Cellular automata

Phys 750 Lecture 3

Cellular automata

- cellular automata system is a grid of cells evolving synchronously according to a discrete global clock
- discrete and (usually) finite set of states in each cell
- computation is exact and deterministic (and in some cases time-reversible)
- evolution rules are local; e.g., $x_i^{(t+1)} := F(x_{\{i,i\pm1\}}^{(t)})$

time step label

cell position label

Causal structure

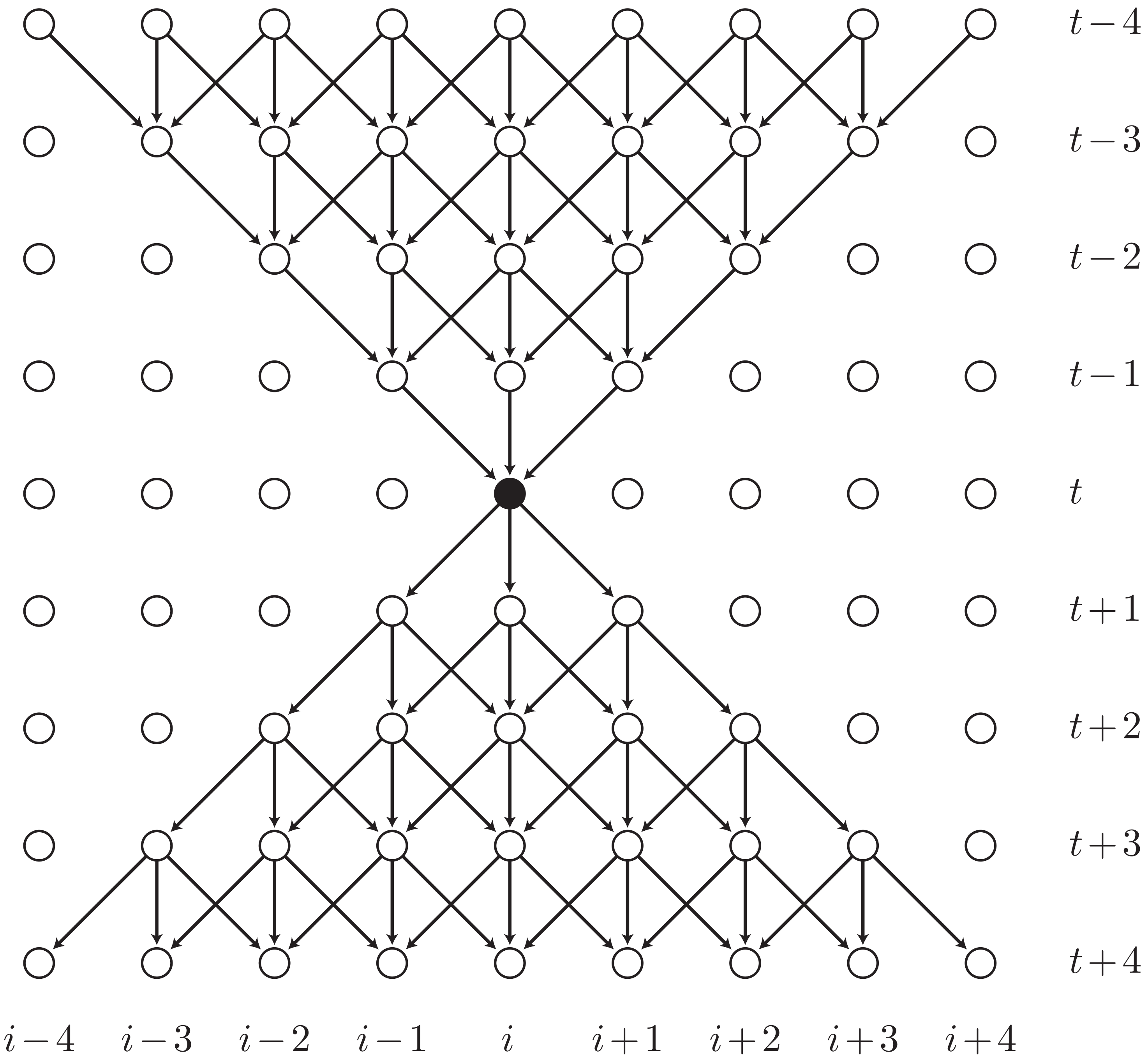
states in the past that were influential

↑

$x_i^{(t)}$

↓

states in the future that can be influenced



Synchronous updates

- Take care to avoid polluting the update step
- This single-array implementation conflates the two adjacent time steps and is incorrect:

```
bool alive[100] = {false, true, ...};
bool F(bool, bool, bool); // rule prototype

void update(void)
{
   for (int i = 0; i < 100; ++i)
      alive[i] = F( alive[(i+99)%100],
                    alive[i],
                    alive[(i+1)%100] );
}
```

Synchronous updates

```
#include <algorithm>
using std::swap;

bool buffer1[100] = {false, true, ...};
bool buffer2[100];

bool *current = buffer1, *next = buffer2;
bool F(bool, bool, bool);

void update(void)
{
   for (int i = 0; i < 100; ++i)
      next[i] = F( current[(i+99)%100],
                   current[i],
                   current[(i+1)%100] );
   swap(current,next);
}
```

$t + 1$

t

Synchronous updates

```
#include <algorithm>
using std::swap;
#include <vector>
using std::vector;

// pointer-free implementation
vector<bool> current(100);
vector<bool> next(100); // two STL containers

void update(void)
{
   for (int i = 0; i < 100; ++i)
      next[i] = F( current[(i+99)%100],
                   current[i],
                   current[(i+1)%100] );
   swap(current,next);
}
```

Excitable media CA

- Discrete model with $E + 1$ possible states in each cell:

Quiescent	$x_i = 0$
Excited	$x_i = E$
Refractory	$x_i \in \{1, 2, \ldots, E - 1\}$

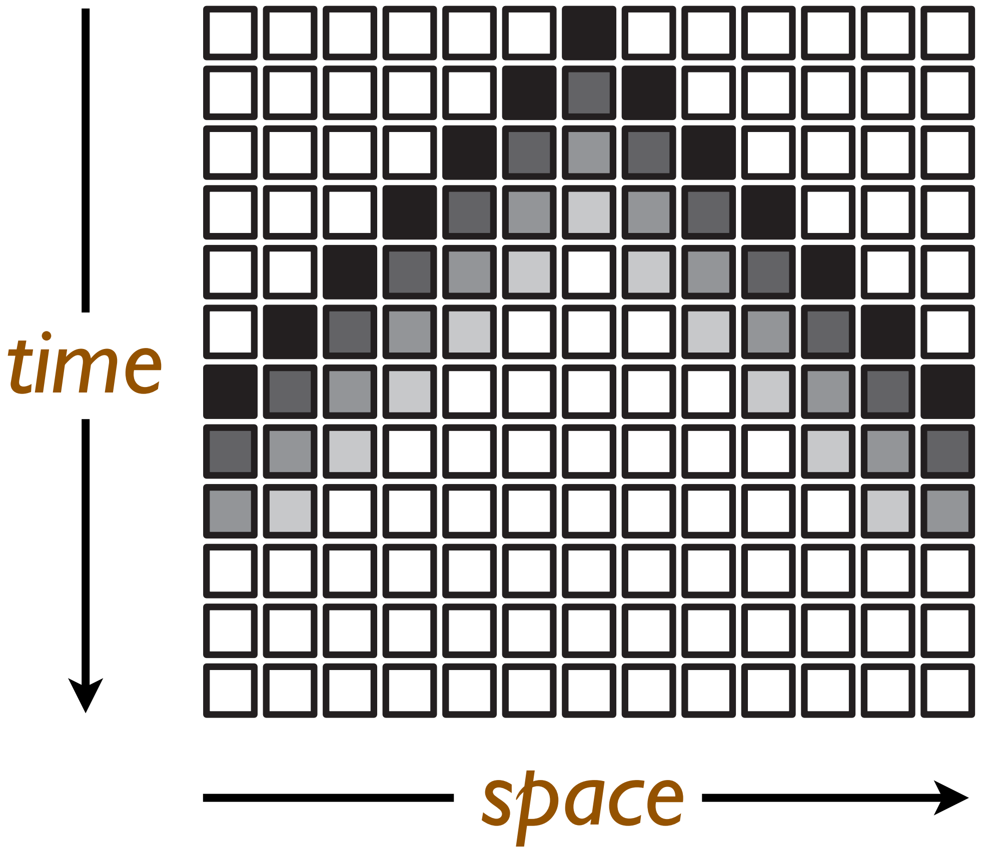
- Quiescent cells are excited by their excited neighbours
- Excited cells relax over the course of $E - 1$ time steps

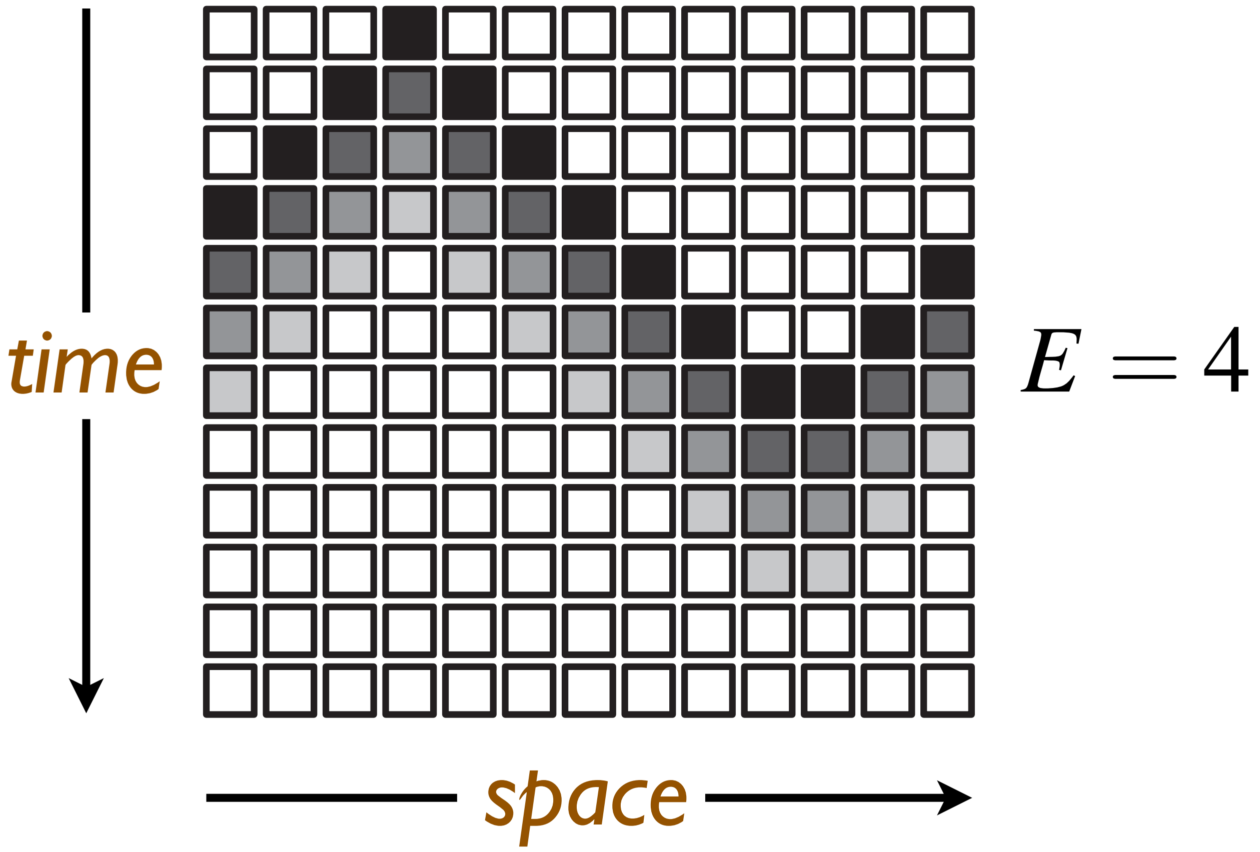
Excitable media CA

- Precise statement of the update rules:

- For the case $x_i^{(t)} = 0$
    - if at least one of the neighbours $x_{j \neq i}^{(t)}$ is excited, assign $x_i^{(t+1)} := E$
    - otherwise, set $x_i^{(t+1)} := 0$
- Otherwise, set $x_i^{(t+1)} := x^{(t)} - 1$

Excitable media CA

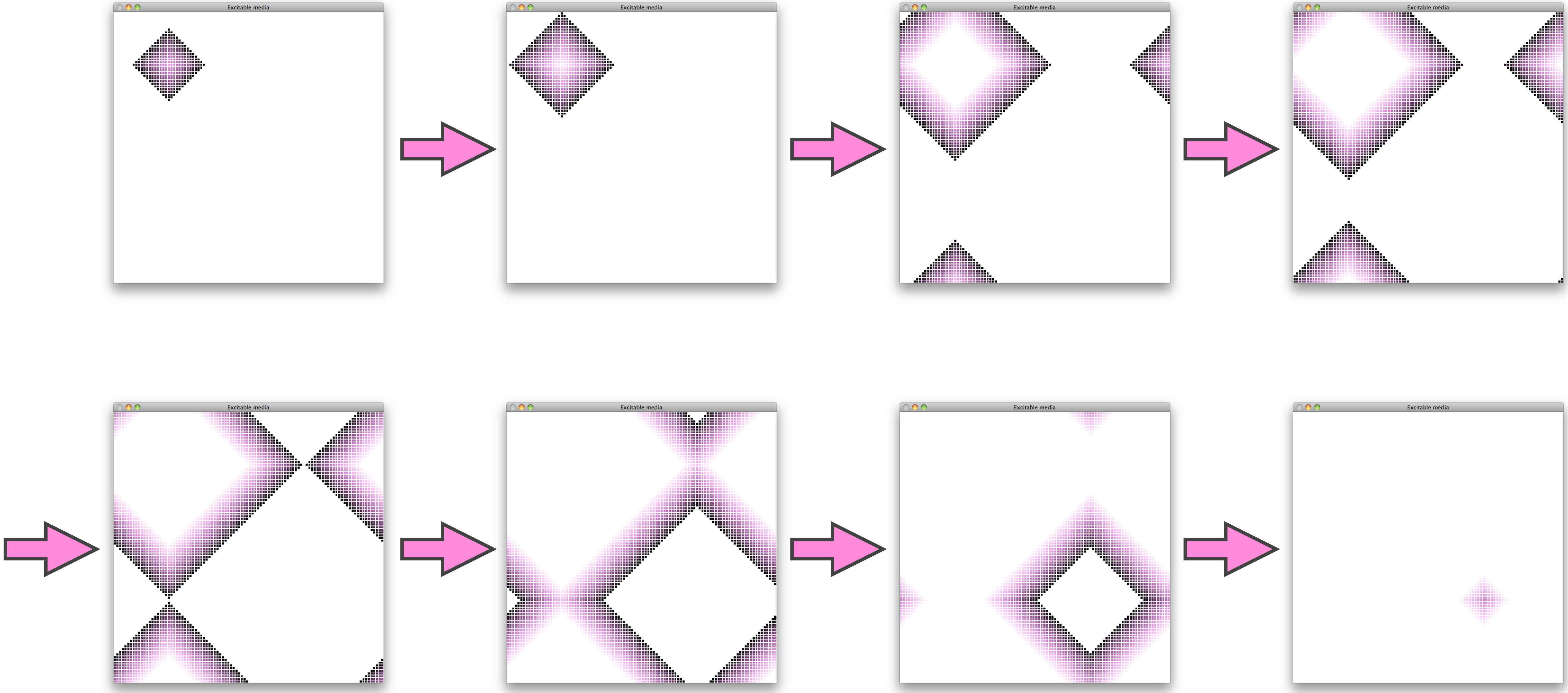




- Localized disturbances propagate away from initial perturbation with definite “momentum” $(E \geq 2)$
- Completely inelastic collisions: “particles” annihilate

Excitable media CA

square lattice, $E = 20$



Excitable media CA

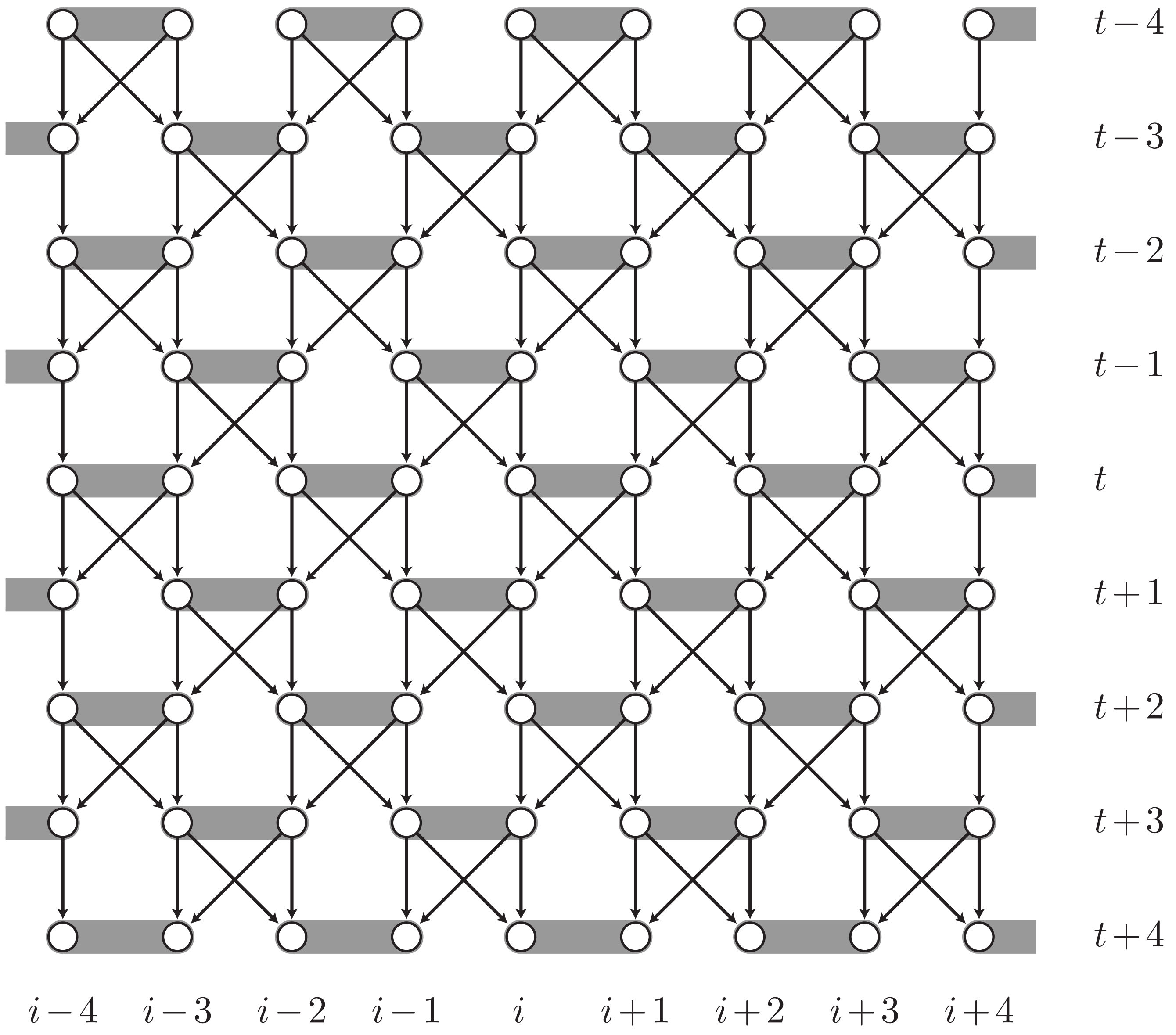
- Partial analogy with wave propagation—corresponding to a pebble dropped in a pond
- Analogy fails in a few important ways:
  - unimpeded wavefront never decays at long distances
  - complete annihilation of colliding excitations
  - cannot recover circular symmetry by coarse-graining

Excitable media CA

- Failures are a consequence of bad choices at the microscopic level:
  - update rules should enforce local energy conservation if we want global energy conservation
  - pay attention to the discretization of space and the connectivity of the lattice

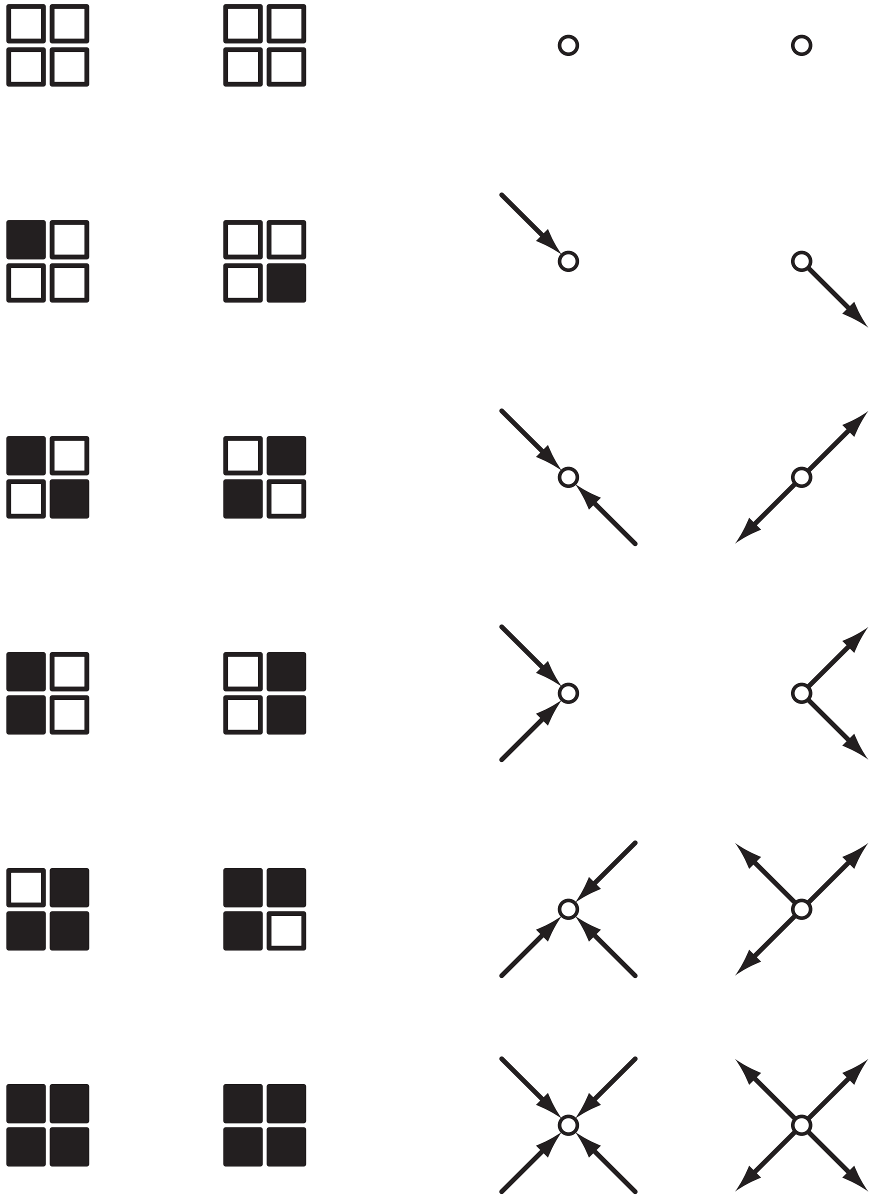
Margolus scheme

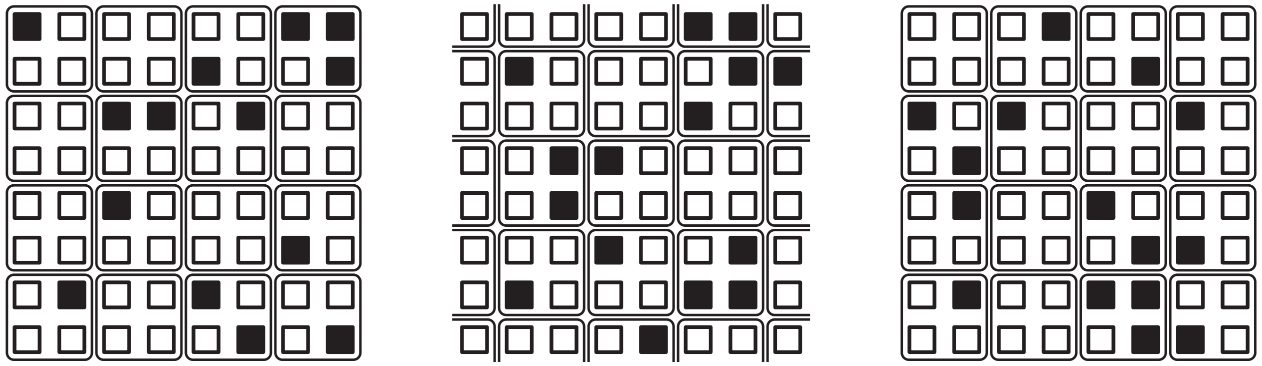
- Space-filling partition of groups of cells, alternating with time
- Each group updated independently
- Easy to enforce conservation laws, especially number conservation
- 1d lattice gas: ■□ ⟷ □■



Square lattice gas

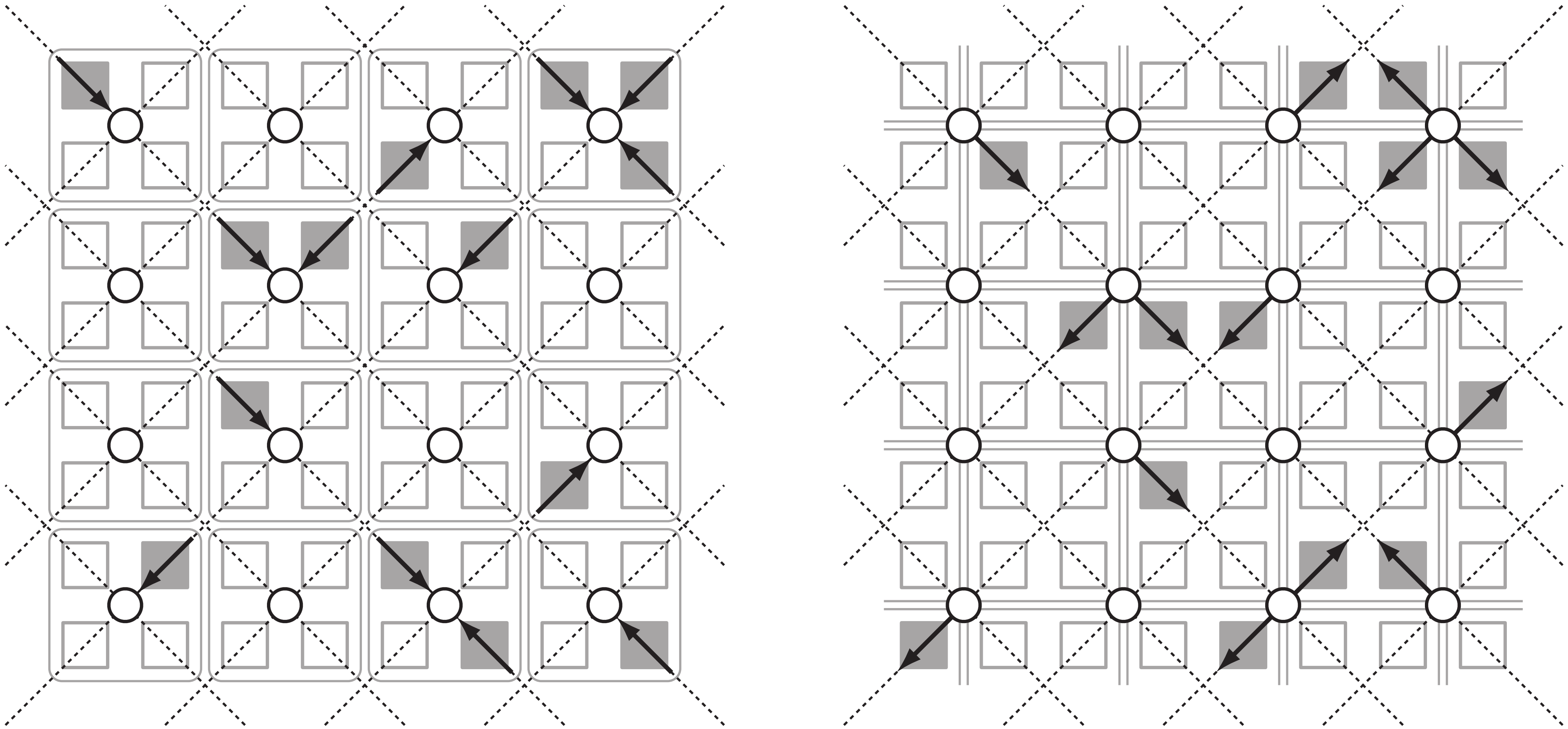
- 4-site Margolus tiling, shifted by (1,1) for odd times
- Updates conserve particle number, energy, and momentum (along the links of the dual lattice)





Square lattice gas

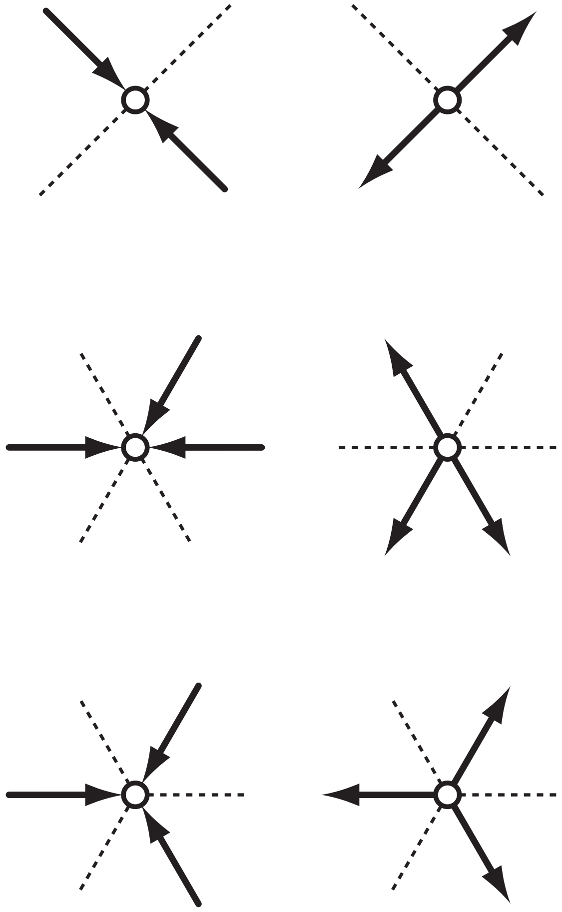
- Velocities allowed along only two orthogonal directions
- Leads to many undesirable hidden invariants



momentum conserved independently along every line of the dual lattice

Coarse-grained behaviour

- Long-distance, long-time behaviour is related to Navier-Stokes hydrodynamics but with an artificial anisotropy
- Spurious invariants survive coarse graining
- Cured by three-body interactions on the triangular lattice



CA generalizations

- Arbitrary grids and lattices
- Causal connections beyond nearest neighbour
- Asynchronous updates
- Random/probabilistic updates
- Boundary conditions