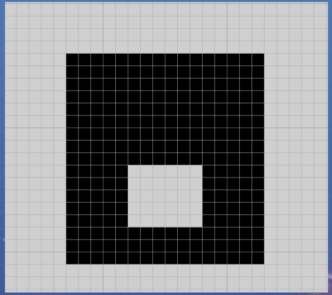
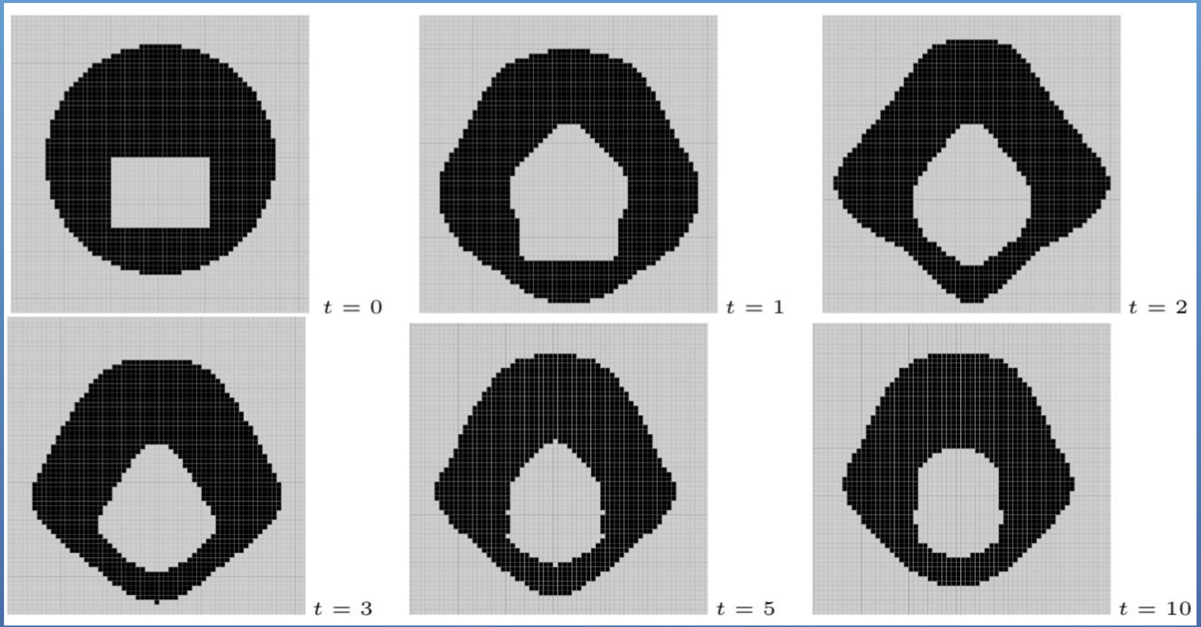


Figure 5: Initial Configuration

CAL-BRIDGE




# Figure 6: Geometric Initial Configuration

An illustration of a geometric initial configuration sculpted into a bug over time = $t$ by the LtL rule with parameters R25,C0,M1,S706..1216,B706..958,NM:

- Dimensions of Initial Configuration:
  - Circle: radius = 24, y – setback = 7, Rectangle: length = 21, width = 15

CAL-BRIDGE

# Scripting Design

A set of Lua scripts has been developed to automate the creation and simulation of geometric initial configurations onto the Golly grid

The most notable script aims to:

- <u>Create and place configurations</u>, based on user-defined parameters, such as:
  - Shape of live and dead sites: circle, rectangle, ellipse
  - Dimensions of shapes: radii, length/width, major/minor axes
  - Vertical "setback" of dead sites: vertical distance from center of live site to center of dead site
- <u>Run a simulation of every configuration</u> created to:
  - Detect surviving patterns, particularly bugs
  - Sort configurations into CSV files, based on their behavior after the simulation has been run, such as patterns that die off

# Classifying Patterns: Wolfram's Framework [7]

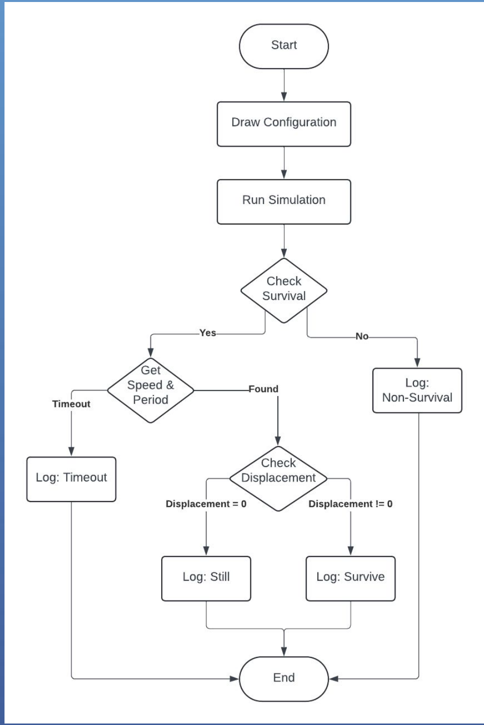



Figure 7: Flowchart of Pattern Classification

- Class 1 (Homogenous States): Dead patterns with no live cells after simulation are classified as non-surviving and logged in "not_survive.csv"
- Class 2 (Periodic Structures): Surviving patterns with no vertical displacement are classified as "still lifes" and logged in "still.csv", while those with displacement are logged in "survive.csv"
- Class 3 (Chaotic Aperiodic Behavior): Patterns that exceed the iteration limit are classified as timeouts and logged in "timeout.csv"
- Class 4 (Complex Localized Structures): Any intricate structures detected during exploration are classified and logged in "survive.csv"

CAL-BRIDGE

# Experimental Results

The following experiment was conducted, based on the user-parameters requested by our Golly-integrated script:

- Name convention for CSV files:
  - Exp1_R25_liveCircle_deadRectangle
- Number of simulation time steps before categorization:
  - 60
- Rule for current grid:
  - R25,C0,M1,S720..1258,B720..978,NM
- Max. iterations in case of runtime error:
  - 1,000
- Shape of live sites:
  - C (circle)
- Bounds of radii
  - 20,25
- Bounds for y-setback
  - 5,15
- Shape of dead sites:
  - R (rectangle)
- Bounds of length/width
  - 17,22 17,22

CAL-BRIDGE

| | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | R25,C0,M1,S720..1258,B720..978,NM | | | | | | | | | |
| 2 | **Dimensions of Live Shape** | **Y Setback** | **Shape of Dead Cells** | **Dimensions of Dead Shape** | **Period** | **dy** | **Population** | **Bound Box Wd** | **Bound Box Ht** | **Hash Value** |
| 3 | 21 | 6 | R | 17 17 | 1 | -3 | 1361 | 49 | 49 | -565013023 |
| 4 | 21 | 7 | R | 17 17 | 1 | -3 | 1361 | 49 | 49 | -565013023 |
| 5 | 21 | 7 | R | 19 17 | 1 | -3 | 1361 | 49 | 49 | -565013023 |
| 6 | 21 | 8 | R | 17 17 | 1 | -3 | 1361 | 49 | 49 | -565013023 |
| 7 | 21 | 8 | R | 17 18 | 1 | -3 | 1361 | 49 | 49 | -565013023 |
| 8 | 21 | 9 | R | 17 17 | 1 | -3 | 1361 | 49 | 49 | -565013023 |
| 9 | 21 | 9 | R | 18 17 | 1 | -3 | 1361 | 49 | 49 | -565013023 |
| 10 | 21 | 9 | R | 19 17 | 1 | -3 | 1361 | 49 | 49 | -565013023 |
| 11 | 21 | 10 | R | 17 18 | 1 | -3 | 1361 | 49 | 49 | -565013023 |
| 12 | 21 | 10 | R | 17 19 | 1 | -3 | 1361 | 49 | 49 | -565013023 |
| 13 | 21 | 10 | R | 18 17 | 36 | -108 | 1378 | 48 | 49 | 1504284469 |
| 14 | 21 | 10 | R | 19 17 | 1 | -3 | 1361 | 49 | 49 | -565013023 |
| 15 | 21 | 10 | R | 20 17 | 36 | -108 | 1378 | 48 | 49 | 1504284469 |
| 16 | 21 | 10 | R | 21 17 | 1 | -3 | 1361 | 49 | 49 | -565013023 |
| 17 | 21 | 11 | R | 17 17 | 1 | -3 | 1361 | 49 | 49 | -565013023 |
| 18 | 21 | 11 | R | 17 20 | 1 | -3 | 1361 | 49 | 49 | -565013023 |
| 19 | 21 | 11 | R | 19 18 | 1 | -3 | 1361 | 49 | 49 | -565013023 |

# Experimental Results

The experiment yielded 2,376 configurations created, simulated, and classified.

- Figure 8 above shows a snapshot of the output of "Exp1_R25_liveCircle_deadRectangle_survive.csv" with 1,163 initial configurations found to sculpt into bugs.

# Analysis

Data-Driven Exploration:

- Automating the creation, simulation, and sorting of geometric configurations produced large datasets
- These datasets allow for comprehensive analysis, supporting the formatting and validation of conjectures regarding LtL patterns

Key Insights:

- Understanding the sensitivity of bugs and other life-like patterns to specific initial configurations
- Identifying common traits in configurations that lead to stable or emergent behaviors

CAL-BRIDGE

# Conclusion

Summary of Key Contributions:

- Introduced an automated method for exploring geometric configurations in Larger than Life cellular automata
- Developed a systematic approach to classify patterns based on their behavior and dynamics
- Generated data to support further conjectures about bug-like patterns and other emergent phenomena and their parameter spaces

Impact and Potential:

- Enables more targeted searches compared to random soup methods, improving pattern recognition
- A foundational approach for the integration of ML for advanced pattern detection

CAL-BRIDGE

E-mail: brandon.ismalej.671@my.csun.edu

# Thank you!

CAL-BRIDGE

# References


- [1] M. Gardner, "Mathematical games - The fantastic combinations of John Conway's new solitaire game 'Life'," Scientific American, Oct. 1970. [Online]. Available: https://www.scientificamerican.com/article/mathematical-games-1970-10/
- [2] K. M. Evans, "Larger than Life: it's so nonlinear," PhD Dissertation, University of Wisconsin-Madison, Madison, WI, 1996.
- [3] T. Rokicki and A. Trevorrow, "Golly: An open source, cross-platform application for exploring Conway's Game of Life and other cellular automata," 2024, version 4.3, accessed on August 11, 2024. [Online]. Available: https://golly.sourceforge.io
- [4] "Glider - LifeWiki." Accessed: Sep. 04, 2024. [Online]. Available: https://conwaylife.com/wiki/Glider
- [5] A. P. Goucher, Mar. 2024. [Online]. Available: https://gitlab.com/apgoucher/apgmera
- [6] K. M. Evans, "Larger than Life: threshold-range scaling of Life's coherent structures," Physica D: Nonlinear Phenomena, vol. 183, no. 1, p. 45–67, Sep. 2003.
- [7] S. Wolfram, Cellular Automata and Complexity: Collected Papers. Boca Raton, FL: CRC Press, 1994. [Online]. Available: https://www.wolframscience.com/reference/


CAL-BRIDGE