Santiago, Chile

June 2001

“Simple” Intrinsically Universal Cellular Automata

Nicolas Ollinger

Laboratoire de l’Informatique du Parallélisme

École Normale Supérieure de Lyon

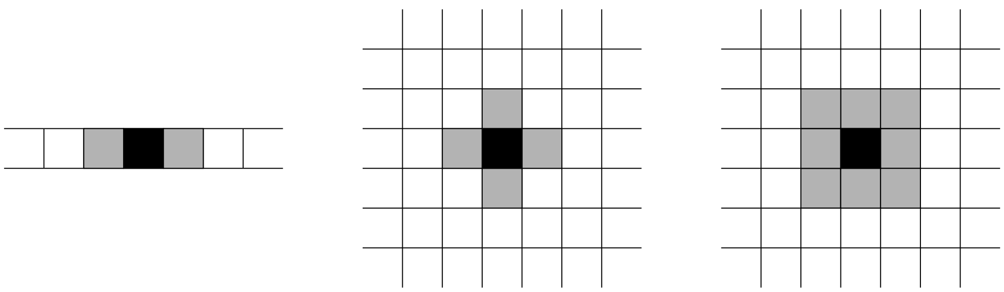
Unité Mixte de Recherche CNRS-INRIA-ENS LYON n^{o} 5668

Contents

Cellular Automata

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

- S the finite set of states of $\mathcal{A}$,
- $N \subset \mathbb{Z}^d$, finite, the neighborhood of $\mathcal{A}$,
- $\delta : S^{|N|} \to S$ 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:

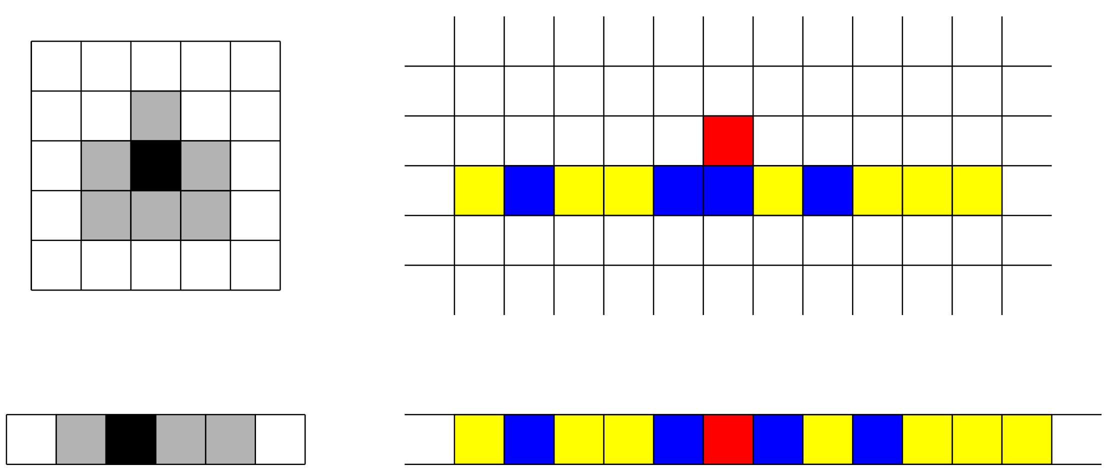
$$G_{\mathcal{A}}\left(C\right)_p = \delta\left(C_{p+N_1}, \ldots, C_{p+N_n}\right).$$

Computation Universality

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

▶ This notion is difficult to formalize.

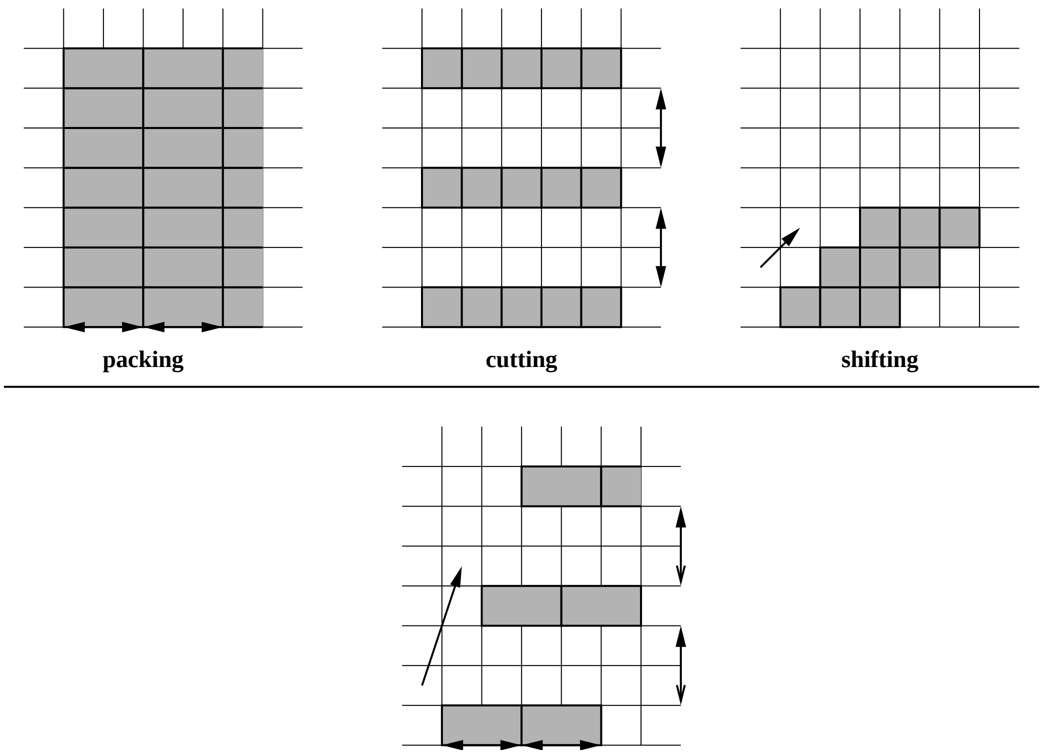
▶ In practice: step-by-step Turing simulation.



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

Intrinsic Simulation

► Geometrical transformations on space-time diagrams:



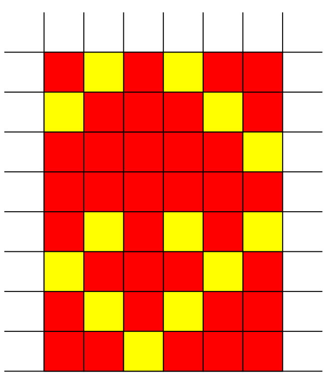
Definition. A d-CA $\mathcal{A}$ *simulates* a d-CA $\mathcal{B}$ if, up to geometrical transformations on both sides, space-time diagrams from $\mathcal{B}$ can be considered as space-time diagrams from $\mathcal{A}$.

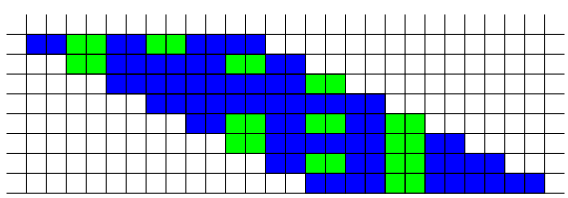
Intrinsic Universality

Definition. A d-CA $\mathcal{A}$ is *intrinsically universal* if $\mathcal{A}$ can simulate any d-CA $\mathcal{B}$.

▶ This definition is equivalent to the following one.

Definition. A d-CA $\mathcal{A}$ is *strongly intrinsically universal* if, for each d-CA $\mathcal{B}$, space-time diagrams from $\mathcal{B}$ can be considered as space-time diagrams from a geometrical transformation of $\mathcal{A}$.





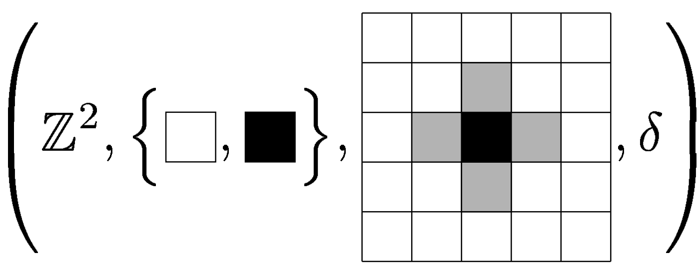
“Simple” Universal CA

year	author	d	n	states	universality
1966	von Neumann	2	5	**29**	intrinsic
1968	Codd	2	5	**8**	intrinsic
1971	Smith	2	7	**7**	computation
1970	Banks	2+	5	**2**	intrinsic
1971	Smith	1	3	**18**	computation
1970	Banks	1	3	**18**	intrinsic
1987	Albert Čulik II	1	3	**14**	intrinsic
1990	Lindgren Nordahl	1	3	**7**	computation

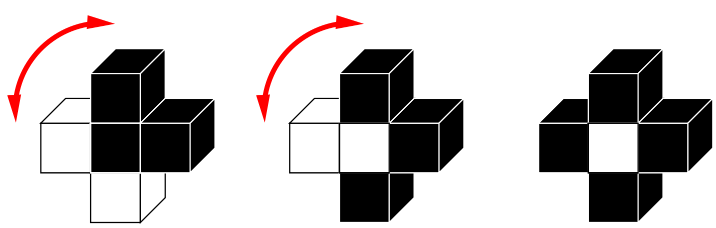
Contents

Banks' Universal CA

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

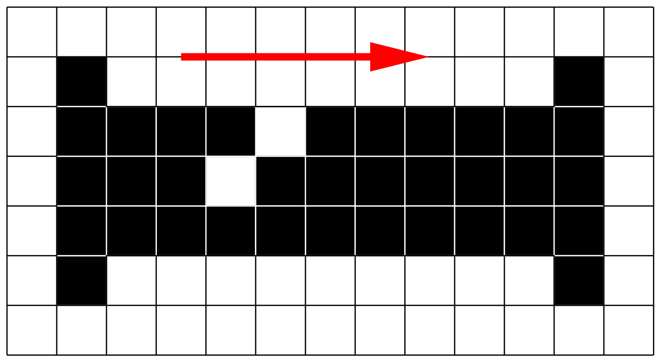


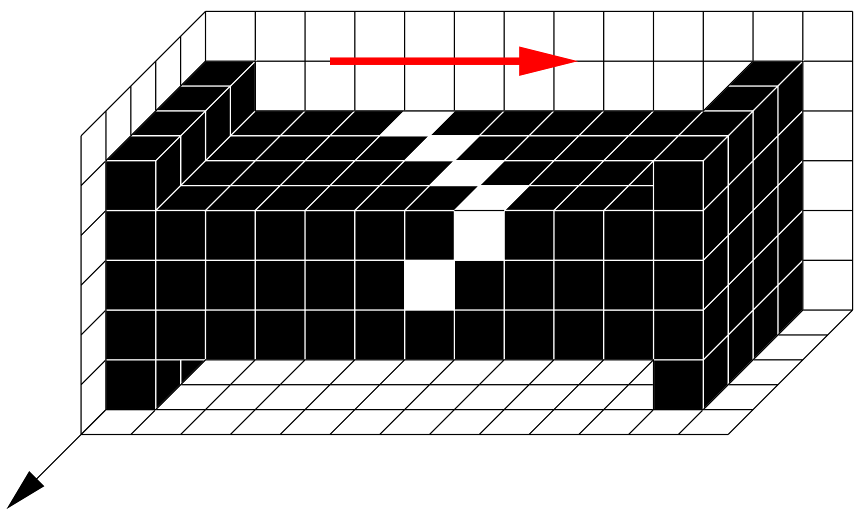
- Neighborhood configurations leading to state change:



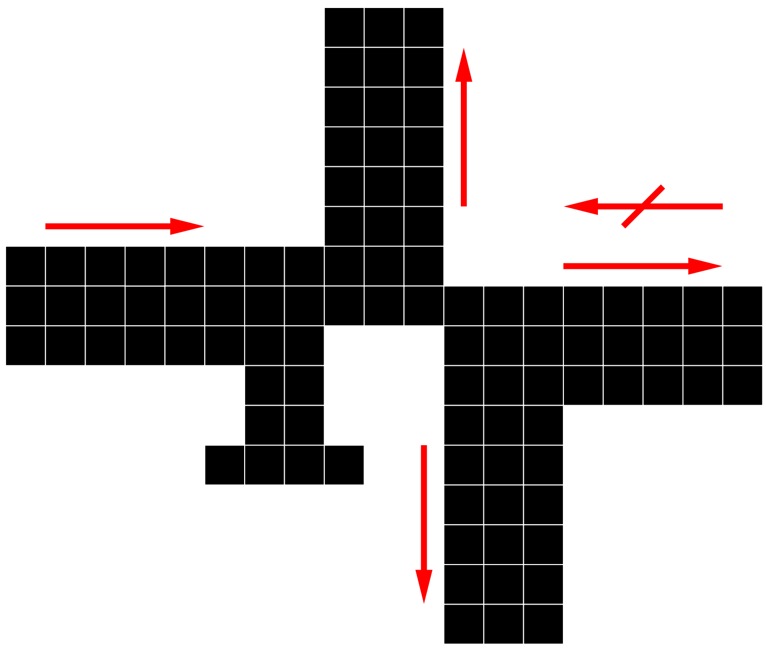
- Universality is proven by boolean circuit simulation.

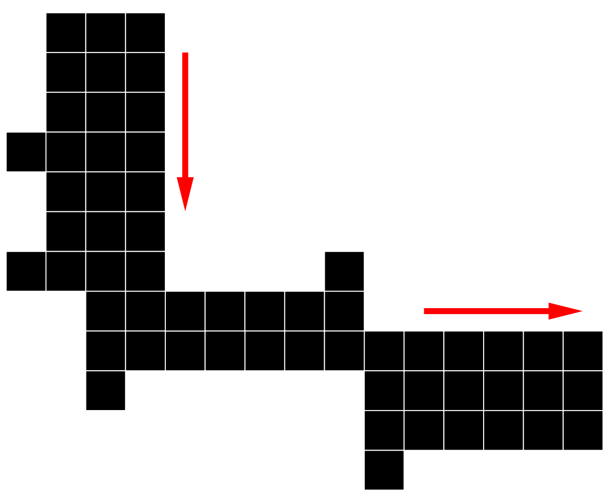
Signals



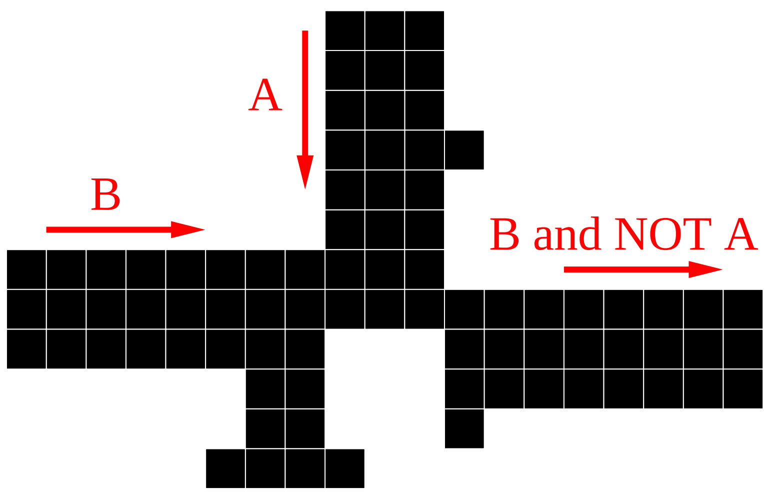


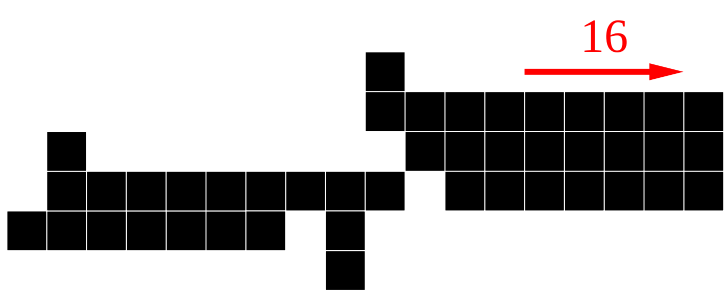
Paths and Fan-out





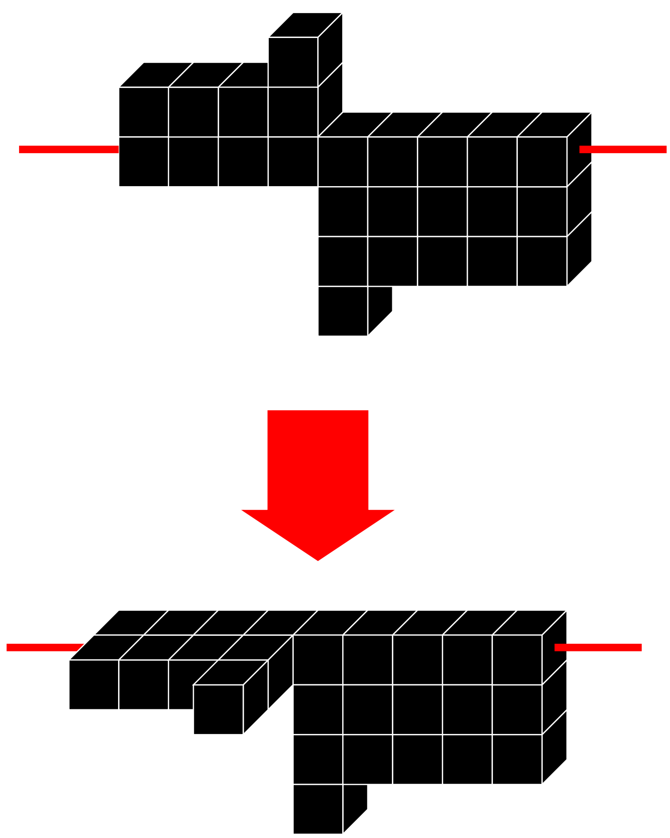
Logic and Clock





Higher Dimensions

- The rule naturally extends to higher dimensions.
- We need a construction to turn in another dimension:



Contents

Linear and Bilinear CA

Definition. A *linear* d-CA is a polynomial d-CA of degree 1. S is a finite commutative ring.

$$\delta(s_1, \ldots, s_n) = \sum_{i=1}^{n} a_i s_i$$

► Linear d-CA exhibit simple behavior (Martin *et al.* 1993)

Definition. A *bilinear* d-CA is a polynomial d-CA of degree 2. S is a finite commutative ring.

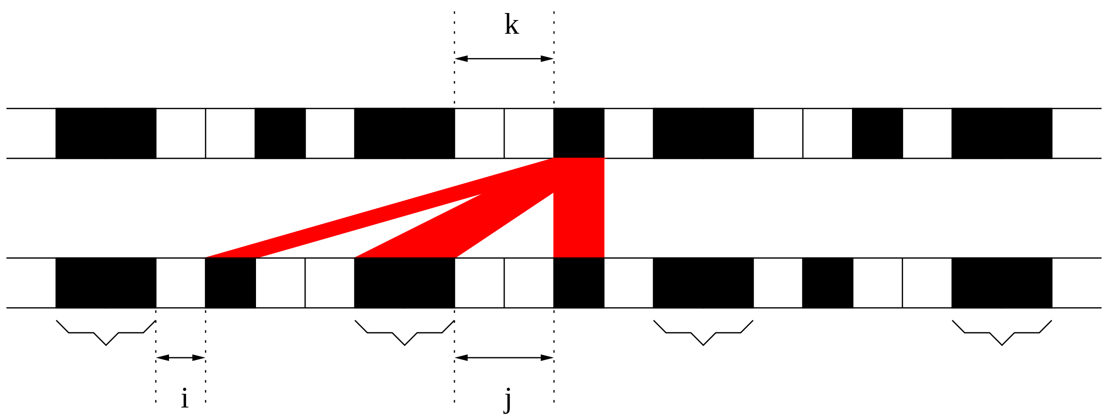
$$\delta(s_1, \ldots, s_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} b_{i,j} s_i s_j$$

► There are intrinsically universal bilinear d-CA

At least over the ring $\mathbb{Z}_{211}^{211}$ (Bartlett and Garzon 1995)

Two-state Universal CA

▶ We construct a simple polynomial 1-CA over $\mathbb{Z}_2$ which simulate a given one-way 1-CA (*i.e.* $N = \{-1, 0\}$).

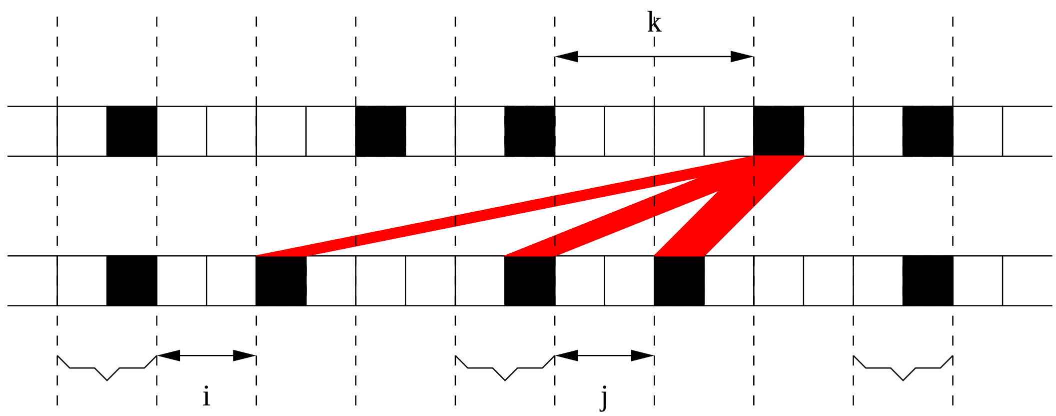


$$\delta(s_1, \ldots, s_n) = \sum \sum s_{-k} s_{-k-1} s_{-k-b+i} s_{-k+j}$$

▶ This polynomial has degree 4.

Degree 3

► Positional information can be encoded into the monomials. Here we use position of state 1 modulo 2.

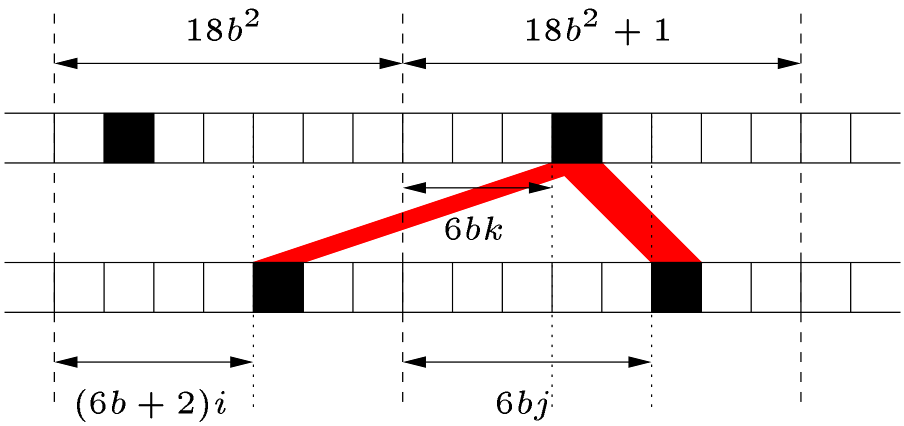


$$\delta\left(s_1, \ldots, s_n\right) = \sum \sum s_{-2k-1} s_{-2k-2b+2i} s_{-2k+2j}$$

► This polynomial has degree 3.

Two-State Universal Bilinear CA

- The idea is to forget the cell border information.



$$\delta(s_1, \ldots, s_n) = \sum \sum s_{18b^2-(6b+2)i+6bk} s_{6bj-6bk} + s_{18b^2+1-6bi+(6b+2)k} s_{(6b+2)j-(6b+2)k}$$

Two-States Bilinear Intrinsically Universal Cellular Automata. FCT'2001

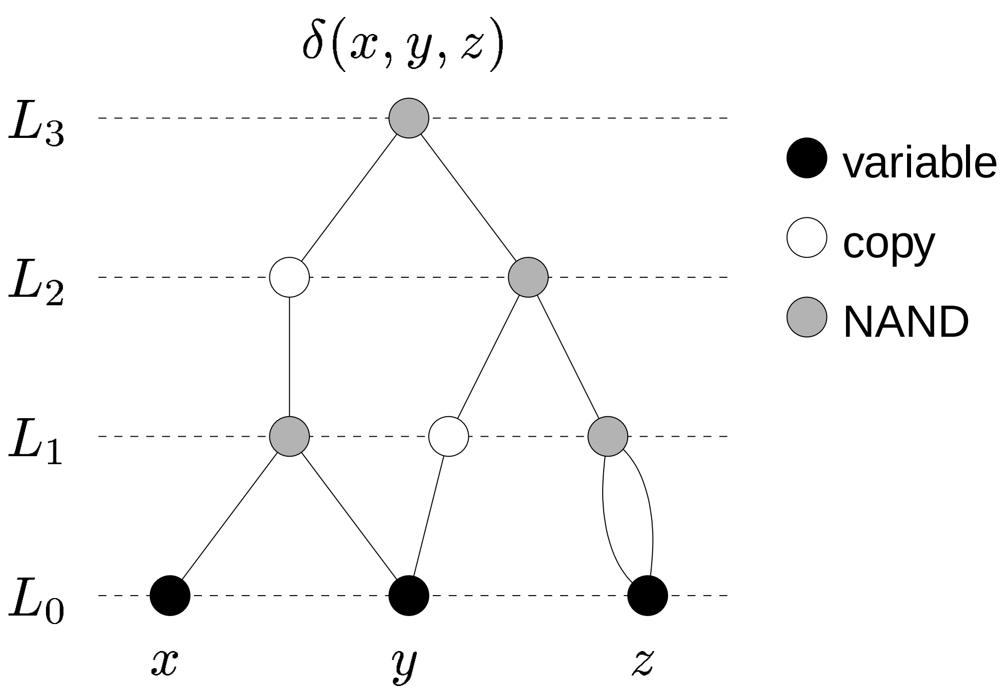
Contents

CA and boolean circuits

▶ We can decompose a CA local rule into k boolean functions where $k = |S|$:

$$\delta_i : \{0,1\}^{nk} \to \{0,1\}$$

▶ To a boolean function we associate a leveled circuit:

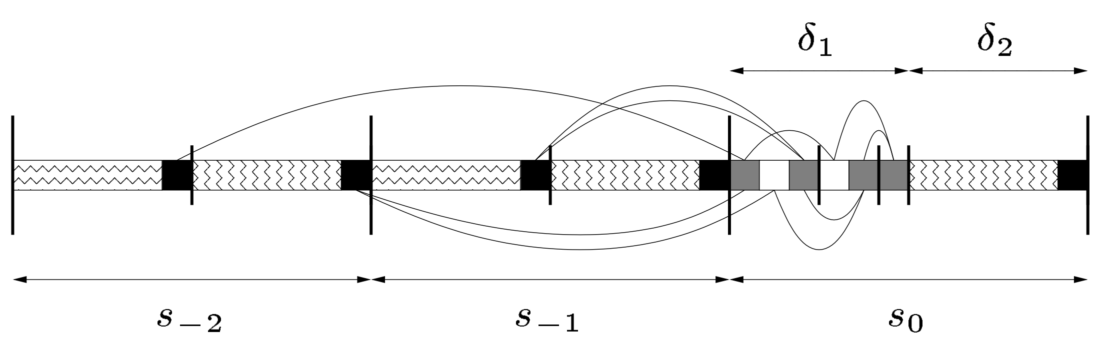


Boolean circuit simulator

▶ A boolean circuit simulator is a 1D dynamical system that simulates a CA via its boolean circuit representation.

Definition. A CA simulator consists of cells containing:

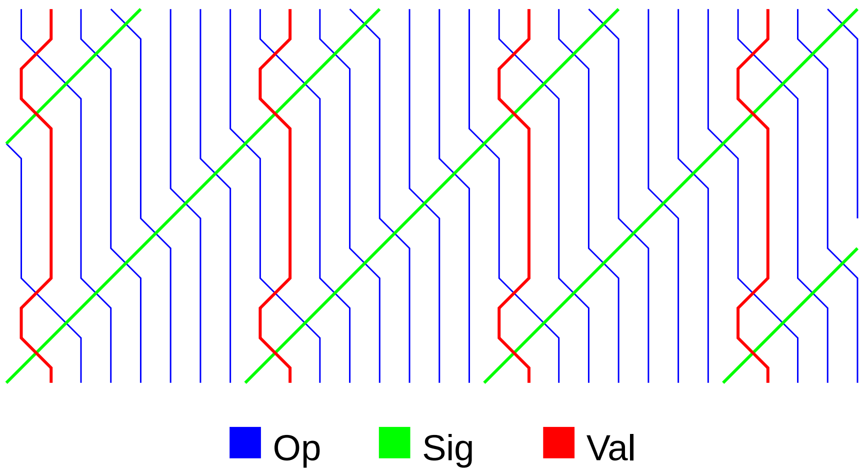
- a boolean value,
- an operator (identity or NAND),
- the relative position of the operands.



Idea. If a 1-CA can simulate a boolean circuit simulator, then it is intrinsically universal.

Microscopic Description

▶ We construct a 3-state 1-CA to move information.



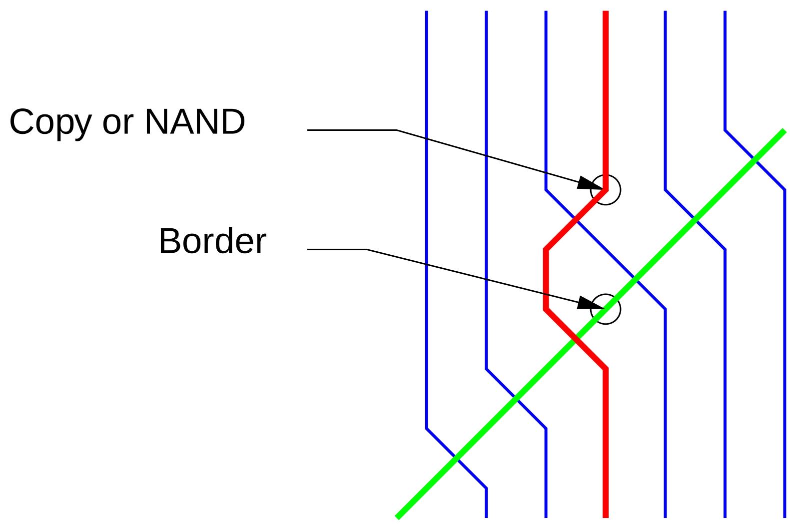
- Sig cells transport boolean values between cells,
- Val cells encode current cells values,
- Op cells encode the operations to execute.

▶ It is enough to encode a Boolean circuit simulator.

8 states

▶ We choose the following direct encoding:

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



7 states

- We choose the following encoding:
  - Sig: 0 or 1 boolean value,
  - Val: 0 or 1 boolean value,
  - Op: Border, Follow or NAND operation.

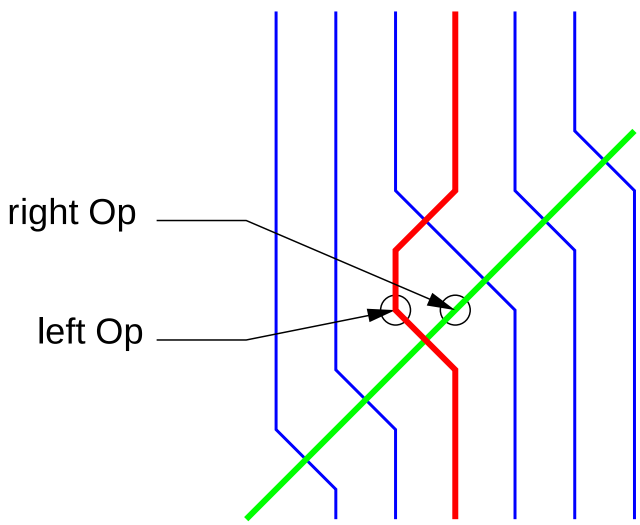
- The previous Copy operation is simulated. We encode a signal x by three consecutive signals 1, x, 0.

- The operations are encoded as follows:

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

6 states

- We choose the following direct encoding:
  - Sig: 0 or 1 boolean value,
  - Val: 0 or 1 boolean value,
  - Op: 0 or 1 boolean value position dependent.



- The construction is more tricky!

The Notion of Universality

► This notion is worth studying but difficult to formalize.

► Is there a 2-state first-neighbors universal 1-CA ?

► Is it possible to define a kind of acceptable programming system on CA where universal programs would be intrinsically universal CA ?