Regional Controllability of Boolean Cellular Automata

Team 16:
Michelle Bruch, Avina Chetana Kalle,
Paripurna Dutta, Sol Guiomar Montero Carrasco,
and Farhan Zahin

Friedrich-Alexander-University Erlangen-Nuernberg

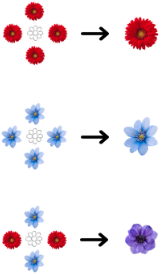
Summer Semester

Index

Introduction

Cellular automata are simple computational models that,

1. Consist of a **grid of cells**.
2. Each with a state that can **change** over time based on a set of **predefined rules**.
3. At each discrete time step, the state of each cell is updated **simultaneously** according to the rules
4. Typically **depends** on the **current state** of the cell and its **neighboring cells**.
5. These rules **determine** how the cells **evolve** and transition to different states.

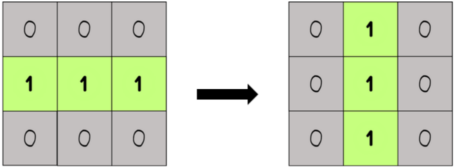


Conway's Game of Life

- Devised by the British mathematician John Horton Conway in 1970.
- It is a zero-player game.
- One interacts with the Game of Life by creating an initial configuration and observing how it evolves.

Example: Conway's Game of Life

1. **Live** cell with < 2 live neighbors $\rightarrow$ **DIES** due to underpopulation.
2. **Live** cell with > 3 live neighbors $\rightarrow$ **DIES** due to overpopulation.
3. **Live** cell with $2/3$ live neighbors $\rightarrow$ **Stays Alive** (Balanced).
4. **Dead** cell with $= 3$ live neighbors $\rightarrow$ **Becomes Alive** (Reproduction).



Preliminary theory

Definition 0

A cellular automaton (CA) is defined by a tuple $A = (\mathcal{L}, \mathcal{S}, \mathcal{N}, f)$, where:

1. $\mathcal{L}$ is a cellular space.
2. $\mathcal{S}$ is a finite set of possible states.
3. $\mathcal{N}$ is a function that defines the neighborhood of a cell c. We denote:

$$\begin{aligned} \mathcal{N} : \mathcal{L} &\longrightarrow \mathcal{L}^r \\ c &\longmapsto \mathcal{N}(c) = (c_{i_1}, c_{i_2}, \cdots, c_{i_r}) \end{aligned}$$

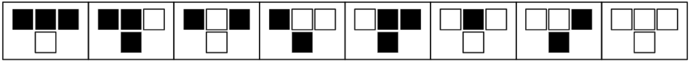
   where c_{i_j} is a cell for $j = 1, \cdots, r$ and r is the size of the neighborhood $\mathcal{N}(c)$ of the cell c.

4. f is the transition function. It is defined as follow:

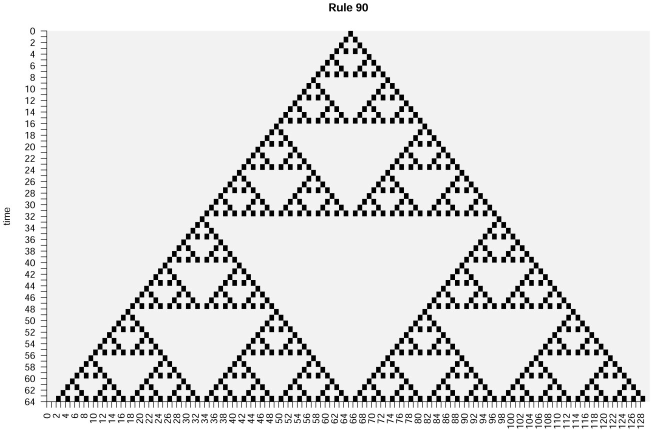
$$\begin{aligned} f : \mathcal{S}^r &\to \mathcal{S} \\ s_t(\mathcal{N}(c)) &\longrightarrow f(s_t\mathcal{N}(c)) = s_{t+1}(c) \end{aligned}$$

   where $s_t(c)$ is the state of a cell c at time t and $s_t(\mathcal{N}(c))$ is the state of the neighborhood of c. [3, 4]

rule 90



0 1 0 1 1 0 1 0



Definition 1 (Global transition function)

The configuration of a CA at time t corresponds to the set $\{s_t(c), c \in \mathcal{L}\}$.

$$F : \mathcal{S}^{\mathcal{L}} \longrightarrow \mathcal{S}^{\mathcal{L}}$$
$$\{s_t(c), c \in \mathcal{L}\} \longrightarrow \{s_{t+1}(c), c \in \mathcal{L}\}$$

We denote the region we want to control by $\omega = \{c_1, \cdots, c_n\}$. [3, 4, 7].

Definition 2 (regional controllable)

The CA is said to be regionally controllable for ω at time T if there exists a control sequence $u = (u_0, \cdots, u_{T-1})$ where $u_i = (u_i(c0), u_i(c_{n+1}))$, $i = 0, \cdots, T-1$ such that:

$$s_T = s_d \quad on \quad \omega$$

where s_T is the final configuration at time T and s_d is the desired configuration [3, 4].

Problem

- A 1D-cellular domain $\mathcal{L}$ of $\mathcal{N}$ cells.
- $S = \{0, 1\}$
- $N(c_i) = (c_{i-1}, c_i, c_{i+1})$
- f is a linear function.
- ω defines controlled region where $\omega = \{c_1, \cdots, c_n\} \subseteq L$.
- The boundary cells of ω where we apply the control are$\{c_0, c_{n+1}\}$.

Aim

To find two suitable sequences of controls to be applied on the left and right boundaries, in order to reach a desired configuration on the subregion ω at a given time T from an initial state.

[1, 3].

Controllability Theorem

The state equation of a Boolean cellular automata can be written as :

$$\begin{cases} s^{t+1} & = Js^t \oplus Bu^t; \quad 0 \leq t \leq T-1 \\ s^0 & \in S^L \end{cases}$$

Theorem 3

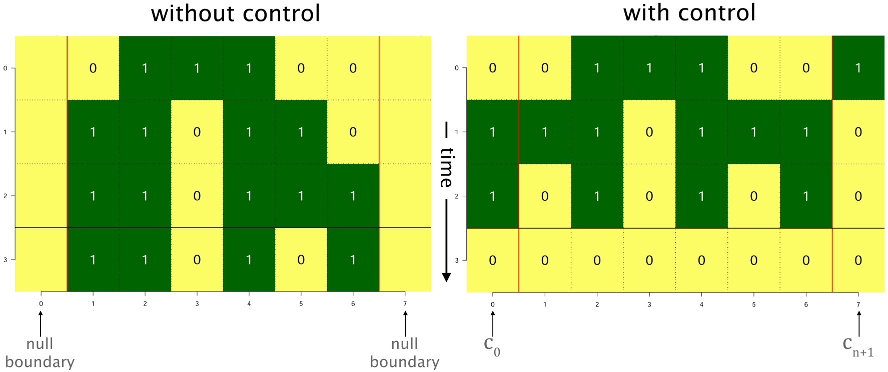
A 1D linear Cellular Automaton is regionally controllable via boundary actions iff:

$$Rank(C) = Rank(B, JB, J^2B, ..., J^{T-1}B) = T \tag{1}$$

Where T is the time horizon and J is the Jacobian matrix. [3]

Wolfram's rule 90

$$f_{90}: \begin{array}{cccccccc} 111 & 110 & 101 & 100 & 011 & 010 & 001 & 000 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{array}$$



Wolfram's rule 90

For the Wolfram rule 90 we get the following Jacobian matrix for $|\omega| = 6$, $T = 6$ and the following B:

$$J = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

From the above, we get the following control matrix:

$$C = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 0 & 3 & 0 \\ 0 & 1 & 0 & 2 & 0 & 5 \\ 1 & 0 & 1 & 0 & 2 & 0 \end{pmatrix}$$

This Control matrix C has full rank then rule 90 is regional controllable.

Algorithm

Step 1 : *Define constants*	**Inputs** :	1.	rule number	:	rule_no
		2.	length of grid	:	n
		3.	time steps	:	T
		4.	initial state	:	$s_0\,(c_1, \ldots, c_n)$
		5.	control	:	$(l, r) = (u\,(c_0)\,, u\,(c_{n+1}))$
	Constants :	6.	neighbourhood	:	$\mathcal{N}\,(c_i)$
		7.	possible states	:	$\mathcal{S} = \{0, 1\}$
Step 2 :	**Initialize automaton**	1.	Create empty grid of size $n \times T$	:	ECA
		2.	set initial condition	:	$\text{ECA}\,[1,] = s_0$
		3.	convert rule no. into 8 bit binary	:	$bin\_rule = \{b_1, b_2, \ldots, b_8\}$
Step 3 :	**Define update function**		$f_{\text{rule\_no}}\,(\mathcal{N}\,(c_i))$	:	eg. $\{1, 1, 1\} \mapsto b_1$
Step 4 :	**Simulate ECA**	1.	For $t \in [1, T-1]$		
		2.	For $i \in [2, n-1]$		
			Compute next state	:	$s\,(c_i)_{t+1} = f_{\text{rule\_no}}\,(\mathcal{N}_t)$
			For $i = 1, n$		
			set control boundary	:	$(c_0, c_{n+1}) = (l, r)$
			Compute next state	:	$s\,(c_i)_{t+1} = f_{\text{rule\_no}}\,(\mathcal{N}_t)$
			Update ECA	:	$\text{ECA}\,[t+1,] = s_{t+1}$
Step 5 :	**Visualize**		Plot ECA		

Definition 4 (Transition graph)

For a given rule number and size ($|\omega|$), the transition graph $\Upsilon = (V, E)$ is defined by edges (E) between possible states (vertices, V).

Definition 5 (Hamiltonian cycle)

A Hamiltonian cycle of a graph G = (V, A) is a simple directed path of G that includes every vertex exactly once.[3]

Theorem 6

If there exists a Hamiltonian cycle in Υ, then the CA is regionally controllable.

Example: Rule 90

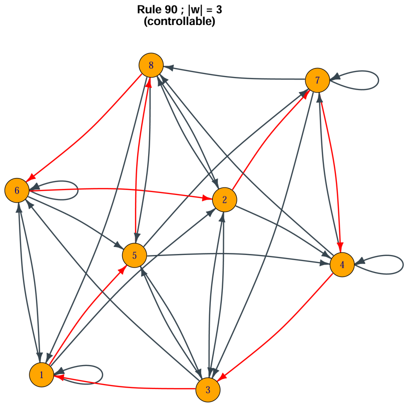


Figure: Transition graph

$$\Upsilon = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Figure: Transition matrix

Example: Controllable but not Hamiltonian

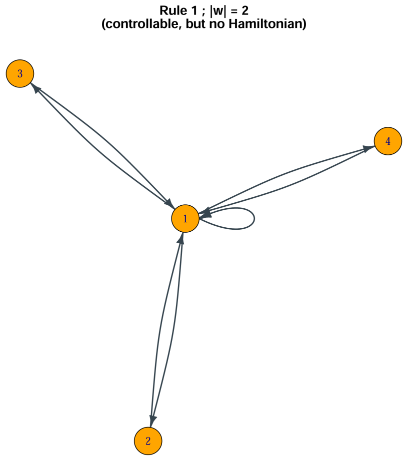


Figure: Transition graph

$$\Upsilon = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Figure: Transition matrix

Transformation matrix

Definition 7 (Transformation matrix)

The (i, j)th element of the transformation matrix, $\mathfrak{C}(t; rule, |\omega|)$, corresponds to the number of paths of length t from vertex i to j.

Theorem 8

A Cellular Automaton is regionally controllable if and only if there exists a t such that the graph associated to the transformation matrix $\mathfrak{C}(t; rule, |\omega|)$ contains a Hamiltonian cycle.

Example: Rule 1

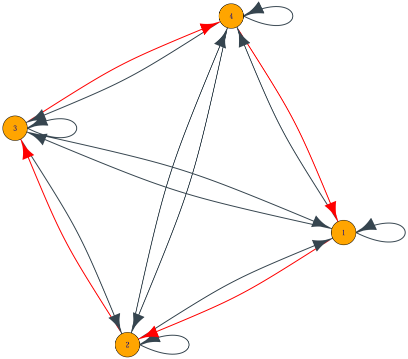


Figure: graph of C(2)

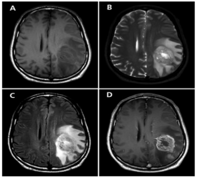
$$\mathfrak{C}(1) = \begin{pmatrix} 1 & 2 & 2 & 2 \\ 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}$$

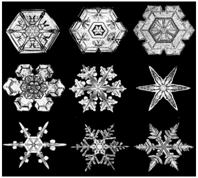
$$\mathfrak{C}(2) = \begin{pmatrix} 3 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} > 0$$

Figure: Transformation matrix

Successful use of CCA in real life

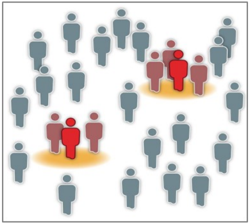
- Biology and Medicine:
  - Modeling cell behavior and growth patterns.
  - Understanding disease spread and epidemiological patterns.
  - Simulating tumor growth and treatment responses.
- Physics and Chemistry:
  - Simulating crystal growth and material properties.
  - Studying phase transitions and fluid dynamics.
  - Analyzing chemical reactions and diffusion processes.

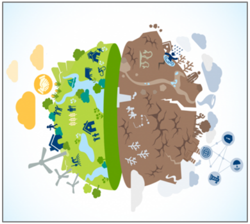




Successful use of CCA in real life

- Ecology and Environment:
  - Modeling ecosystem dynamics and biodiversity.
  - Analyzing climate change and environmental disruptions.





Limitations

- Computational Complexity.
- Lack of Continuous Representation.
- Limited Interaction Range.
- Simplistic Rules and Behavior.

Conclusions

- The problem of the regional controllability of Boolean CA was studied and some necessary and sufficient conditions were established using:
  1. Graph Theory
  2. Kalman condition

Bibliography I

Beros, A., Chyba, M., and Markovichenko, O. (2019). Controlled cellular automata. *Networks and Heterogeneous Media*, 14:1–22.

Borchers, H. (2022). W. r: Finds a hamiltonian path or cycle.

Dridi, S., El Yacoubi, S., and Bagnoli, F. (2021). Kalman condition and new algorithm approach for regional controllability of peripherally-linear elementary cellular automata via boundary actions.

Dridi, S., El Yacoubi, S., Bagnoli, F., and Fontaine, A. (2019). A graph theory approach for regional controllability of boolean cellular automata.

El, S. and El Jai, A. (2002). Cellular automata modelling and spreadability. *Pergamon Mathematical and Computer Modelling Mathematical and Computer Modelling*, 36:1059–1074.

El Yacoubi, S., Plénet, T., Dridi, S., Bagnoli, F., Lefèvre, L., and Raïevsky, C. (2021). Some control and observation issues in cellular automata.

Hassan, E., Said, Z., and Castillo, M. O. (2020). Studies in systems, decision and control 243 recent advances in modeling, analysis and systems control: Theoretical aspects and applications.

Ognyanova, K. (2021). Static and dynamic network visualization with r.

Rhodes, A. D. (2013). The algebraic structure of cellular automata.

Vichniac, G. (1990). Boolean derivatives on cellular automata. *Physica D*, 45:63–74.

How do we know if a graph has a Hamiltonian graph?

Definition 9

A Hamiltonian graph is a path that visits each vertex of the graph exactly once.

- Brute force search algorithms would be very slow because given an n-vertex graph, there might be n! different possible paths to check for Hamiltonian cycles.
- In this project, a backtracking algorithm (from the 'adagio') was implemented

Nondeterministic polynomial problem

A problem is called NP (nondeterministic polynomial)if its solution can be guessed and verified in polynomial time; nondeterministic means that no particular rule is followed to make the guess. For example:

- Sudoku game.
- The Traveling Salesman Problem.
  - “Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?”

Transition graph

- $\Upsilon(V, A)$ is the transition graph with V as vertices representing each possible configuration of the region ω and A as the set of arcs
- For $v_1, v_2 \in V$: if there exists a arc between v_1 and v_2 that means there exists a control $u = (l, r) \in \{(0, 0)(0, 1), (1, 0), (1, 1)\}$ such that the rule applied on v_1 using u maps to v_2.

Proof of Theorem 6

If a graph $\Upsilon = (V, E)$ contains a Hamiltonian cycle, then there exists a path between any two arbitrary vertices, s_i and s_j.

This path (of length T) is a sequence of edges that represents a control sequence $\{u_1, ..., u_T\}$ such that s_i goes to s_j in T time steps.

Hence, considering the arbitrary states s_i and s_j to be the initial condition and the desired configuration respectively, we have shown that the CA corresponding to Υ is regionally controllable.