The Quest for Small Universal Cellular Automata

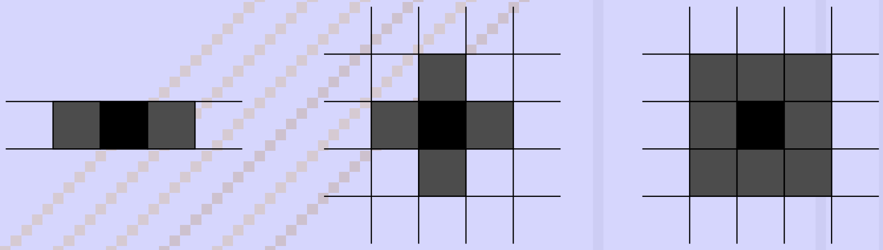
Nicolas Ollinger
LIP, ENS Lyon, France

8 july 2002 / ICALP’2002 / Málaga, Spain

Cellular Automata

Definition. A d-*CA* $\mathcal{A}$ is a 4-uple $(\mathbb{Z}^d, S, N, \delta)$ where:

- S is the finite state set of $\mathcal{A}$;
- $N \subset \mathbb{Z}^d$, finite, is the neighborhood of $\mathcal{A}$;

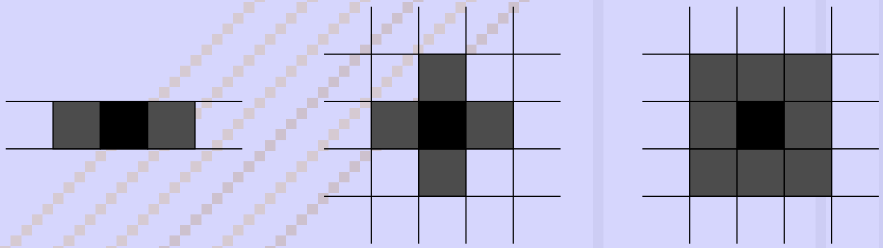


- $\delta : S^{|N|} \to S$ is the local rule of $\mathcal{A}$.

Cellular Automata

Definition. A d-*CA* $\mathcal{A}$ is a 4-uple $(\mathbb{Z}^d, S, N, \delta)$ where:

- S is the finite state set of $\mathcal{A}$;
- $N \subset \mathbb{Z}^d$, finite, is the neighborhood of $\mathcal{A}$;



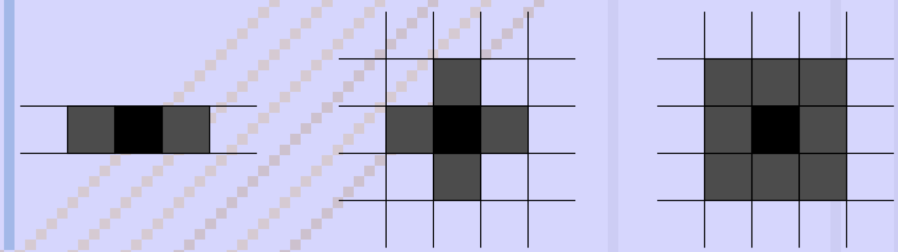
- $\delta : S^{|N|} \to S$ is the local rule of $\mathcal{A}$.

A *configuration* C is a mapping from $\mathbb{Z}^d$ to S.

Cellular Automata

Definition. A d-*CA* $\mathcal{A}$ is a 4-uple $(\mathbb{Z}^d, S, N, \delta)$ where:

- S is the finite state set of $\mathcal{A}$;
- $N \subset \mathbb{Z}^d$, finite, is the neighborhood of $\mathcal{A}$;



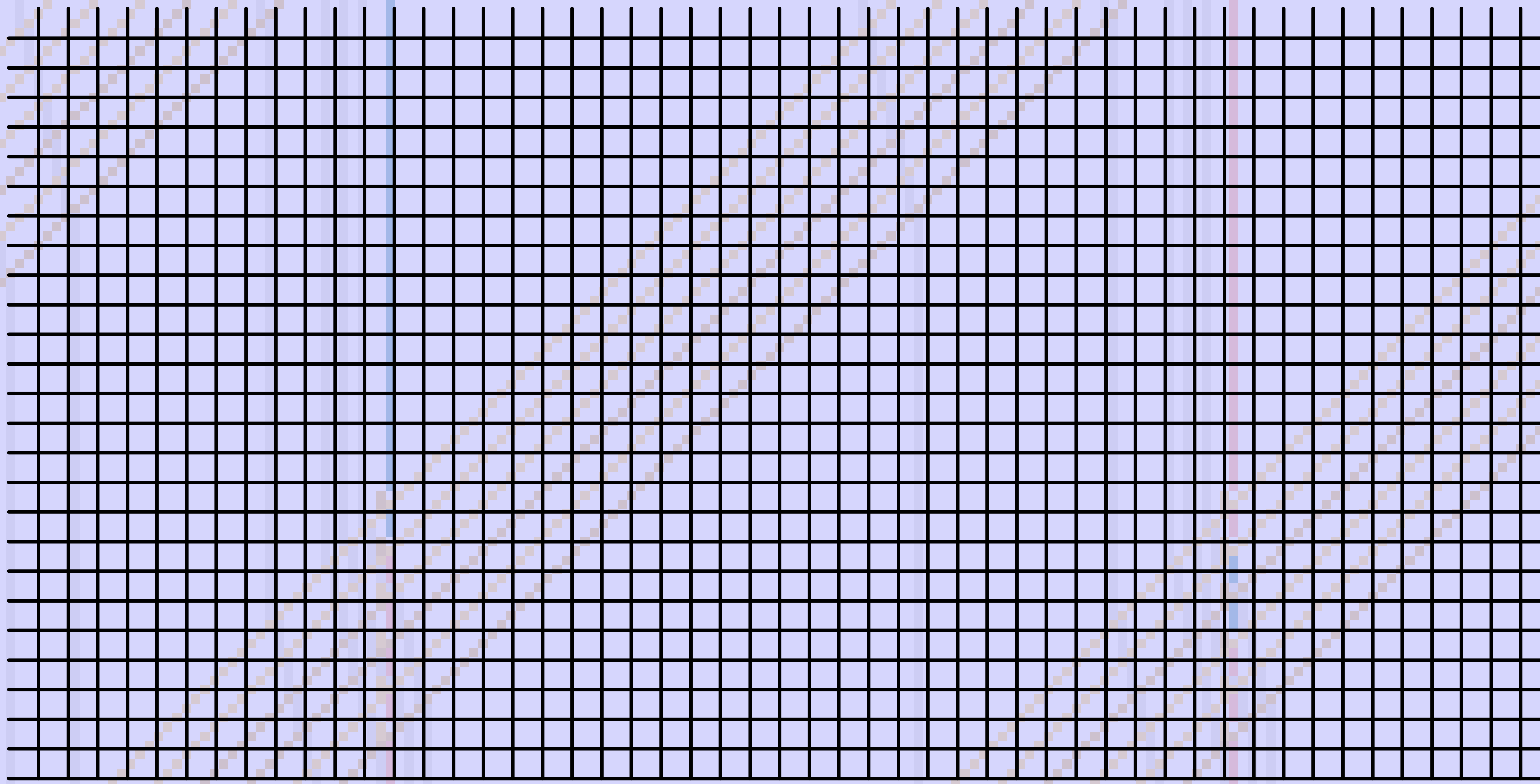
- $\delta : S^{|N|} \to S$ is the local rule of $\mathcal{A}$.

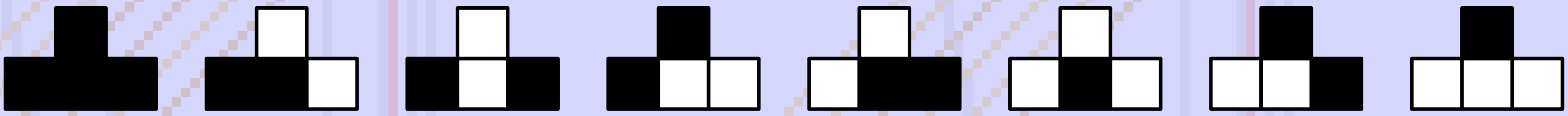
A *configuration* C is a mapping from $\mathbb{Z}^d$ to S.

The *global rule* applies δ uniformly according to N:

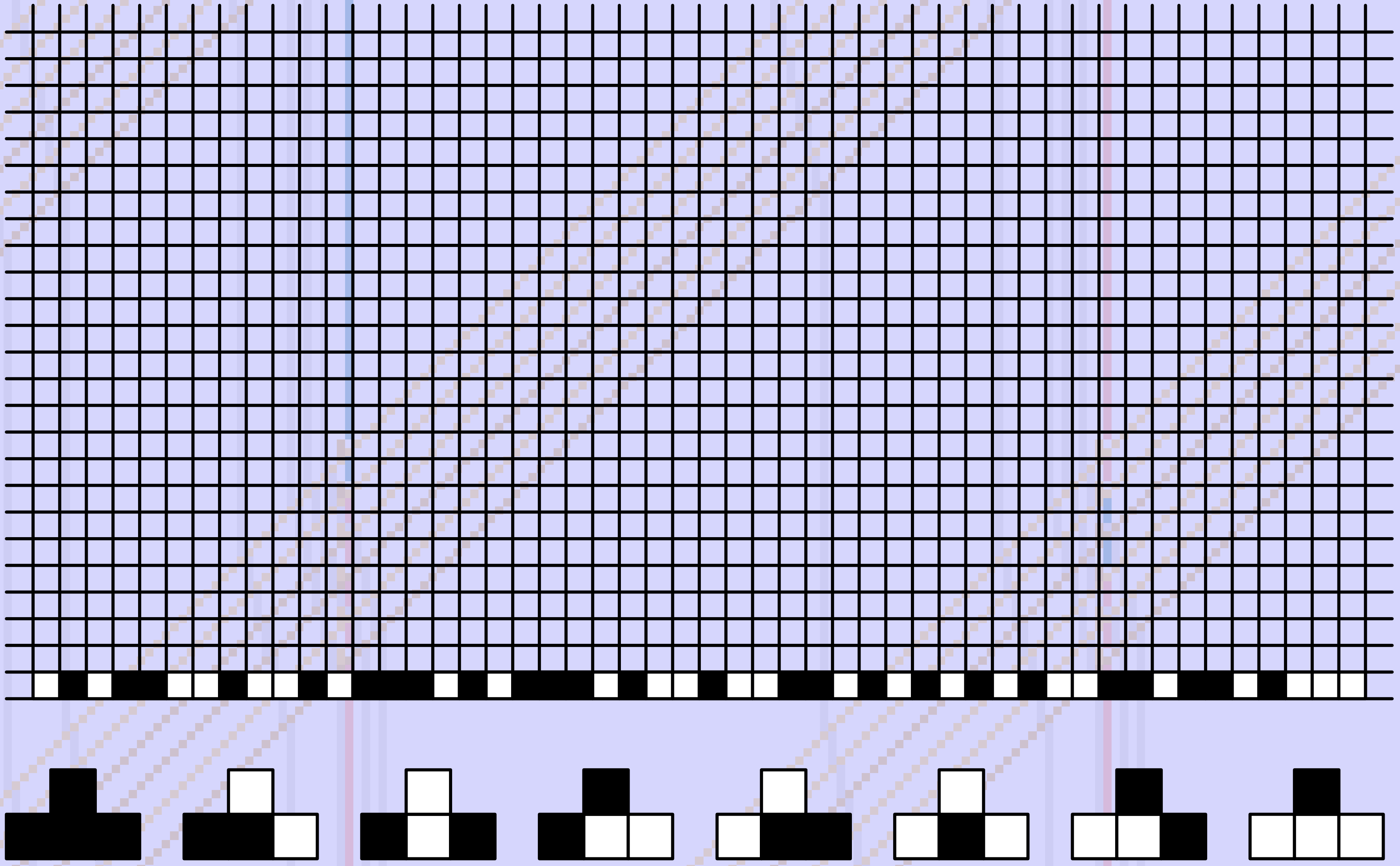
$$\forall p \in \mathbb{Z}^d, \quad G(C)_p = \delta\left(C_{p+N_1}, \ldots, C_{p+N_\nu}\right)$$

Space-Time Diagram

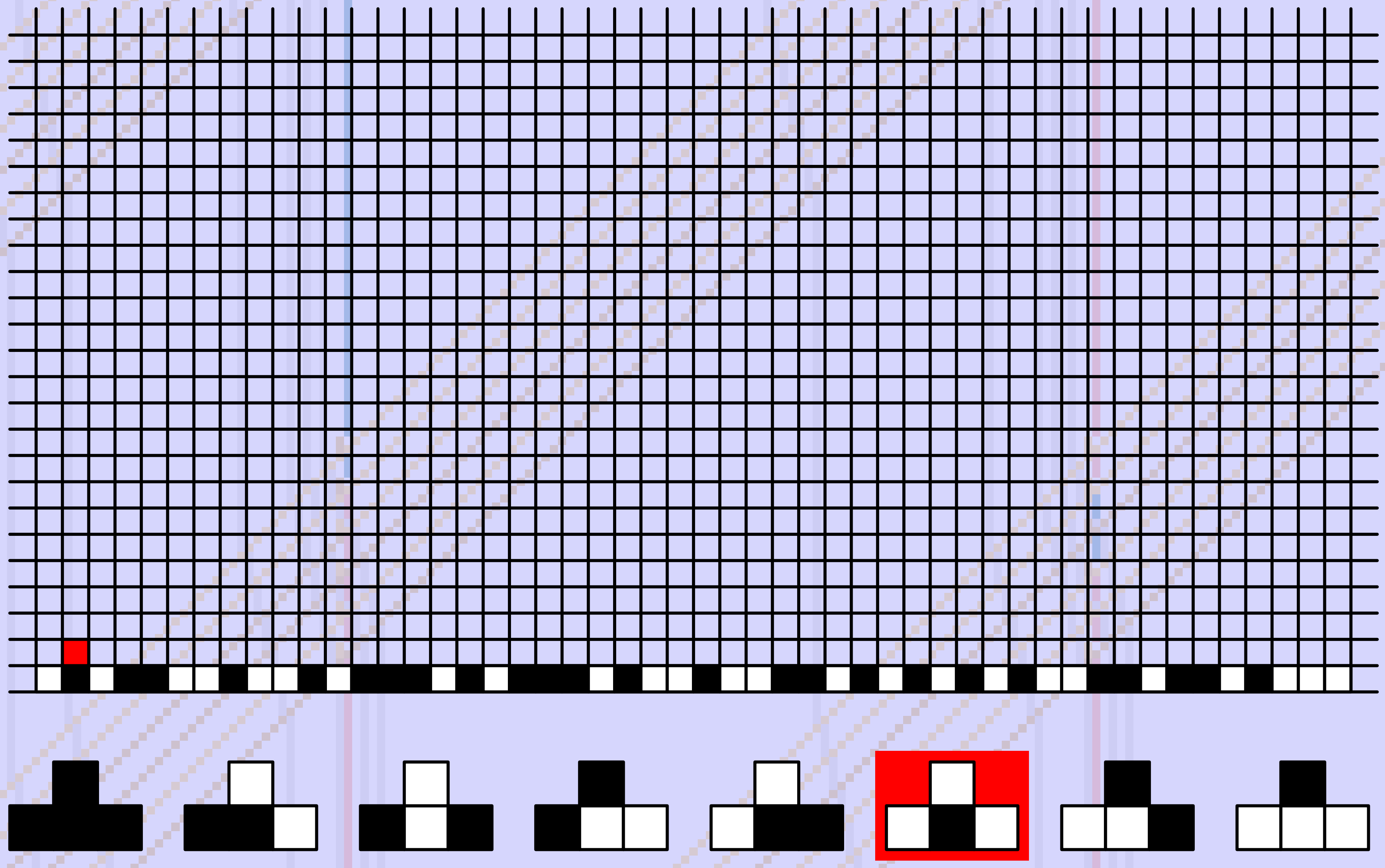




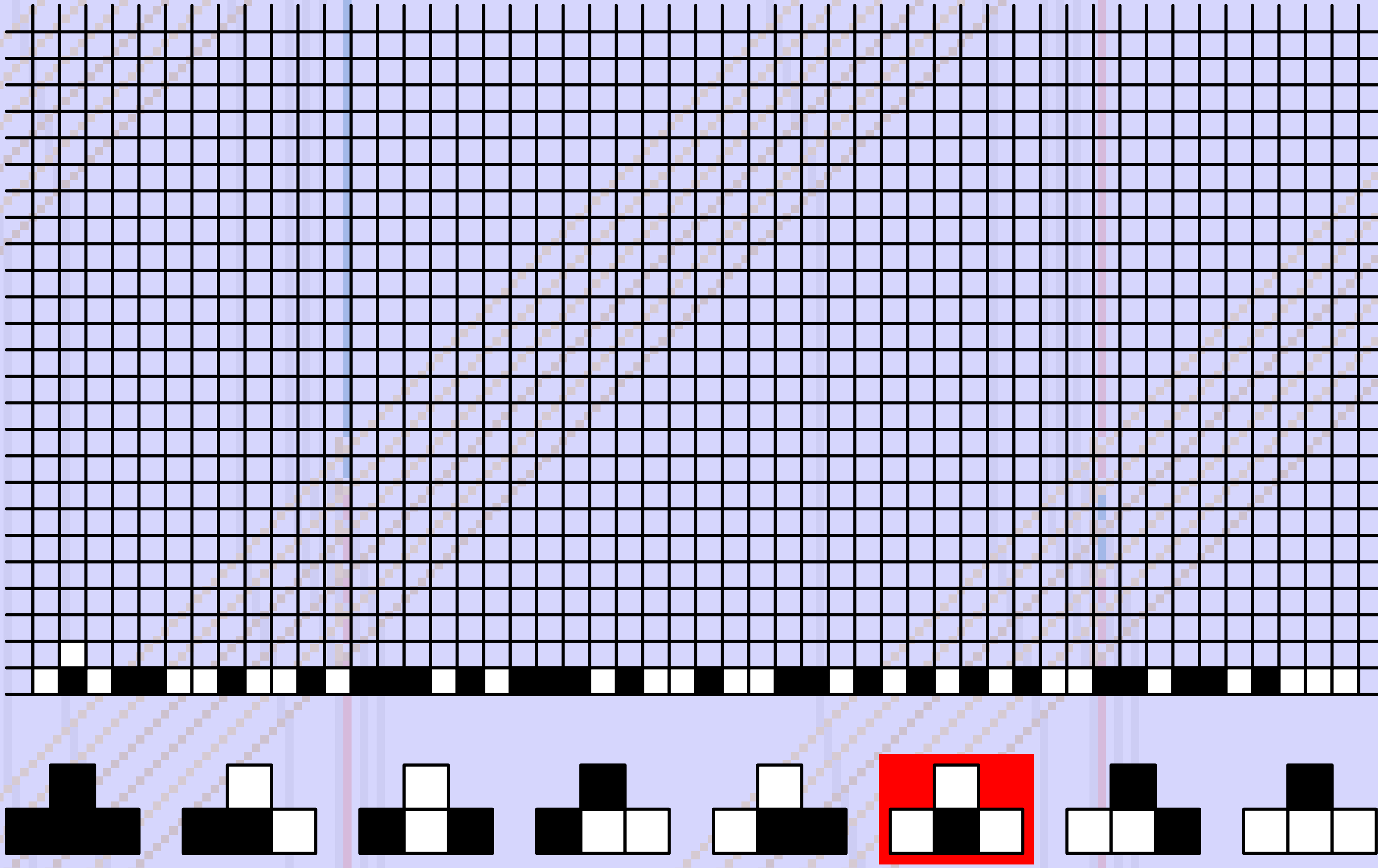
Space-Time Diagram



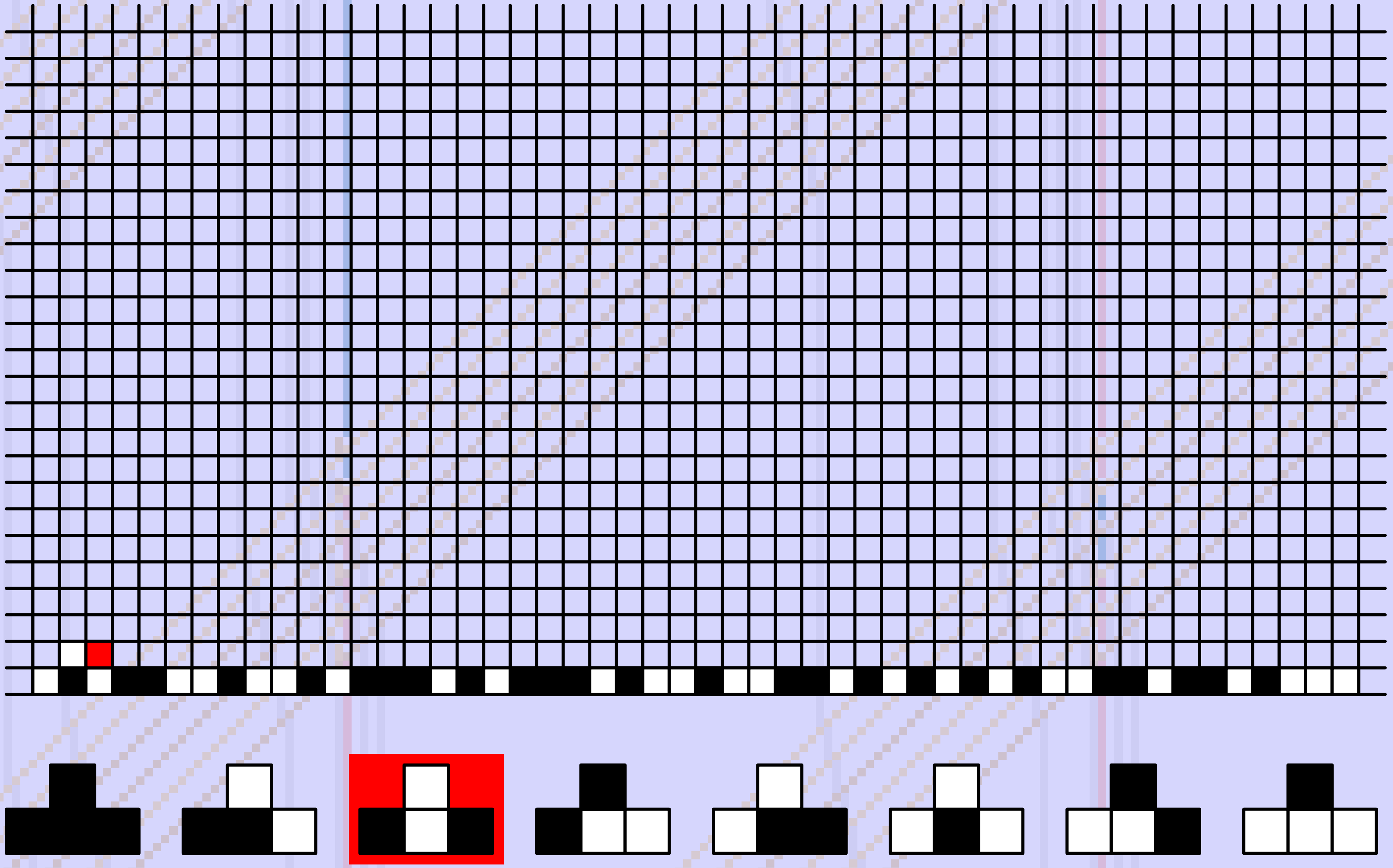
Space-Time Diagram



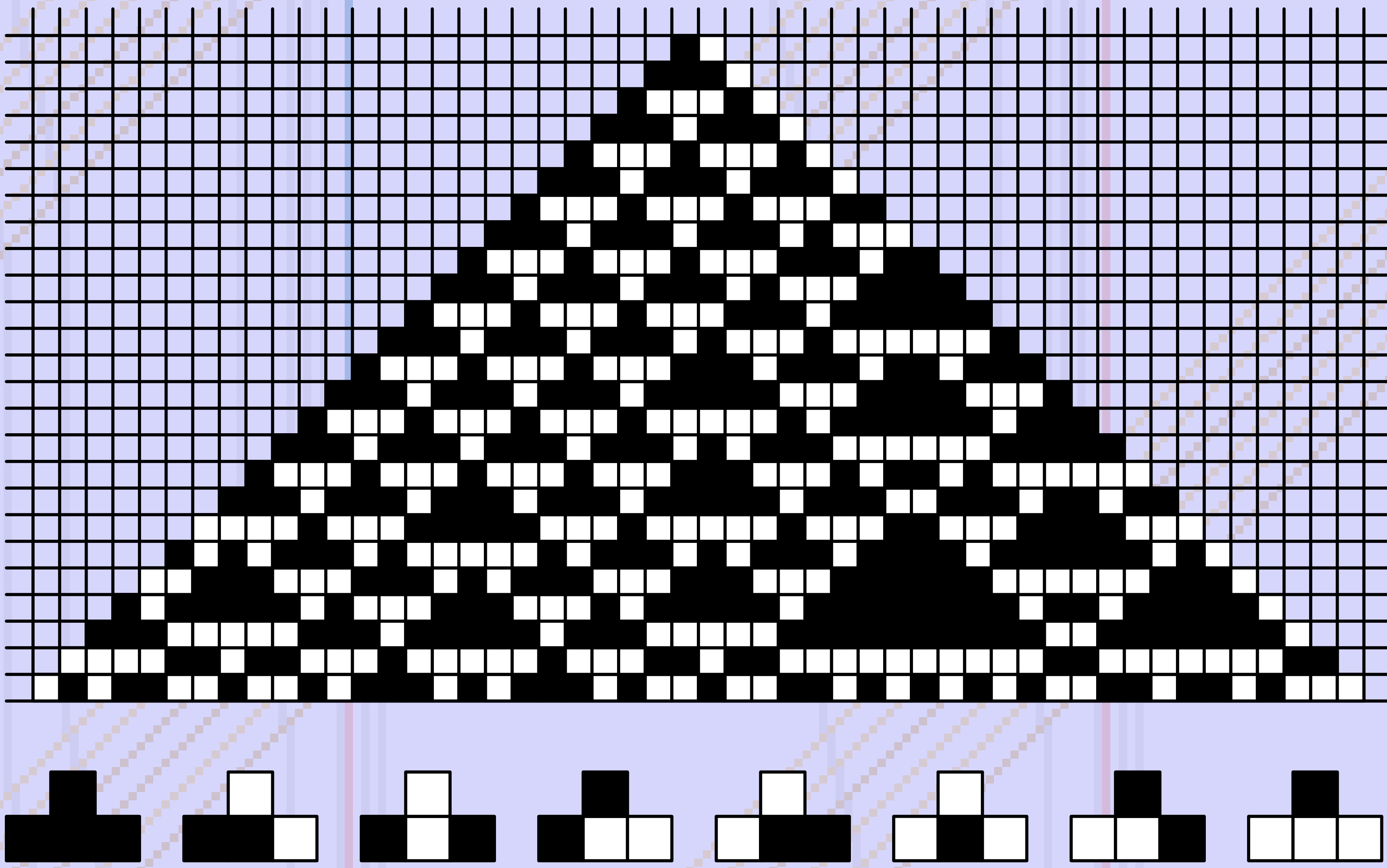
Space-Time Diagram



Space-Time Diagram



Space-Time Diagram



Computation Universality

Idea. A CA is *computation universal* if it can **compute** any partial recursive function.

Computation Universality

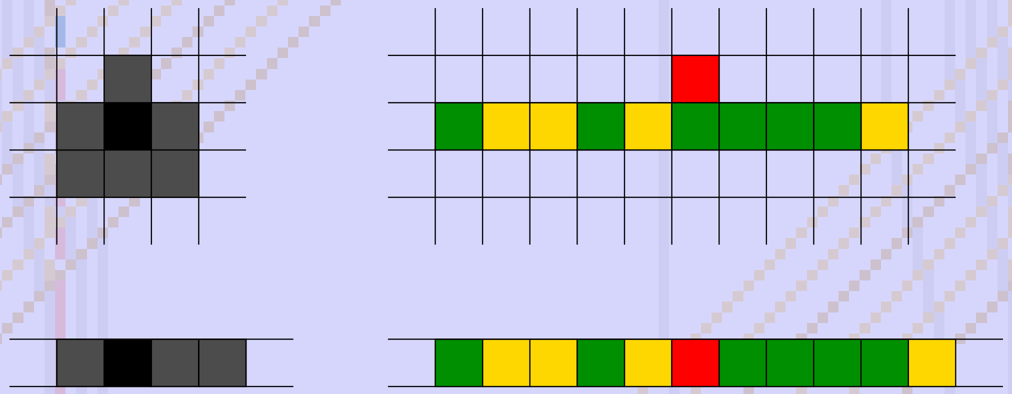
Idea. A CA is *computation universal* if it can **compute** any partial recursive function.

- This notion is rather difficult to formalize...

Computation Universality

Idea. A CA is *computation universal* if it can **compute** any partial recursive function.

- This notion is rather difficult to formalize...
- In practice: step-by-step Turing machine simulation.



A. R. Smith III. Simple Computation-Universal Cellular Spaces. 1971

Inducing an Order on CA (1)

Idea. A CA $\mathcal{A}$ is **less complex** than a CA $\mathcal{B}$ if, up to some renaming of states and some rescaling, every space-time diagram of $\mathcal{A}$ is a space-time diagram of $\mathcal{B}$.

Inducing an Order on CA (1)

Idea. A CA $\mathcal{A}$ is **less complex** than a CA $\mathcal{B}$ if, up to some renaming of states and some rescaling, every space-time diagram of $\mathcal{A}$ is a space-time diagram of $\mathcal{B}$.

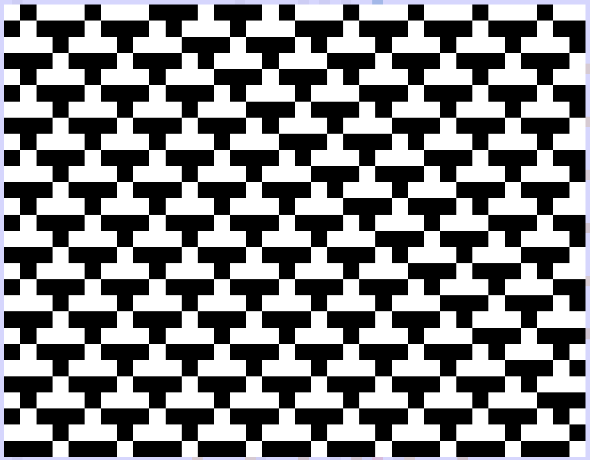
Definition. $\mathcal{A} \subseteq \mathcal{B}$ if there exists an injective mapping φ from $S_{\mathcal{A}}$ into $S_{\mathcal{B}}$ such that this diagram commutes:

$$
\begin{array}{ccc}
C & \xrightarrow{\ \varphi\ } & \overline{\varphi}(C) \\
{\scriptstyle G_{\mathcal{A}}}\big\downarrow & & \big\downarrow{\scriptstyle G_{\mathcal{B}}} \\
G_{\mathcal{A}}(C) & \xrightarrow[\ \varphi\ ]{} & \overline{\varphi}(G_{\mathcal{A}}(C))
\end{array}
$$

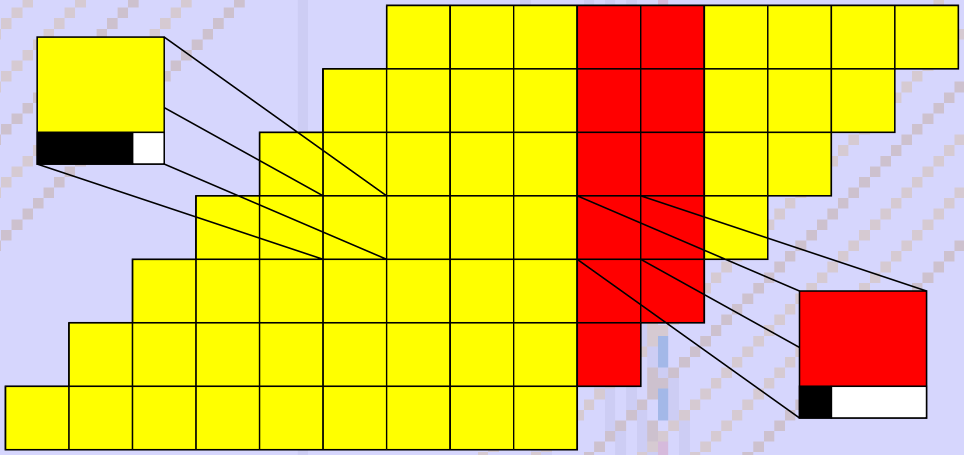
Inducing an Order on CA (2)

Definition. The $\langle m, n, k \rangle$ rescaling of $\mathcal{A}$ is defined by:

$$G_{\mathcal{A}}^{\langle m,n,k \rangle} = \sigma^k \circ o^m \circ G_{\mathcal{A}}^n \circ o^{-m} .$$



$\mathcal{A}$

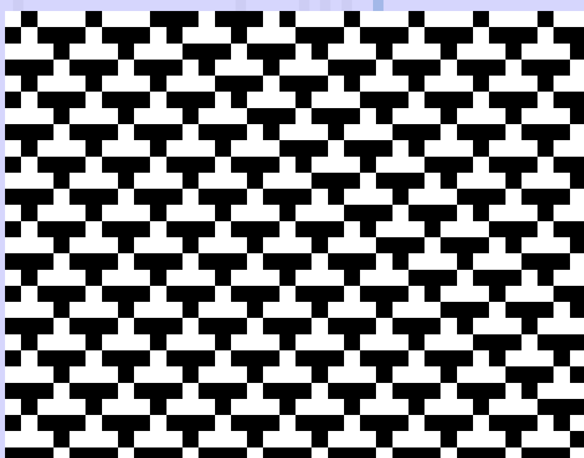


$\mathcal{A}^{\langle 4,4,1 \rangle}$

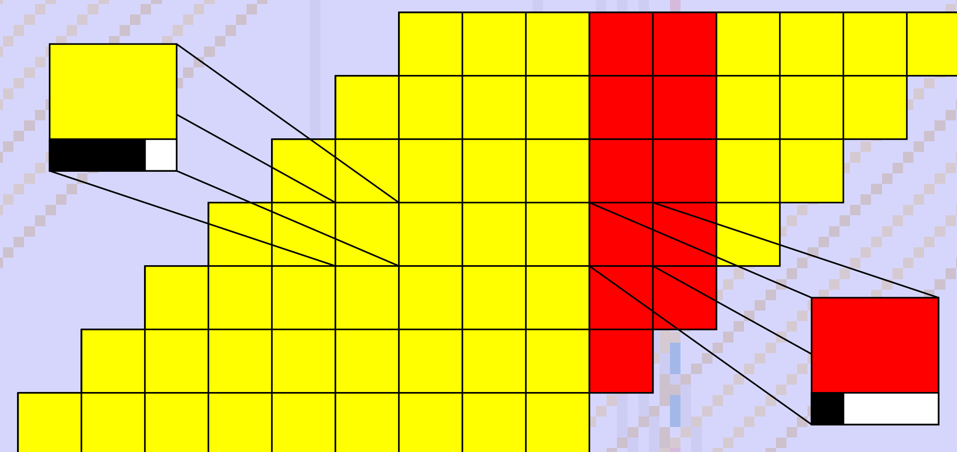
Inducing an Order on CA (2)

Definition. The $\langle m, n, k\rangle$ rescaling of $\mathcal{A}$ is defined by:

$$G_{\mathcal{A}}^{\langle m,n,k\rangle} = \sigma^{k} \circ o^{m} \circ G_{\mathcal{A}}^{n} \circ o^{-m} \;.$$



$\mathcal{A}$



$\mathcal{A}^{\langle 4,4,1\rangle}$

Definition. $\mathcal{A} \leqslant \mathcal{B}$ if there exist $\langle m, n, k\rangle$ and $\langle m', n', k'\rangle$ such that $\mathcal{A}^{\langle m,n,k\rangle} \subseteq \mathcal{B}^{\langle m',n',k'\rangle}$.

Inducing an Order on CA (3)

Proposition. The relation $\leqslant$ is a quasi-order on CA.

- The induced order admits a maximal equivalence class.

Definition. A CA $\mathcal{A}$ is *intrinsically universal* if:

$$\forall \mathcal{B}, \exists \langle m, n, k \rangle \,, \quad \mathcal{B} \subseteq \mathcal{A}^{\langle m,n,k \rangle} \,.$$

Inducing an Order on CA (3)

Proposition. The relation $\leqslant$ is a quasi-order on CA.

- The induced order admits a maximal equivalence class.

Definition. A CA $\mathcal{A}$ is *intrinsically universal* if:

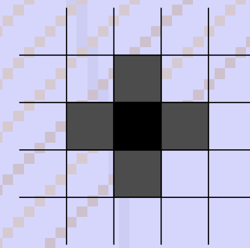
$$\forall \mathcal{B}, \exists \langle m, n, k \rangle \,, \quad \mathcal{B} \subseteq \mathcal{A}^{\langle m,n,k \rangle} \,.$$

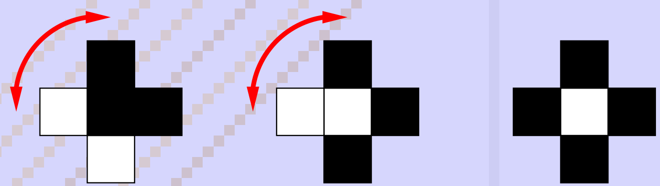
Proposition. Every intrinsically universal CA is computation universal. **The converse is false**.

Simple Universal CA

year	author	d	$\vert N \vert$	states	universality
1966	von Neumann	2	5	29	intrinsic
1968	Codd	2	5	8	intrinsic
1970	Banks	**2**	**5**	**2**	**intrinsic**
		1	3	18	intrinsic
1971	Smith III	2	7	7	computation
		1	3	18	computation
1987	Albert & Culik II	1	3	14	intrinsic
1990	Lindgren & Nordhal	**1**	**3**	**7**	**computation**
2002	NO	**1**	**3**	**6**	**intrinsic**
2002	Cook & Wolfram	**1**	**3**	**2**	**computation**

Banks' Universal 2D-CA

$\left(\mathbb{Z}^2, \{\blacksquare, \square\},\right.$  $\left., \delta\right)$



E. R. Banks. Universality in Cellular Automata. 1970

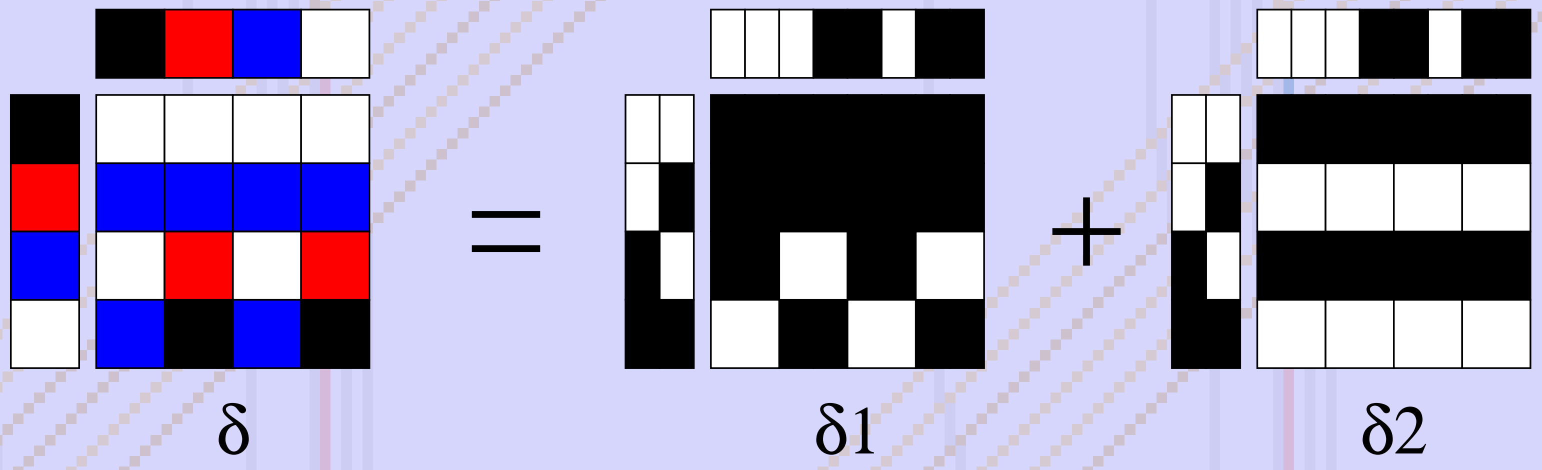
Idea. Emulate logical circuits by building:

- wires transporting binary signals
- logical gates AND, OR and NOT
- wires crossing

CA and Boolean Circuits (1)

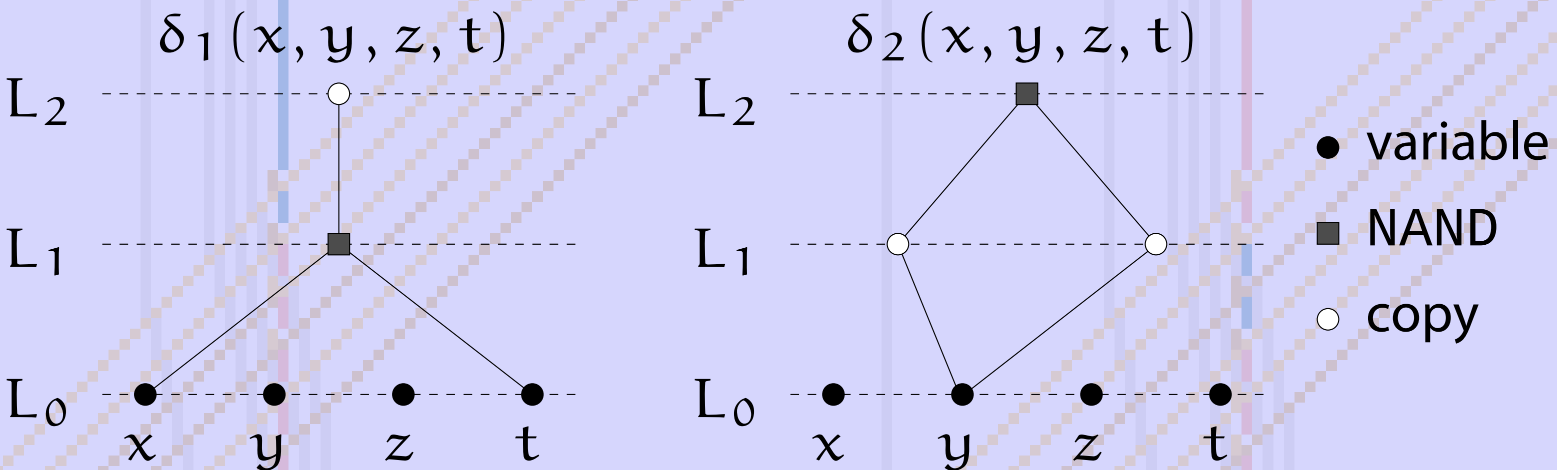
- We decompose a CA local rule into k boolean functions where $k = \lceil \log_2 |S| \rceil$:

$$\delta_i : \{0,1\}^{|N|k} \to \{0,1\}.$$



CA and Boolean Circuits (2)

- To a boolean function we associate a leveled circuit:



$\delta_1(x, y, z, t) = x \mid t$ $\delta_2(x, y, z, t) = y \mid y$

Boolean Circuit Simulator

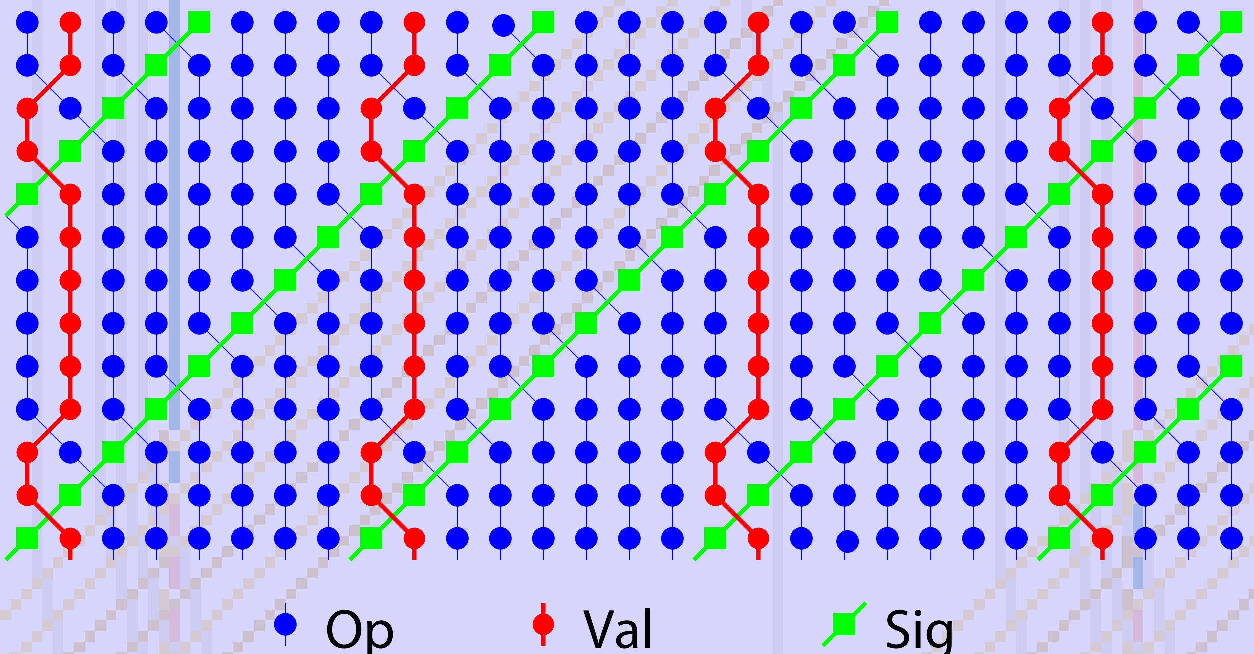
A BC Simulator is a 1D dynamical system that simulate a CA via its boolean circuits. Each cell contains:

- a boolean value;
- an operator (identity or NAND);
- the relative positions of the operands.

		δ_1		δ_2			δ_1		δ_2			δ_1		δ_2			
boolean value	· · ·	?	x	?	?	y	?	z	?	?	t	?	u	?	?	v	· · ·
operator		|	c	c	c	|	|	c	c	c	|	|	c	c	c	|	
operand 1		1	1	8	9	1	1	1	8	9	1	1	1	8	9	1	
operand 2		9				2	9				2	9				2	
				s_0					s_1					s_2			

Microscopic Description

We build a 3-state 1D-CA to move information.

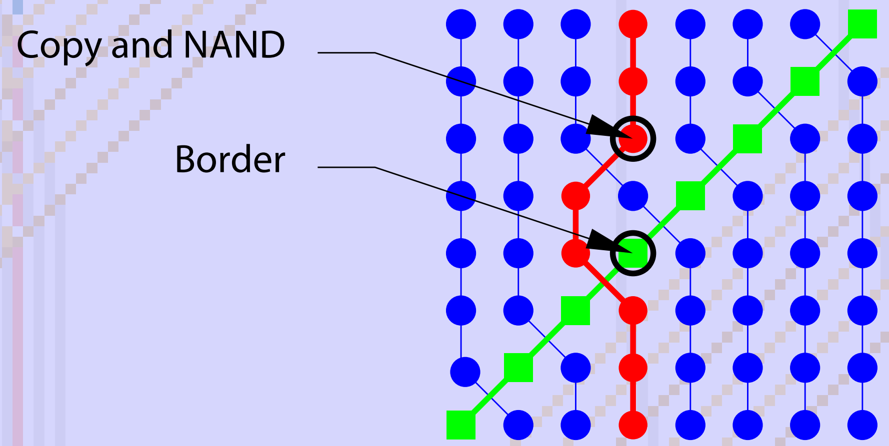


- Sig cells transport boolean values between cells;
- Val cells encode current meta-cell value;
- Op vells encode the operation to execute.

8 States

Direct encoding:

- Sig: 0 or 1 boolean value;
- Val: 0 or 1 boolean value;
- Op: Border, Copy, Follow or NAND operation.



7 States

New encoding:

- Sig: 0 or 1 boolean value;
- Val: 0 or 1 boolean value;
- Op: Border, Follow or NAND operation.

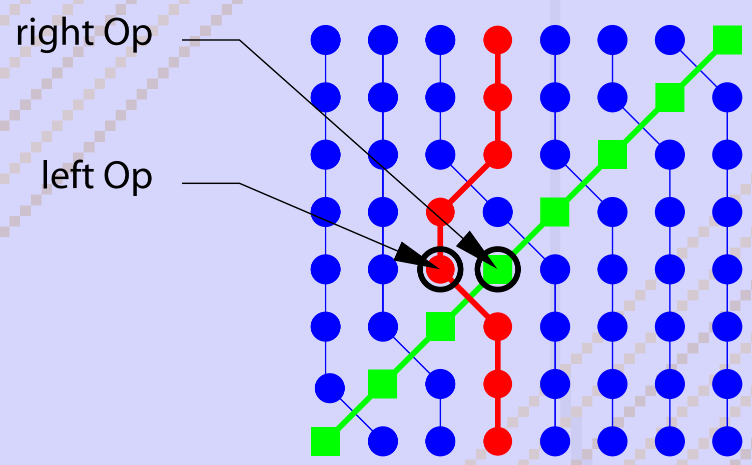
The Copy operation is emulated. We encode a signal x by three consecutive signals $1, x, 0$.

old Op	new 3 Ops
Border	Follow, Border, Follow
Follow	Follow, Follow, Follow
NAND	Follow, NAND, Follow
Copy	**NAND, NAND, NAND**

6 States

More tricky!

New encoding with Op a boolean value which meaning becomes position dependant...



Going further

- We get 6 states as a product 3×2. What about 2×2?

- Cook and Wolfram have proven a 2 states rule computation universal. Is it also intrinsically universal ?

- Good formal definition of computation universality ?