Linear Acceleration for 1-dim Cellular Automata

Véronique Terrier

GREYC - Université de Caen

Context

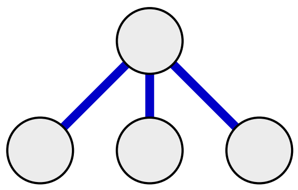
Algorithmic aspects of cellular automata

Generic tools:

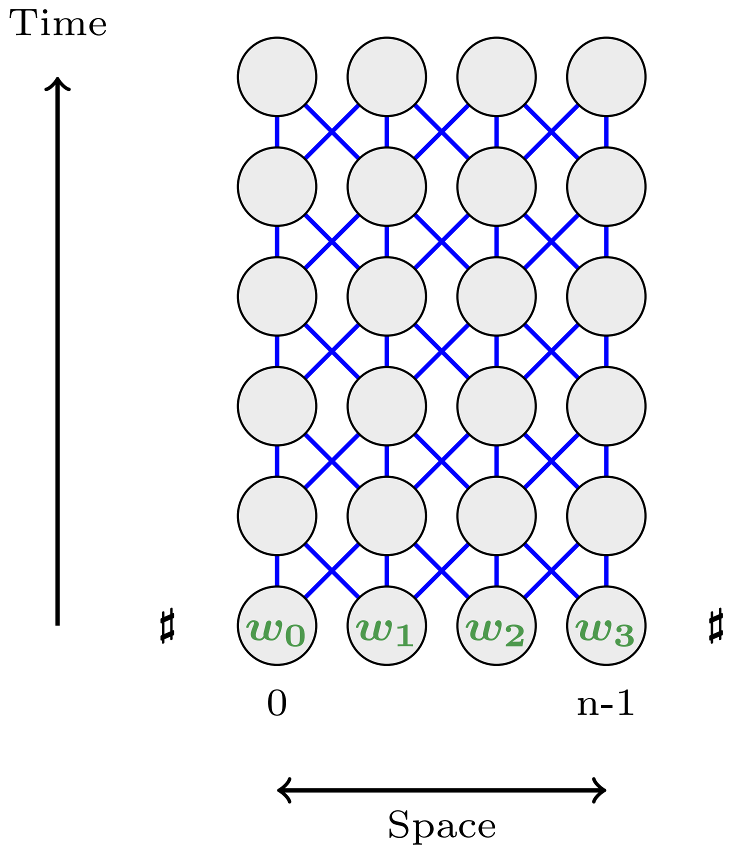
- Signals
- Particle and collisions
- Geometric transformations

Dependency graph

A directed graph which reflects the neighborhood constraint on the space-time diagram.



The local dependencies induced by the neighborhood



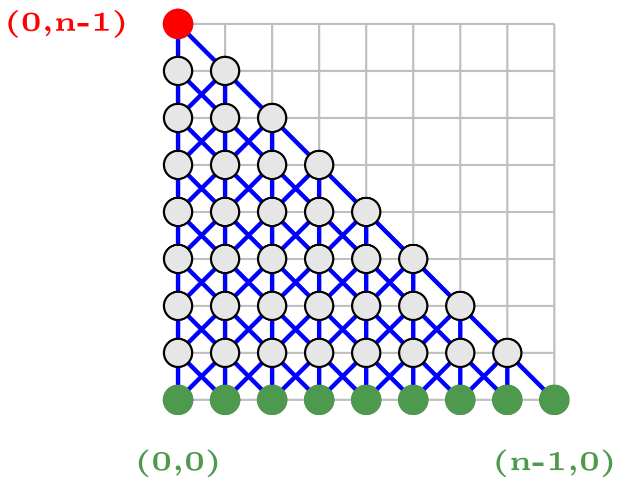
Affine transformation

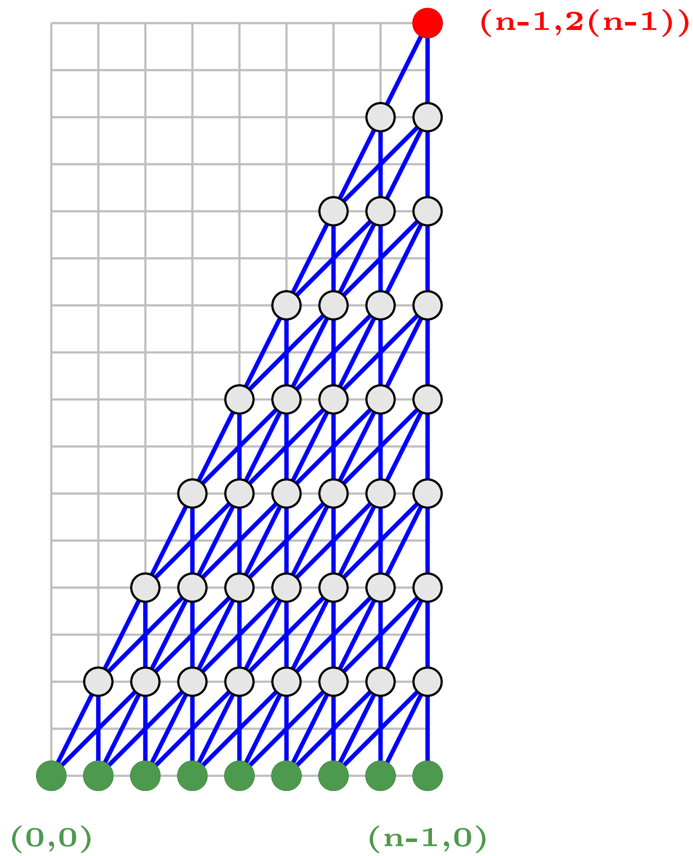
Homogeneous transformations which apply on the graph dependency

$$\mathbf{C} = (-\mathbf{1}, \mathbf{0})$$

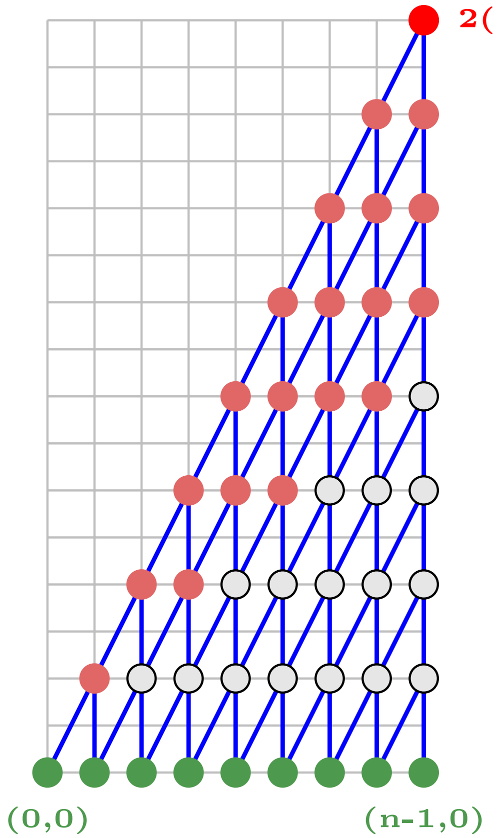
$$\boldsymbol{T} = \begin{pmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{2} \end{pmatrix}$$

$$\Pi(site) = C + T \times (site - C)$$





Affine transformation

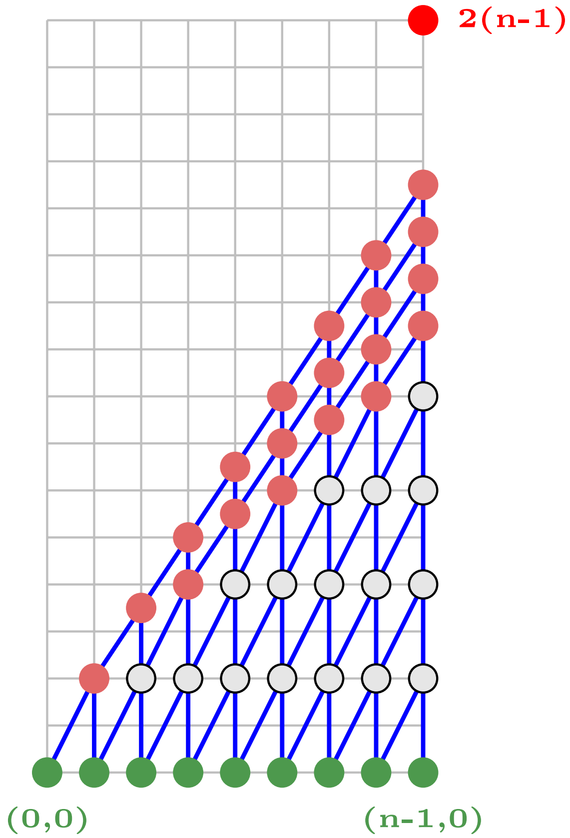


applied on a specific area

$$\mathcal{X} = \{(c,t) : t - c > 0\}$$

$$C = (-1, 0)$$

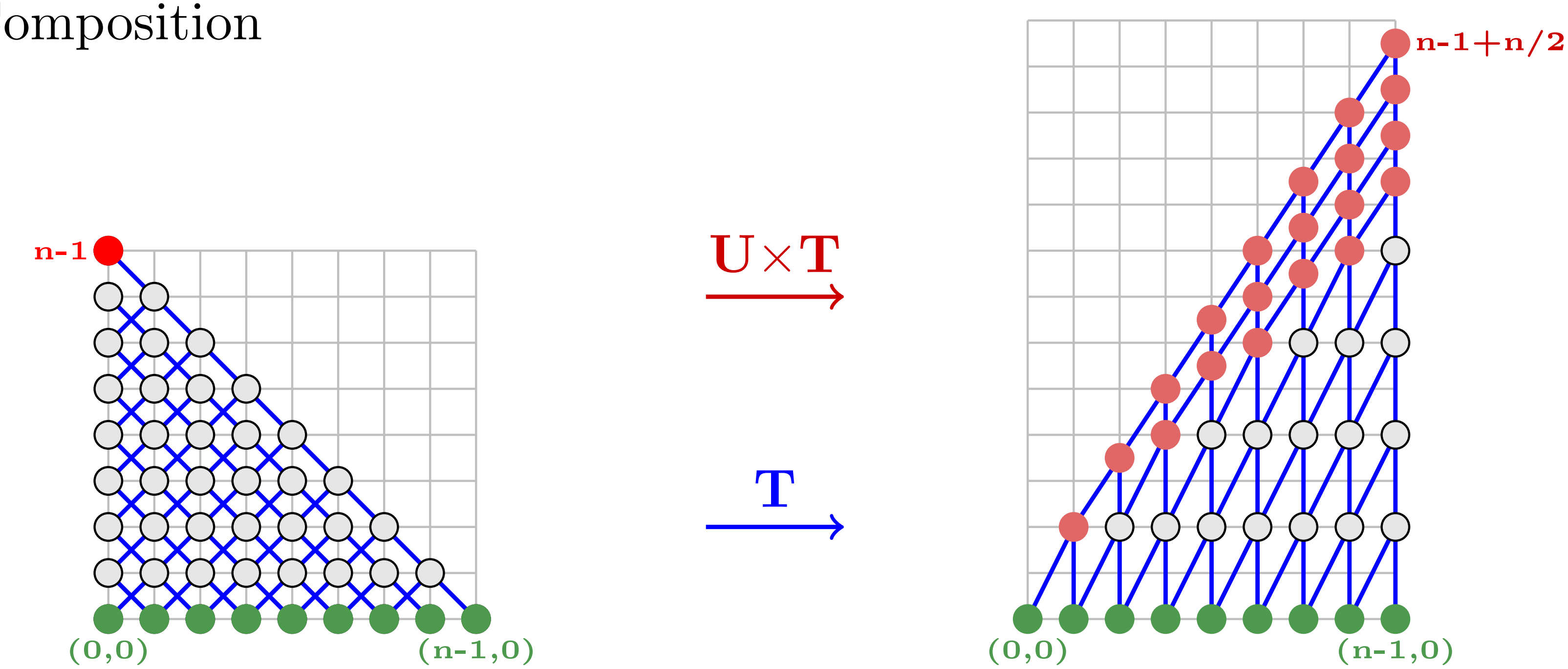
$$U = \begin{pmatrix} 1 & 0 \\ 1/2 & 1/2 \end{pmatrix}$$



$$\Pi(site) = \begin{cases} C + U \times (site - C) & \text{if } site \in \mathcal{X} \\ site & \text{otherwise} \end{cases}$$

Affine transformation

Composition

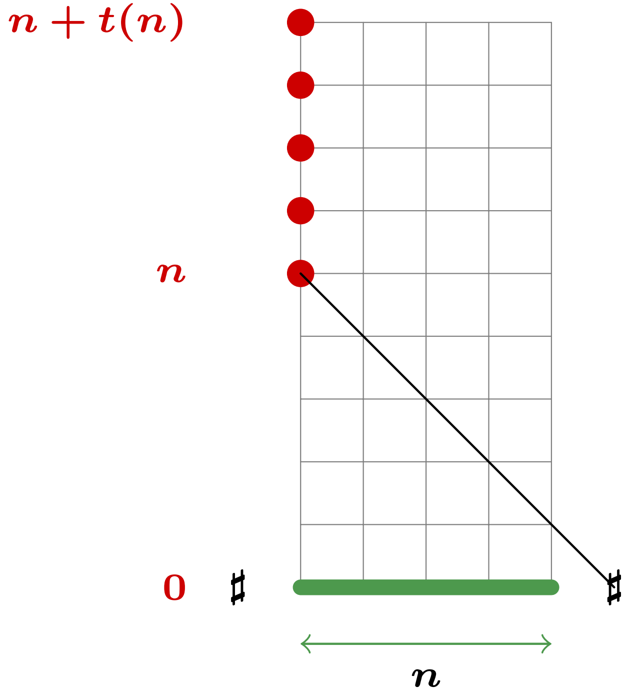


Which transformations can be realized on CA ?

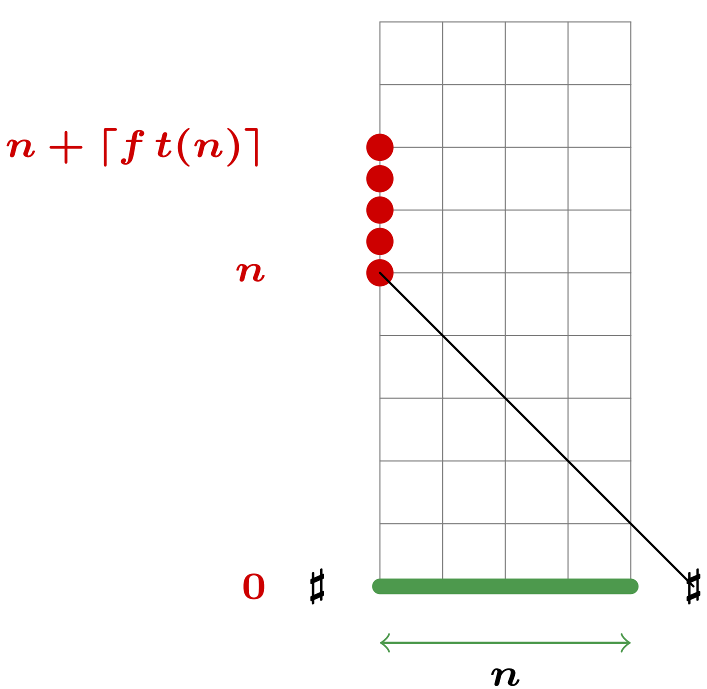
Linear Acceleration

$t : \mathbb{N} \to \mathbb{N}$ a function. $f \in \mathbb{Q}$ a positive ratio

From any CA which recognizes some language $\mathbf{L}$ in time $n + t(n)$,
we can construct another CA which recognizes $\mathbf{L}$ in time $n + \lceil f\, t(n) \rceil$.

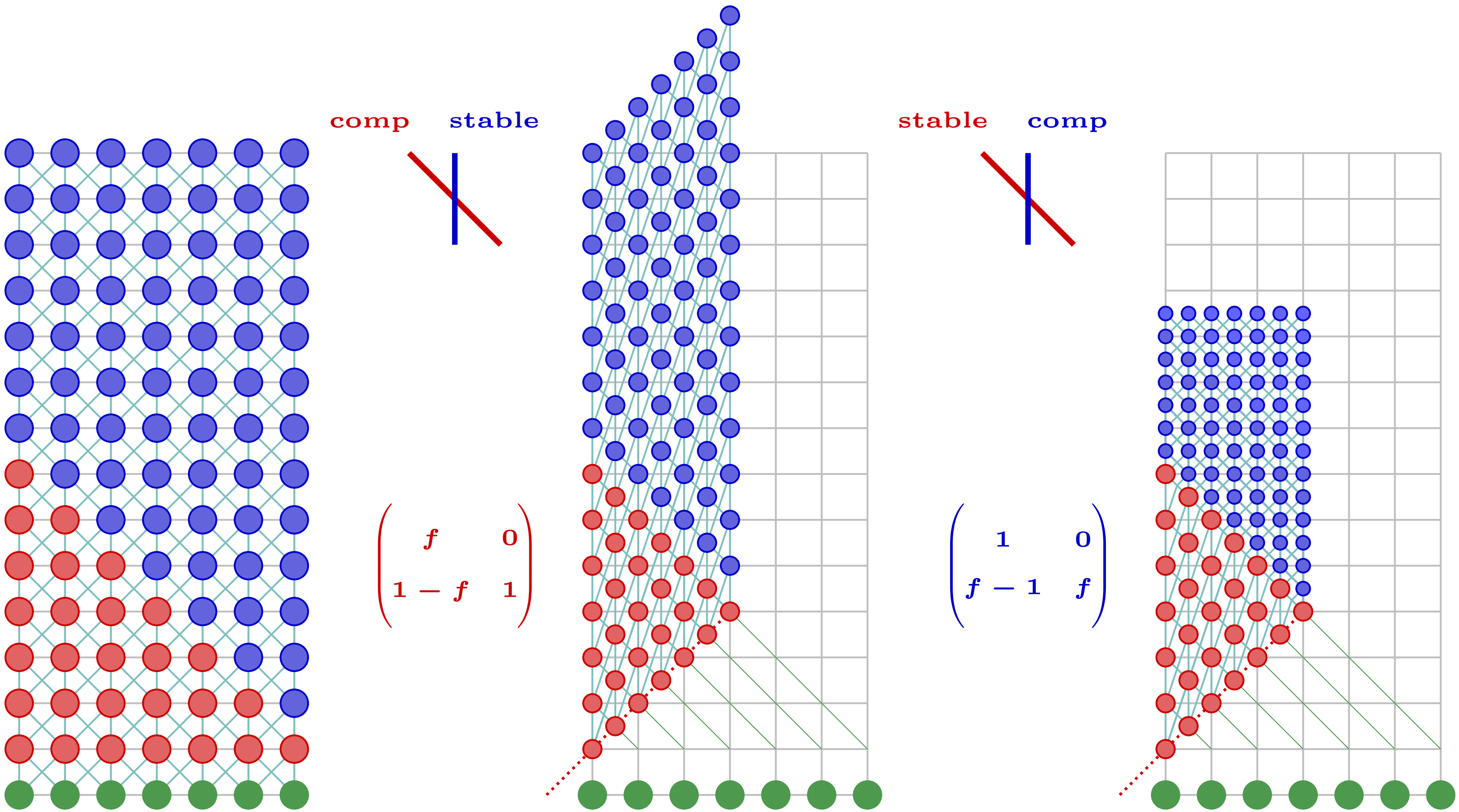


The initial CA $\mathcal{A}$

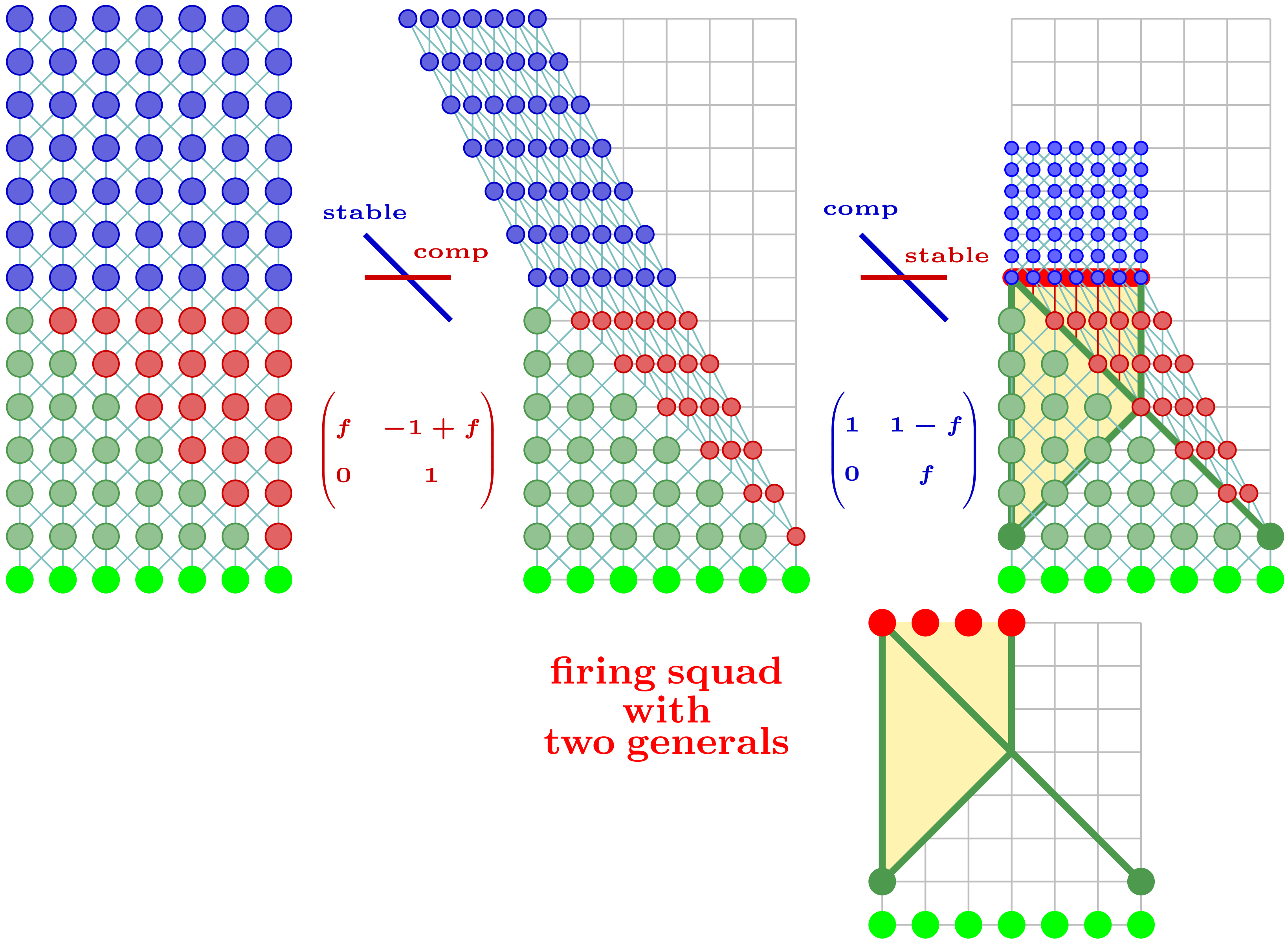


The accelerated CA $\mathcal{B}$

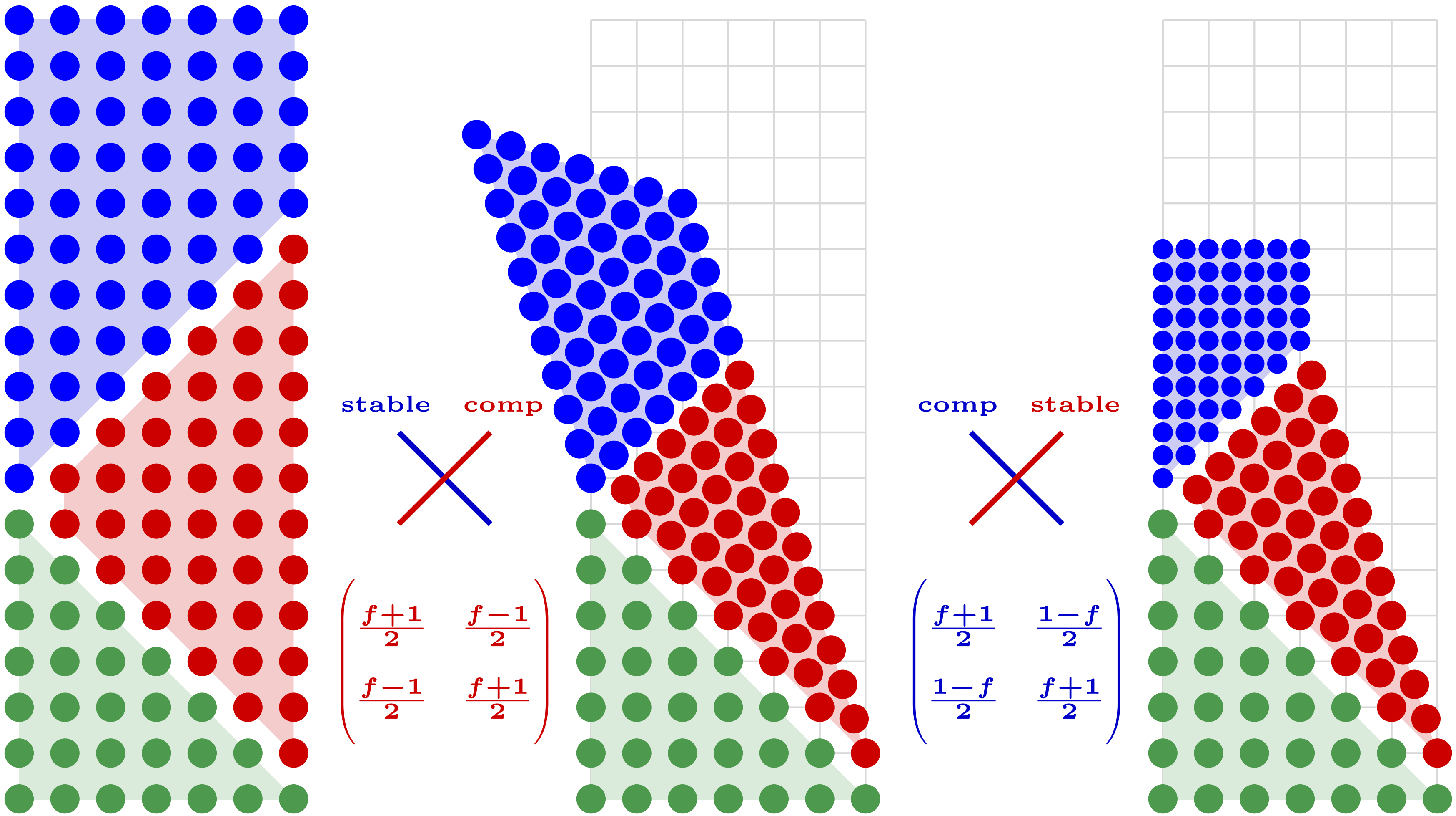
Beyer algorithm



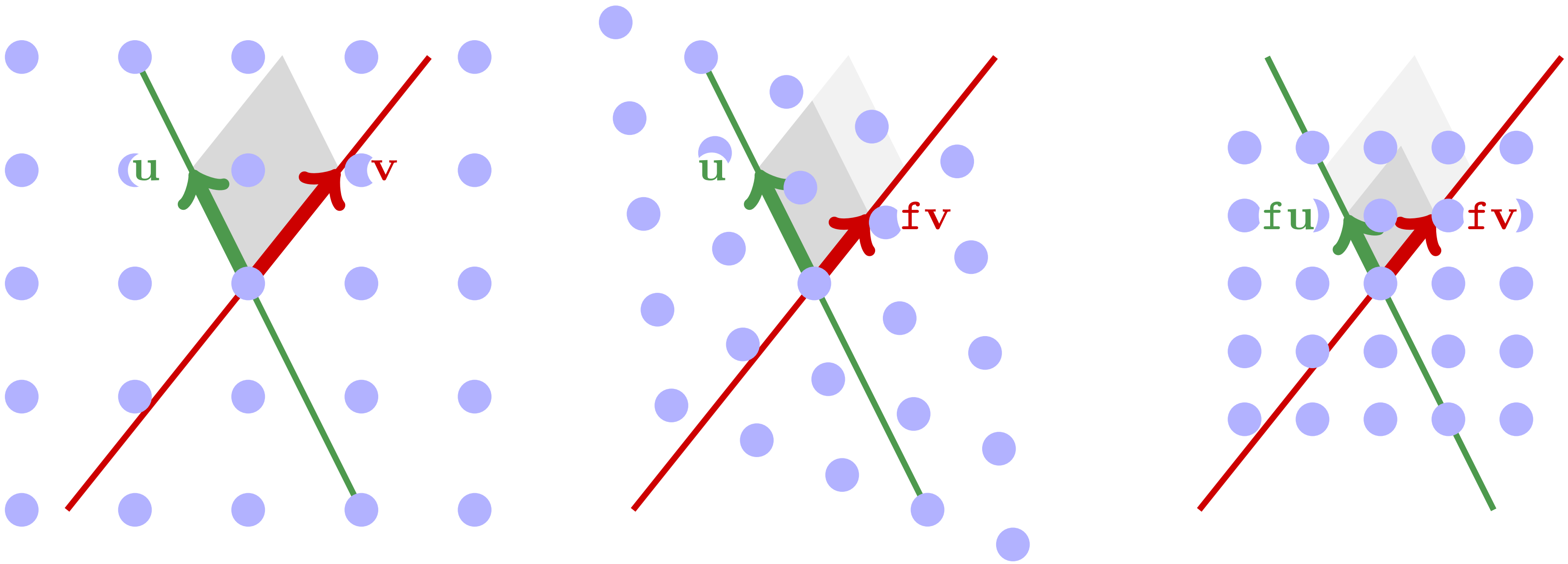
Acceleration with a Firing Squad *(Mazoyer & Reimen)*



Symmetric variant *(Heen)*



A characteristic of the above acceleration algorithms



Two axes: **u** , **v**

Composition of two complementary compressions:

- The first one compresses along **v** and leaves **u** stable
- The second one compresses along **u** and leaves **v** stable

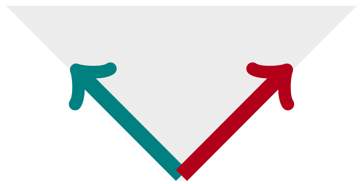
Validity

Which compressions can be realized on CA ?

Different requirements:

- Neighborhood constraint
- Input constraint
- Areas construction

Neighborhood constraint



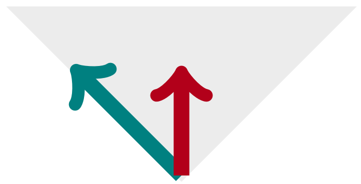
The neighborhood extremities

$\boldsymbol{e_1} = (-1, 1)$ and $\boldsymbol{e_2} = (1, 1)$

Their images must be contained in the neighborhood cone

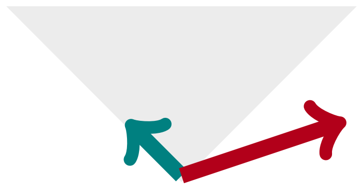
$$\{\alpha \mathbf{e_1} + \beta \mathbf{e_2} : \alpha, \beta \geq 0\}$$

Valid transformation



the first compression involved in the acceleration with a FS

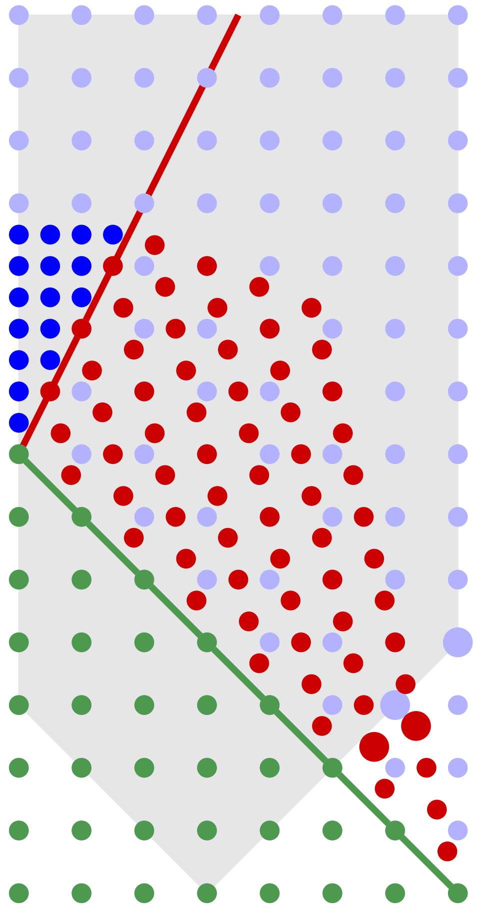
Non valid transformation



the second compression involved in the acceleration with a FS

Input constraint

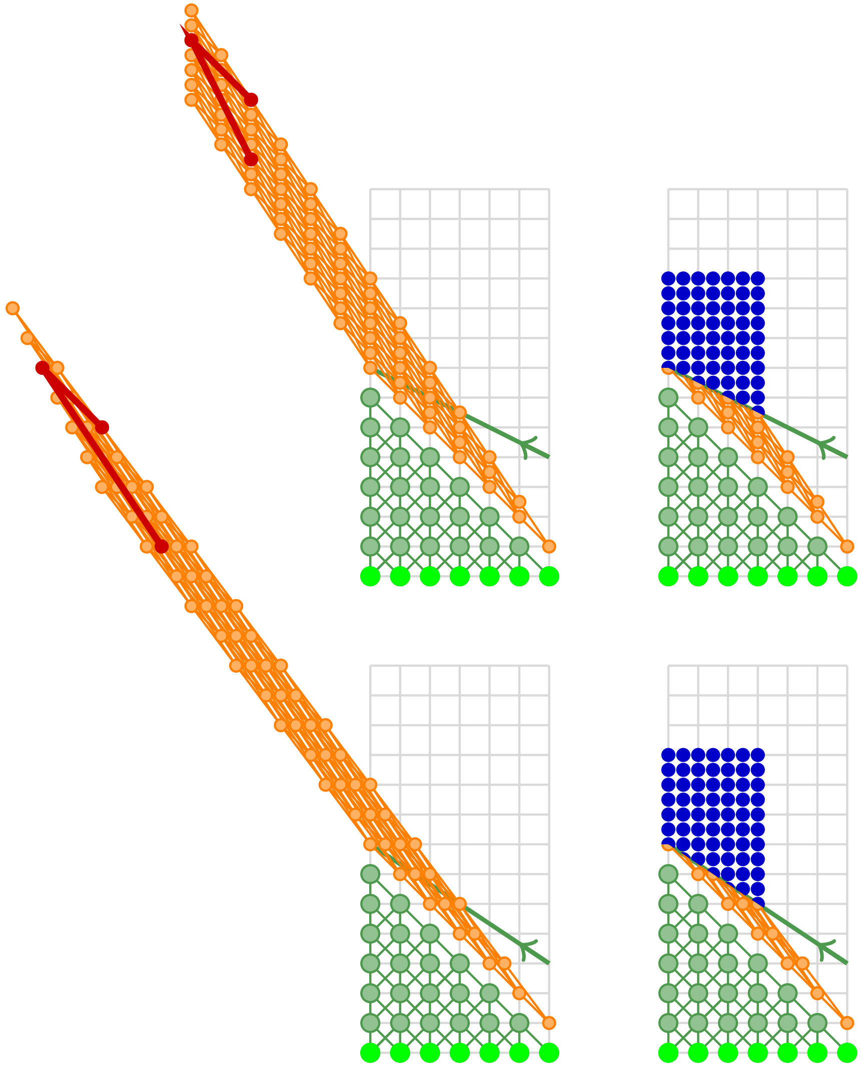
The consequences of $(i, 0)$ are contained in the cone leaving from $(i, 0)$



Areas construction

$u = (-1, 1)$
$v = (1, -\frac{1}{2})$

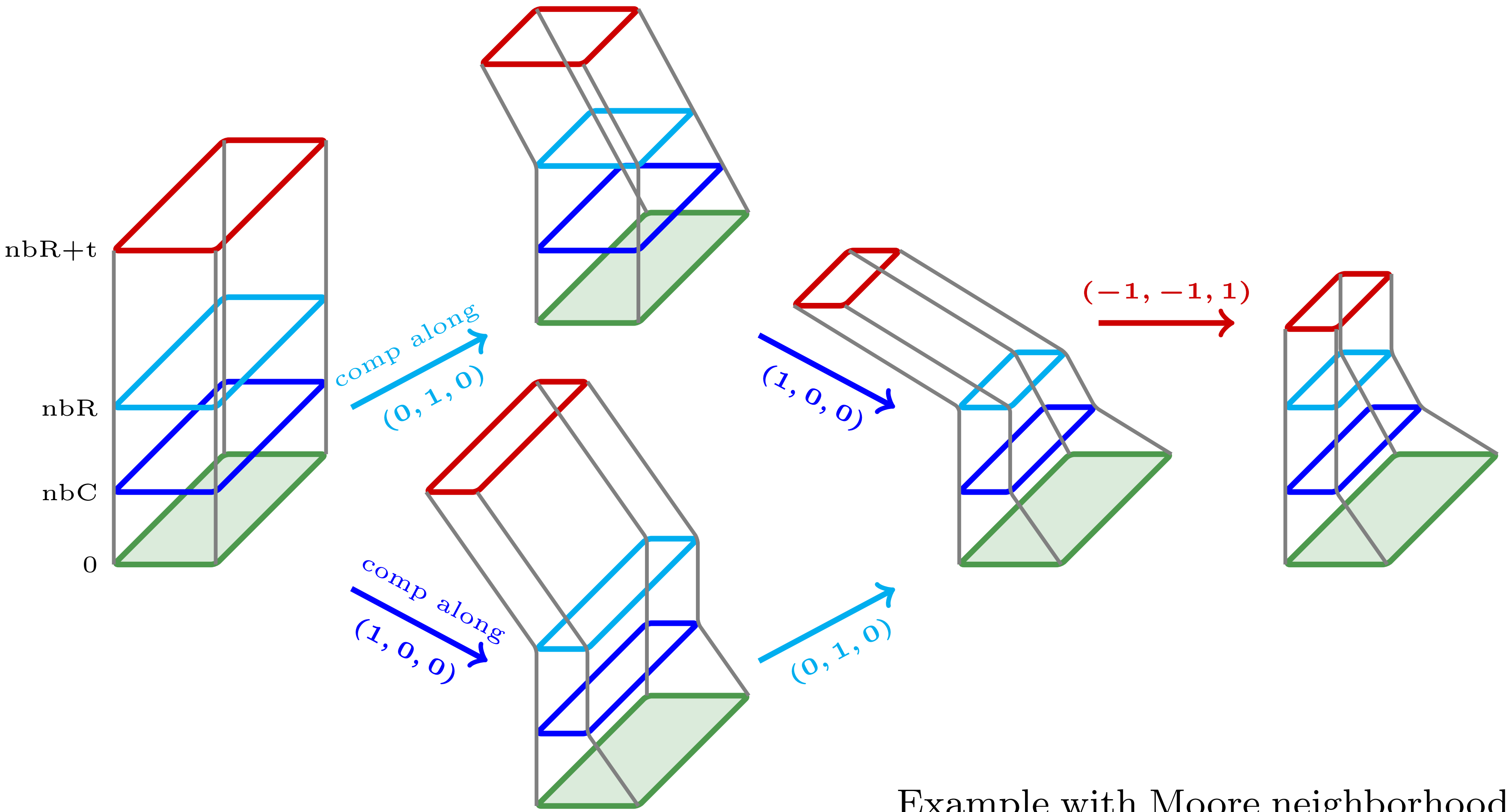
$u = (-1, 1)$
$v = (1, -\frac{2}{3})$



The compressions apply on specific areas.

Which areas can we delineate?

In higher dimensions



Example with Moore neighborhood

Combination of 3 complementary compressions
with Firing Squad and freezing