Ahmedabad University

TOC Assignment 2

Group Members:

Name	Enrollment no.
Nevil Jobanputra	AU2140209
Aryan Prajapati	AU2140090
Buland Shah	AU2140088
Sujal Virani	AU2140139
Aaditya Yadav	AU2140094
Muskan Chauhan	AU22L20003
Anvi Sheth	AU22L20001

Website Link: https://evolutionofprogram.wixsite.com/toc-project

Elementary Cellular Automata

Model 1: Context-Aware Cellular Automaton

a) Existing Model Considered

- The starting point is the classic elementary cellular automaton:
  - Each cell's state is determined by the state of its left, center, and right neighbors in the previous row.
  - Evolution follows a fixed global rule (e.g., Rule 30, Rule 110).

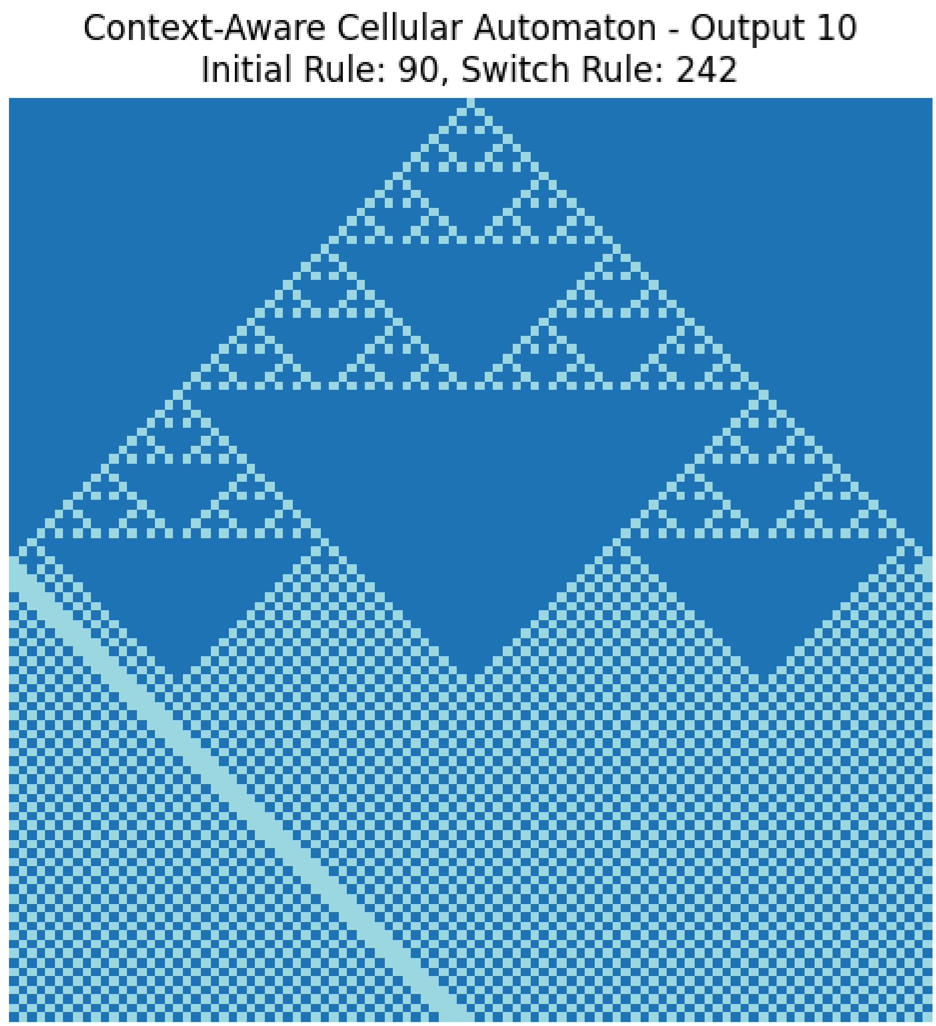
b) The Way Uniformity Is Broken

- Rule Switching:
  - The automaton dynamically switches between two rules based on detected patterns.
  - If clusters of active cells (consecutive 1s) exceed a threshold (e.g., more than 2), the system switches from the initial rule to the switch rule.
- Global Context Awareness:
  - The decision to switch rules depends on analyzing the entire row, breaking the uniform application of a single rule.

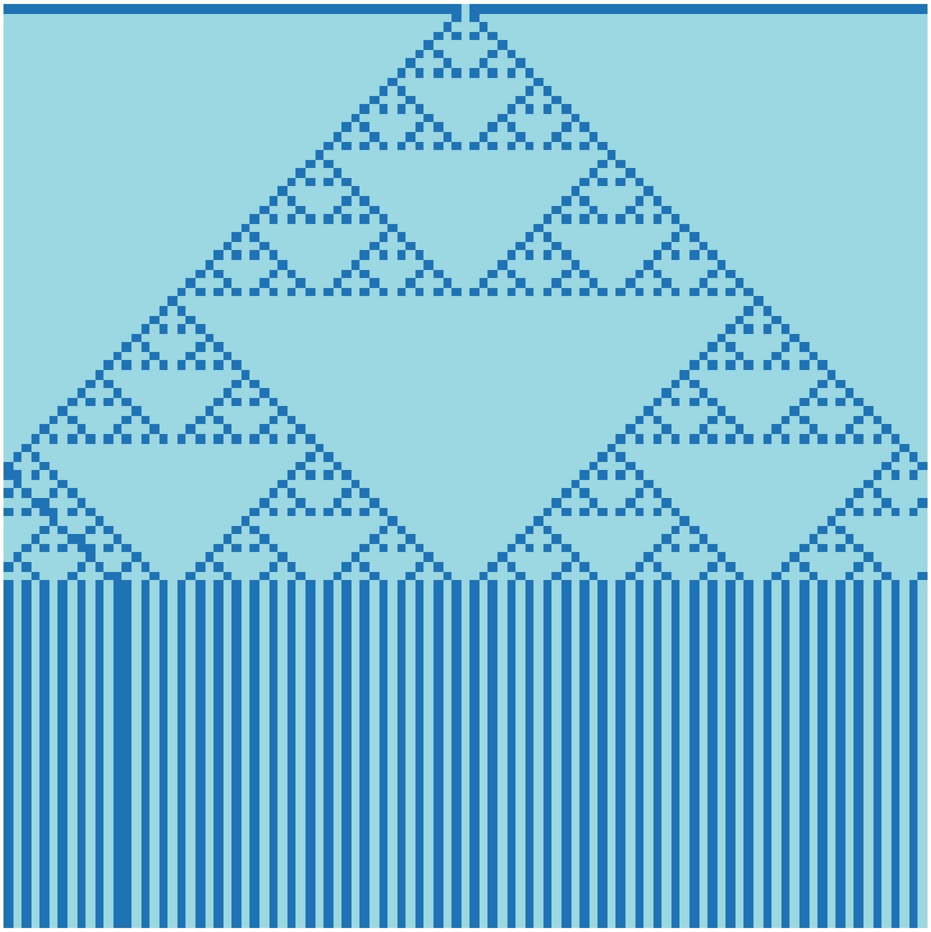
c) Details of the Proposed Model

- Two rules are defined:
  - Initial Rule: Governs evolution under normal circumstances.
  - Switch Rule: Activated when certain patterns (e.g., clusters) are detected.
- Each cell updates synchronously according to the current rule.
- Pattern Detection Logic:
  - The automaton identifies clusters by counting consecutive 1s in the grid.
  - Rule switching occurs dynamically based on these patterns, breaking the uniformity of evolution.

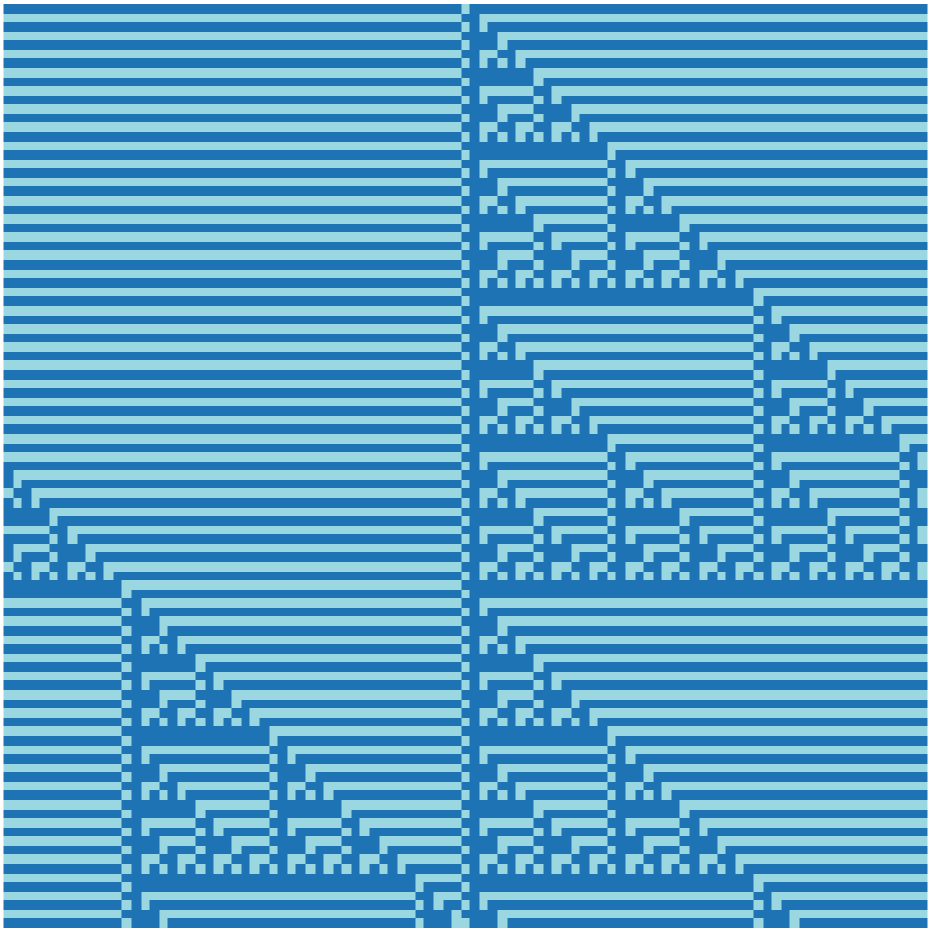
d) Best Possible Simulation Patterns



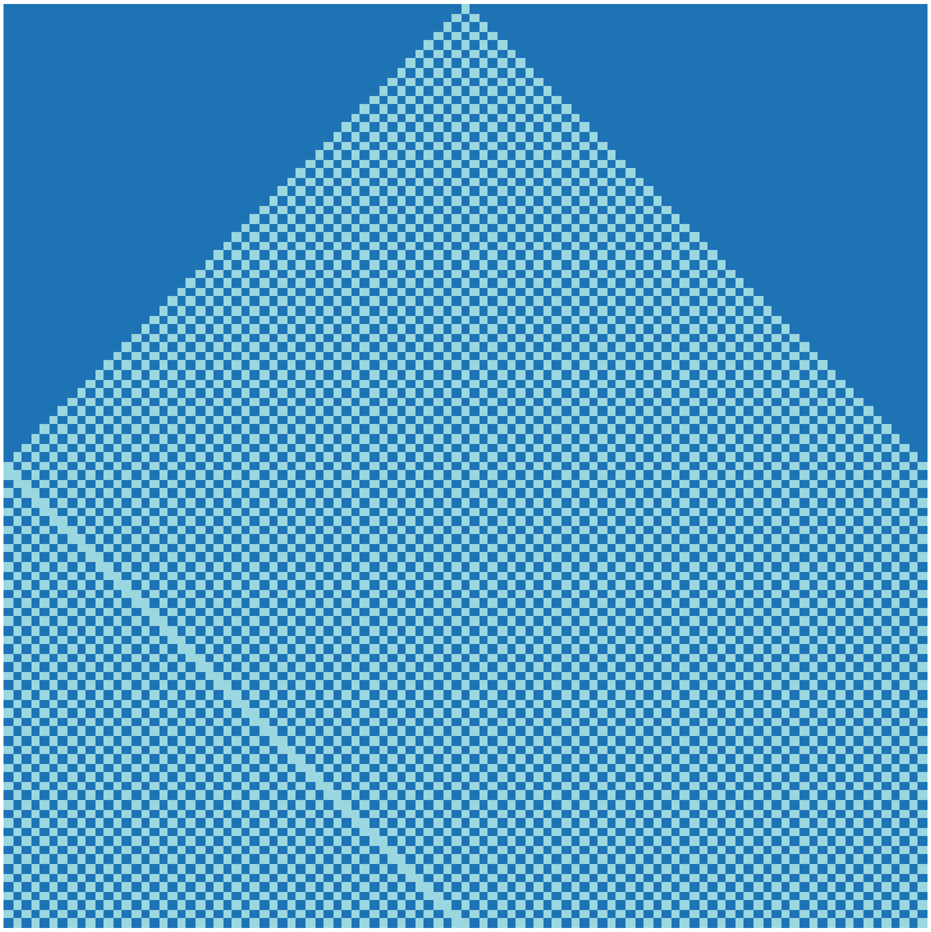
Context-Aware Cellular Automaton - Output 9
Initial Rule: 13, Switch Rule: 180



Context-Aware Cellular Automaton - Output 6
Initial Rule: 195, Switch Rule: 28



Context-Aware Cellular Automaton - Output 2
Initial Rule: 242, Switch Rule: 219



Model 2: Spatially Varying Update Frequency Cellular Automaton

a) Existing Model Considered

- The base is again the elementary cellular automaton:
  - Uniform update timing: All cells update their states simultaneously.
  - Fixed global rule governs evolution.

b) The Way Uniformity Is Broken

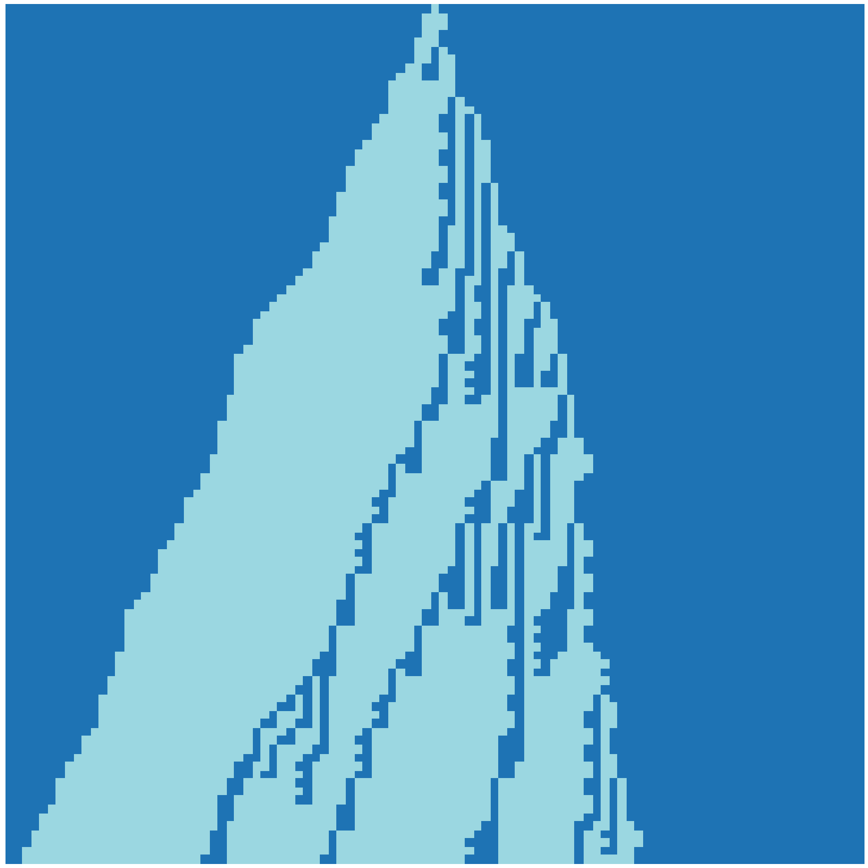
- Spatial Variation in Update Frequency:
  - Each cell is assigned a random update frequency (1 to 5 steps).
  - Not all cells update their states at the same time.
- Asynchronous Evolution:
  - Cells evolve according to the same rule but at different time intervals, creating localized variations.

c) Details of the Proposed Model

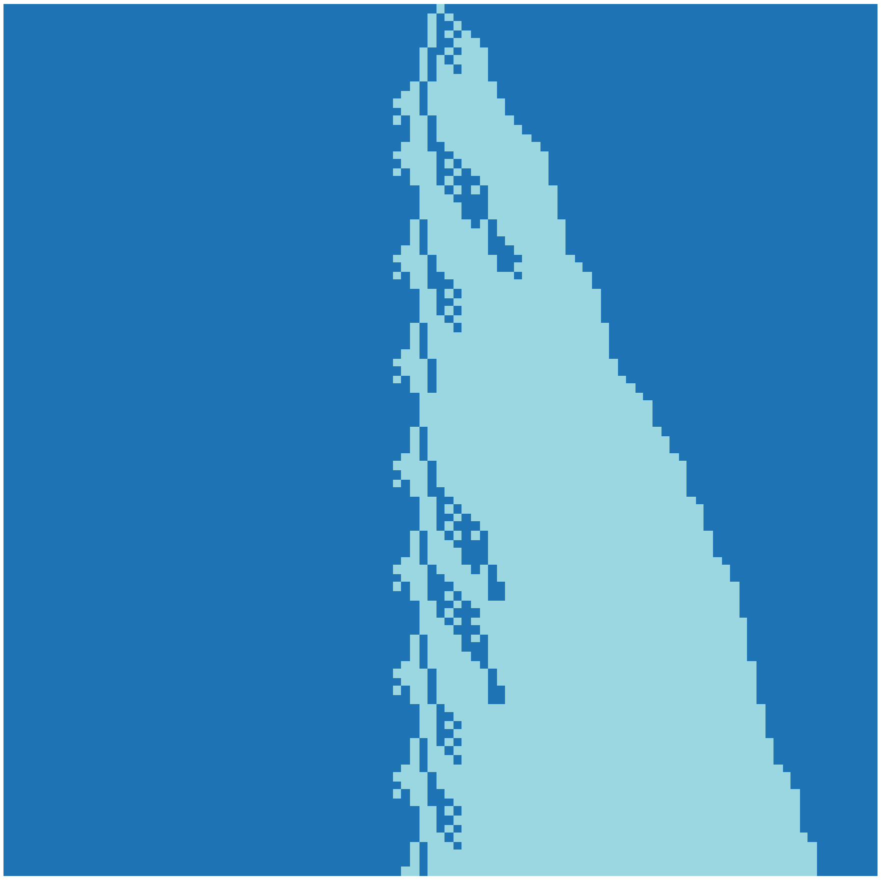
- A single rule governs all cells' state transitions.
- Each cell is assigned a random frequency dictating how often it updates:
  - Cells with a frequency of 1 update at every step.
  - Cells with a frequency of 3 updates once every 3 steps.
- The randomness of update intervals introduces spatial and temporal heterogeneity.

d) Best Possible Simulation Patterns

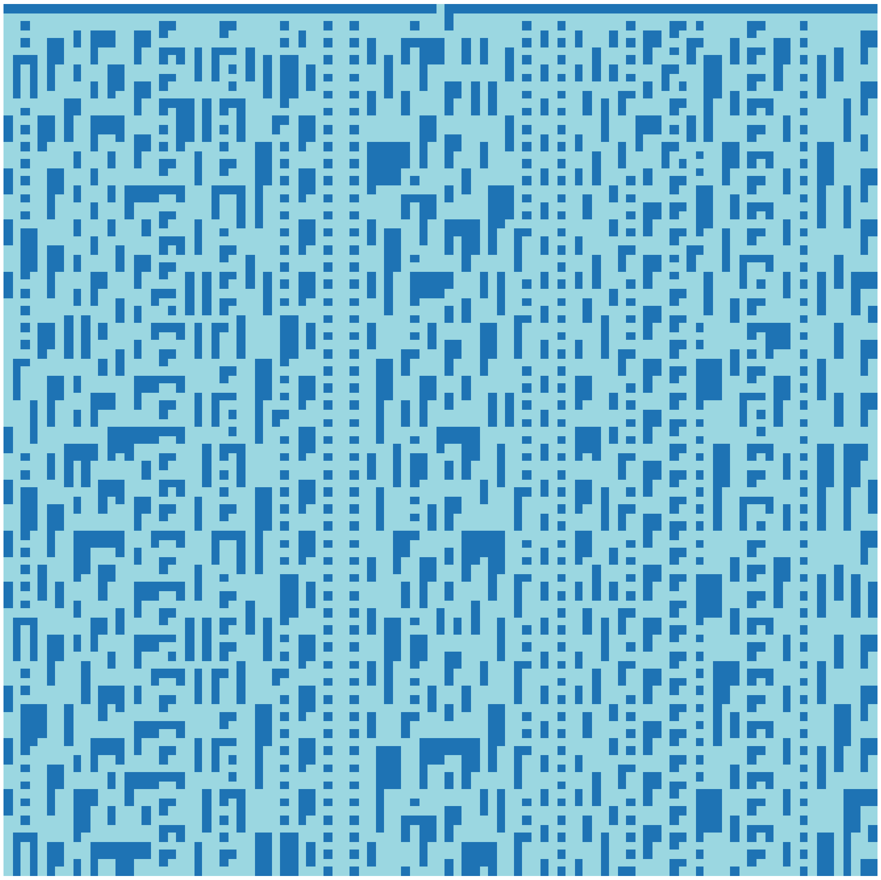
Rule 158 - Spatially Varying Update Frequency Cellular Automaton - Output 10



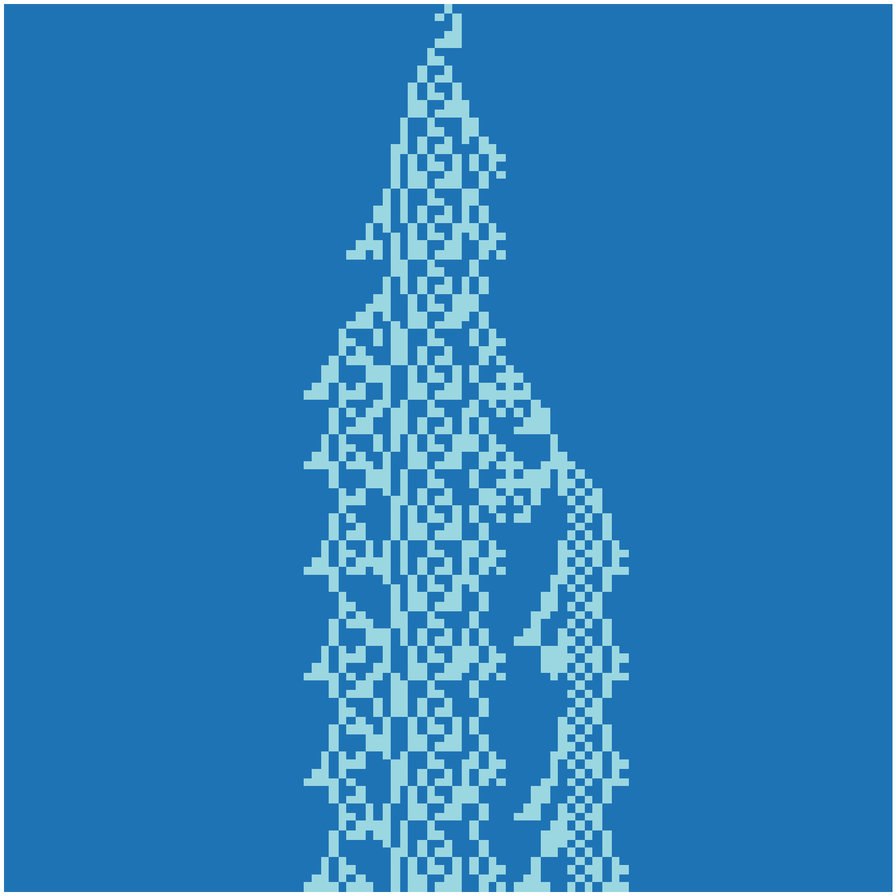
Rule 242 - Spatially Varying Update Frequency Cellular Automaton - Output 9



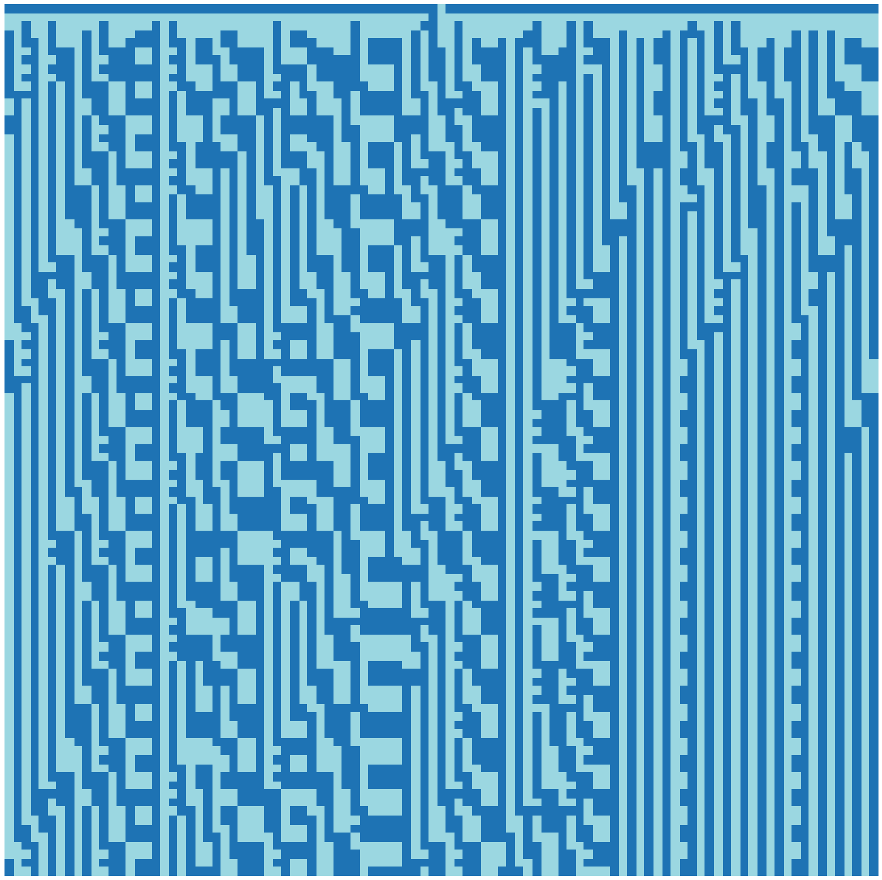
Rule 111 - Spatially Varying Update Frequency Cellular Automaton - Output 5



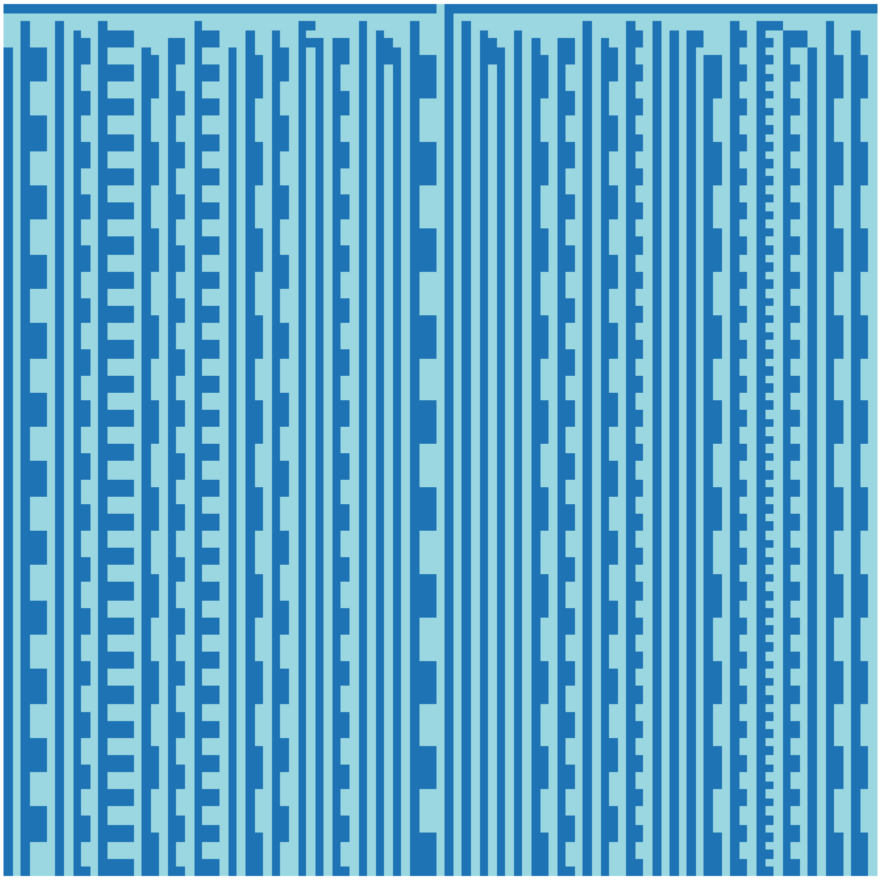
Rule 50 - Spatially Varying Update Frequency Cellular Automaton - Output 7



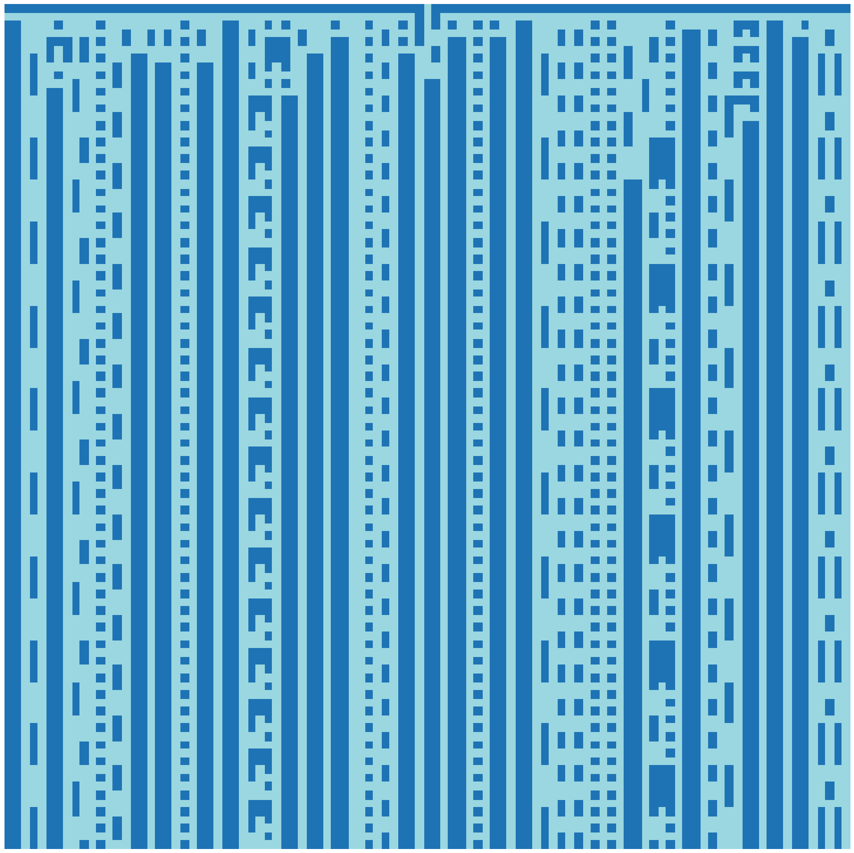
Rule 21 - Spatially Varying Update Frequency Cellular Automaton - Output 21

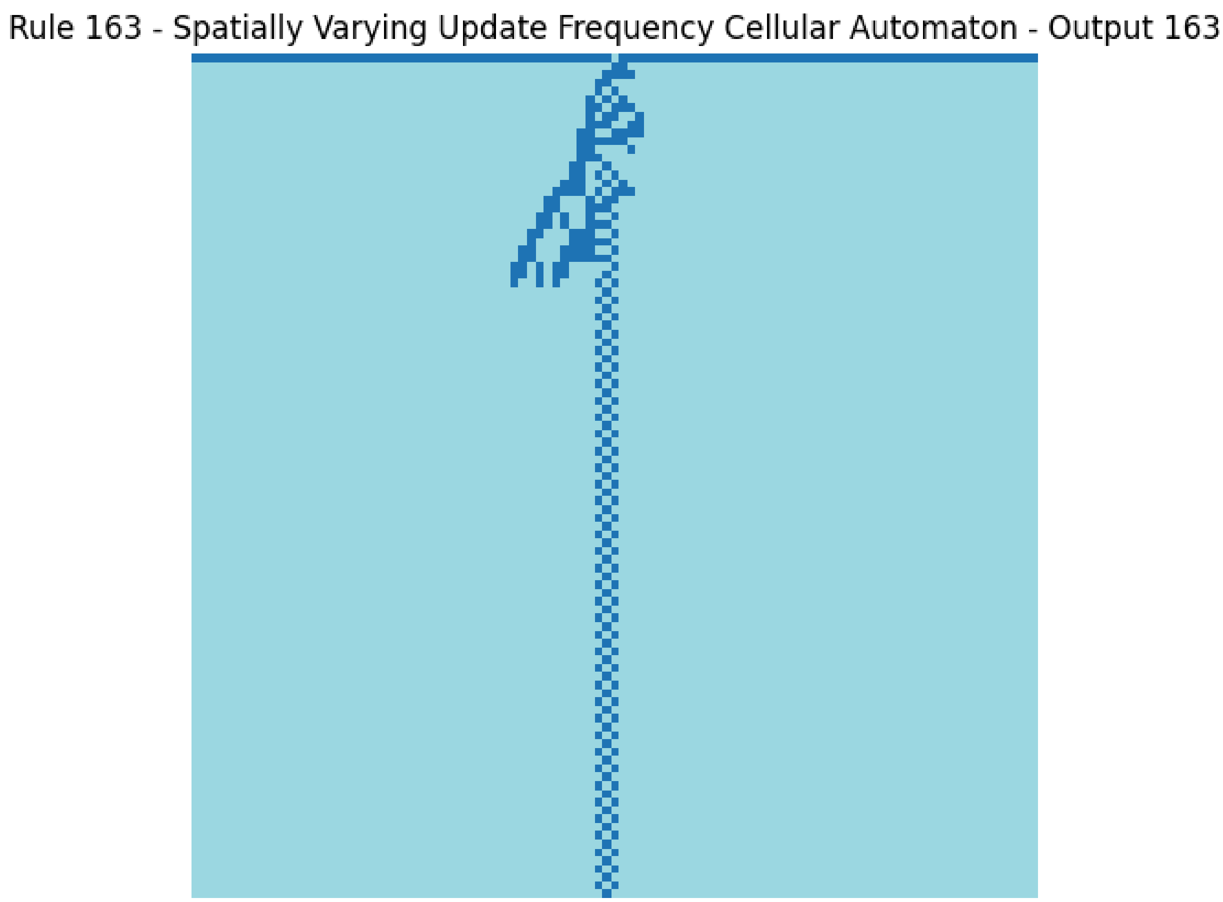


Rule 71 - Spatially Varying Update Frequency Cellular Automaton - Output 71



Rule 109 - Spatially Varying Update Frequency Cellular Automaton - Output 109





Game of Life

a) Existing Model Considered

The existing model is a cellular automaton similar to Conway's Game of Life with these rules:

- Local Rule: All cells follow the same rule for determining "alive" or "dead", based on a uniform threshold.
- Neighborhood Dependency: Each cell updates based on the 8 surrounding cells, following a uniform rule (e.g., "alive if 3 neighbors are alive").

- Update Pattern: All cells are updated synchronously, with each cell's new state calculated at the same time.

The model uses uniform initialization, synchronous updates, and a consistent neighborhood rule for all cells.

b) The Way Uniformity is Broken

In the proposed model:

- Local Rule:
  - The randomness is added to the local rule, making the cell’s next state probabilistic, based on a random threshold for each pixel. The uniformity is broken because the rule that dictates whether a cell becomes alive or not is no longer deterministic for every cell.
- Neighborhood Dependency:
  - While the neighborhood dependency (checking 8 neighbors) remains the same, the randomness applied to each cell introduces a non-deterministic outcome based on the same neighborhood, breaking the uniformity in how the outcome is determined.
- Update Pattern:
  - While cells still update together synchronously, the introduction of randomness per cell adds a non-deterministic behavior within the synchronized update.

c) Details of the Proposed Model

The proposed model introduces randomness:

- Random Threshold:
  - A random threshold (randomThreshold) based on pixel position and frame number determines if a cell stays alive.
- Random Probability:
  - A probability value influences whether a cell updates its state, adding non-determinism.
- Key Change:
  - Cells no longer follow a deterministic rule, as their state is influenced by randomness.

d) Best Possible Simulation Patterns

Our website contains all the generated simulations for the updated model:

Link: https://evolutionofprogram.wixsite.com/toc-project